DISS. ETH NO. 29820

Co-Design of Complex Systems: From Autonomy to Future Mobility Systems

A dissertation submitted to attain the degree of

DOCTOR OF SCIENCES OF ETH ZURICH
(Dr. sc. ETH Zurich)

presented by

GIOELE ZARDINI
MSc ETH in Mechanical and Process Engineering, ETH Zurich
born on February 3, 1993

accepted on the recommendation of

Prof. Dr. Emilio Frazzoli (ETH Zurich)
Dr. Andrea Censi (ETH Zurich)
Prof. Dr. Marco Pavone (Stanford University)

2023

to So

Acknowledgments

Over the past years, I have been incredibly fortunate, and there is a lengthy list of individuals I would like to express my gratitude to.

First of all, I want to thank the doctoral thesis committee. These three individuals have played pivotal roles in my journey, and I consider each of them a stroke of luck in my life. I consider Emilio, Andrea, and Marco phenomenal, complementary advisors, and at the same time trustworthy friends.

My introduction to Emilio occurred through his inaugural seminar at ETH, and I was immediately captivated by his crystal-clear vision, charm, and friendliness. I instantly decided I wanted to work with him, and since that day, he has been a constant source of guidance and inspiration, both academically and in life. His words "don't be afraid of taking the road less travelled" resonated in my head throughout this journey. He has always believed in me and supported me, and I owe him a lot of what I have achieved. Also, I wish to someday master travels and jet lag as he does. The second stroke of luck was my encounter with Andrea. He not only imparted academic rigor and a sophisticated approach to scientific writing and exposure but also instilled in me the importance of asking the right questions, scientific integrity, and the need to operate "one level up". His perspectives have always been very refreshing and he is definitely the jolly in the card deck of engineering research, when the game happens during the day in pacific time ("mornings don't exist"). I learned so much from him. My third stroke of luck was working with Marco. I met him during my visit to his lab at Stanford in early 2019 and we have been interacting since. I have been fascinated by his ability to convey big-picture concepts across a wide range of topics (breadth) while simultaneously developing intricate and effective tools that are challenging to explain (depth). He is as wise as a 60 years old person, and as energic as a 20 years old one, all while being around 40. He was also the first to advise me on improving my writing, or as he puts it, writing "up to standards", and our interactions have been incredibly enriching. Joining his group for a (very short) postdoc promises to unveil more magic.

This journey would have not been the same without Nicolas Lanzetti and Alessandro Zanardi, who are not only excellent co-authors, but also dear friends for life. Nicolas has been a fantastic companion of adventures (mostly food-related ones) since the early undergrad days, and I admire his pure soul, kindness, and positive attitude. If he will not get lost in a Wasserstein space, I'm sure that even more adventures will be possible. Alessandro has been a true "Ph.D. buddy": we shared the Ph.D. exeprience, inside and outside the lab, and, although working on different topics and having different characters, we have enriched each other's research and personal life in multiple ways.

A deep thanks goes to the other lab members and colleagues who made everyday special, including Jonathan Lorand, Dejan Milojevic, Marc Albert, Ezzat Elokda, Claudio Ruch, Julian Zilly, Maurilio Di Cicco, Jacopo Tani, Mingjia He, Matteo Penlington, Annina Fattor, Marc Neumann, Elena Arcari, Carlo Sferrazza, Saverio Bolognani, Florian Dörfler, Riccardo Bonalli, Dario Paccagnan, and Maximilian Schiffer.

Now it's time to thank the category theory gang. I want to thank Dr. David Spivak for having introduced me to the magic world of category theory. In the beginning of the Ph.D. I had the chance to spend some time

with him at MIT, working on ideas on event-based systems [1], and his mentorship (made of long discussions at the blackboard, a lot of coffee, and many questions) allowed me to fall in love with the subject. This indeed served as an "ice-breaker" for the thesis. I want to thank the "Applied Category Theory Adjoint School" team, including Vincent Wang, Bryce Clarke, Maurine Songa, Emma Chollet, and Michael Johnson, who bring dear memories. During this project (which converged in an awarded paper on lenses [2]), we met regularly online, from three (at times four) different continents, keeping up the spirits in difficult pandemic times.

During my time at ETH, I had the great opportunity of mentoring students for semester and masters theses. This experience was exceptional and enriching at various levels. In order of completion, I want to thank Richard Von Moos, Johannes Conradi, David Gerber, Aleksandar Petrov, Gianmarco Bernasconi, Laura Guerrini, Yannik Glauser, Giulio Carassai, Sirish Srinivasan, Claudia Godignon, Marius Furter, Alessio Mina, Zelio Suter, Rohan Flogel-Shetty, Christian Hartnik, Sylvain Fricker, Leon Zueger, Carla Paillardon, Luca Autunno, Ryan Mukherjee, Luca Sandel, Yujun Huang, Alessandro Cecconi, Manuel Schneider, and Pjeter Berisha.

I want to thank my parents Gian Anton and Giordana, and my brother Giona, for their unconditional support and love over the past years (and if you are wondering, yes, we are the "G's"). Finally, but most importantly, I want to extend my deepest gratitude to my wife Sonia. The past 13 years with you have been phenomenal, and I feel like the luckiest person alive. This is just the beginning.

Abstract

The contemporary era struggles with the intricate challenge of designing “complex systems”. These systems are characterized by intricate webs of interactions that interlace their components, giving rise to multifaceted complexities, springing from at least two sources.

First, the co-design of complex systems (e.g., a large network of cyber-physical systems) demands the simultaneous selection of heterogeneous components (e.g., hardware vs. software parts), while satisfying system constraints and accounting for multiple objectives. Second, different components are interconnected through interactions, and their design cannot be decoupled (e.g., within a mobility system).

Navigating this complexity necessitates innovative approaches, and this thesis responds to this imperative by focusing on the theory of co-design. Our exploration extends from the design of individual platforms, such as autonomous vehicles, to the orchestration of entire mobility systems built upon such platforms. In particular, we delve into the theoretical foundations of a monotone theory of co-design, establishing a robust mathematical framework, leveraging category theory to elucidate key concepts, including compositionality and functorial solution schemes in co-design.

Notably, this thesis offers not only an understanding of the theoretical underpinnings, but also practical guidance for applying them to a diverse array of real-world problems, revolving around the domain of embodied intelligence. The presented toolbox empowers efficient computation of optimal design solutions tailored to specific tasks and, in its novelty, paves the way for several possibilities for future research.

Zusammenfassung

Die heutige Zeit ist mit der schwierigen Herausforderung konfrontiert, “komplexe Systeme” zu entwerfen. Diese Systeme sind durch komplizierte Netze von Wechselwirkungen gekennzeichnet, die ihre Komponenten miteinander verflechten, was zu einer vielschichtigen Komplexität führt, die mindestens zwei Ursachen hat.

Erstens erfordert die Mitgestaltung komplexer Systeme (z.B. eines grossen Netzes cyber-physischer Systeme) die gleichzeitige Auswahl von Komponenten heterogener Natur (z.B. Hardware- und Softwareteile), während gleichzeitig die Systembeschränkungen erfüllt und mehrere Ziele berücksichtigt werden müssen. Zweitens sind verschiedene Komponenten durch Interaktionen miteinander verbunden, und ihr Design kann nicht entkoppelt werden (z.B. innerhalb eines Mobilitätssystems).

Die Beherrschung dieser Komplexität erfordert innovative Ansätze, und diese Arbeit reagiert auf diese Notwendigkeit, indem sie sich auf die Theorie des Co-Designs konzentriert. Unsere Untersuchung erstreckt sich vom Design einzelner Plattformen, wie z.B. autonomer Fahrzeuge, bis hin zur Orchestrierung ganzer Mobilitätssysteme, die auf solchen Plattformen aufbauen. Insbesondere befassen wir uns mit den theoretischen Grundlagen einer monotonen Theorie des Co-Designs und schaffen einen robusten mathematischen Rahmen, indem wir die Kategorientheorie nutzen, um Schlüsselkonzepte wie Kompositionalität und funktionale Lösungsschemata im Co-Design zu erläutern.

Diese Arbeit bietet nicht nur ein Verständnis der theoretischen Grundlagen, sondern auch praktische Anleitungen für deren Anwendung auf eine Vielzahl von realen Problemen, die sich um den Bereich der verkörperten Intelligenz drehen. Der vorgestellte Werkzeugkasten ermöglicht die effiziente Berechnung optimaler, auf spezifische Aufgaben zugeschnittener Designlösungen und eröffnet in seiner Neuartigkeit eine Fülle von Möglichkeiten für zukünftige Forschung.

Contents

1 Introduction

The proper study of mankind is the science of design.
—Herbert A. Simon[1]

The design of "complex systems" stands out as one of the paramount challenges of this century. Such systems are labeled as "complex" not only due to the intricacies of their individual components, but also because their functioning hinges on complex *interactions* among these components. An illustrative case in point for the complexity is the domain of *cities*.

Over the past few decades, cities worldwide have witnessed an unprecedented wave of urbanization. Presently, a staggering 55% of the global population calls urban areas their home, and by 2050 the proportion is expected to reach 68% [3]. As a direct consequence of the population density growth, urban travel has surged, causing a series of associated externalities [4]. In this dynamic landscape, urban planners are confronted with the formidable challenge of adapting their transportation systems to accomodate the escalating demands of society.

This task is inherently intricate for a multitude of reasons. First, cities must not only anticipate and cater to the evolving travel needs of their population [5], but also strive for equity and fairness in their transportation strategies [6]. Second, these strategies necessitate a careful consideration of their impact on other mobility providers, such as private mobility service providers (e.g., ride-hailing companies, micromobility services, and, in the future, Autonomous Mobility-on-Demand (AMoD) systems). Such services have witnessed remarkable growth in recent years, exemplified by the 1,000% increase in daily trips made by ride-hailing companies in New York City from 2012 to 2019 [7]. While extending more travel options to commuters, these systems operate while capitalizing on public resources such as roads and public spaces. They operate with a profit-oriented approach, and, at times, generate potentially disruptive consequences for both the efficiency of the transportation network, and society at large [8]–[10].

Third, these policies must be devised in alignment with global sustainability objectives, and mindful of their implications for other interconnected systems. It is a well-documented fact that cities bear a significant responsibility, contributing to 78% of the world's energy consumption and over 60% of global greenhouse gas emissions, with transportation accounting for a substantial 30% of this total in the United States [11]. Sustainability has taken center stage in policymaking worldwide, as evidenced by initiatives such as New York City's plan to elevate sustainable transportation from 68% to 80% [7] and the European Union's ambitious aim to slash emissions by 90% by 2050 [12]. New decisions and technologies

[1] Simon is the winner of the 1978 Nobel Prize in Economics.

promise to introduce profound changes to a series of other vital systems, including existing energy systems, infrastructure planning, urban development, employment patterns, lifestyle choices, and more. How should we invest for the next century's infrastructure? Which technologies should we develop to fight climate change? How can automation help? How will power grids deal with mobility systems of the future? These questions are deeply intertwined, rendering the comprehension of their complexity a seemingly insurmountable challenge, both for human minds and computer algorithms.

In light of these considerations, it becomes evident that tackling these socio-technical design problems necessitates innovative approaches. The complexity of these intertwined systems underscores the urgency for methods which enable their collaborative design – an imperative call for co-design methodologies. In this thesis, we will focus on a theory of co-design and its application to embodied intelligence, all the way from the design of a single platform (i.e., an autonomous vehicle (AV)) to the design of an entire mobility system leveraging such platforms.

In Section 1.1 we will explain what we mean by *automated design*. We will then present the desiderata and challenges for a co-design framework (Section 1.2), related work for the co-design of embodied intelligence (Section 1.3), and contributions of this thesis (Section 1.4).

1.1 What is (automated) "design"?

We take a broad view of what it means to "design", that is not limited to engineering. Citing Hebert Simon's *The sciences of the artificial* ([13], Chapter 5):

> *Engineers are not the only professional designers. Everyone designs who devises courses of action aimed at changing existing situations into preferred ones. The intellectual activity that produces material artifacts is no different fundamentally from the one that prescribes remedies for a sick patient or the one that devises a new sales plan for a company or a social welfare policy for a state. Design, so construed, is the core of all professional training; it is the principal mark that distinguishes the professions from the sciences. Schools of engineering, as well as schools of architecture, business, education, law, and medicine, are all centrally concerned with the process of design.*

The metaphors employed in this thesis are biased towards the engineering perspective. Indeed, it is easy for everybody to imagine creating a physical machine out of simple components, and to grasp the inherent decisions and trade-offs that must be navigated in such a process.

However, the theory to be discussed is applicable to other disciplines, if one takes a more abstract view of what is a system and a component. For example,

in urban transportation planning, the components are roads, mobility options, travel demand modeling, etc. In other disciplines, "components" can be logical instead of physical. For example, a public transit authority might ask how to design an incentive scheme such that such scheme (a "component") will move the system to a more desirable set of states (e.g., demand shift from private cars to public transit). In this context, another component could be a "simulator" or model of the "reaction" of the population to be analyzed.

Automated design

In this thesis, we are interested in methodologies which, given a complex system to create, facilitate and automate the design process, allowing the designer to smoothly navigate the complexity of the task. In particular, we want to:

- Structure the process of specifying a design problem, all the way from the architecture of the design, to the objectives and functions it has to fulfill;
- Automate the process of finding solutions to the design problem;
- Simplify the process of analyzing the obtained designs, and re-iterate, changing specifications.

In other words, we want a set of tools which, given a certain task specification, knowledge of principles from multiple domains, and a set of design options between which we need to choose, produces optimal designs, based on some performance metrics we care about (Fig. 1).

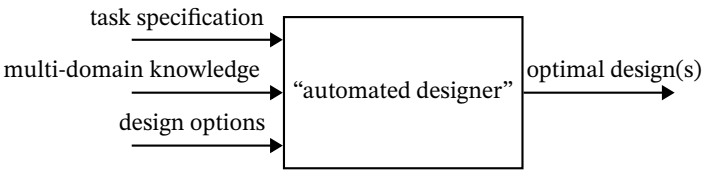


Figure 1: Vision for the "automated designer".

Let's look at a couple of examples to gain intuition.

Example 1.1 (Robotics)**.** In the context of co-designing a robot's autonomy stack, a typical scenario involves several key steps. First, there's the defined task, often a specific mission such as scanning an area for gas leakages. Then, one has the foundational principles governing robot autonomy, coupled with a thorough description of the robot's dynamics and its interactions with its sorroundings. Additionally, we are equipped with an assortment of components to pick from, which feature a range of choices including sensors, actuators, algorithms, and related parameters.

The essence of the design process lies in its ability to generate a set of optimal design decisions. Such desicions are guided by a set of performance metrics, tailored to the specific context. For instance, one might consider a trade-off between

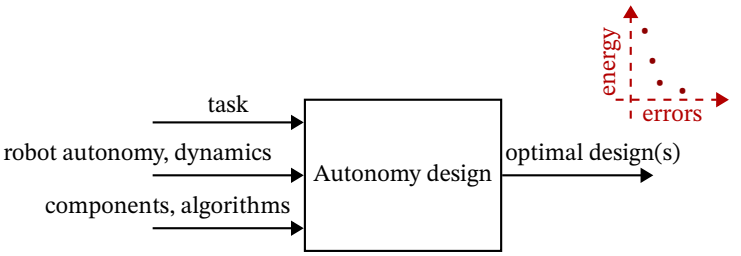


Figure 2: Vision for the "automated designer" in the context of autonomous systems.

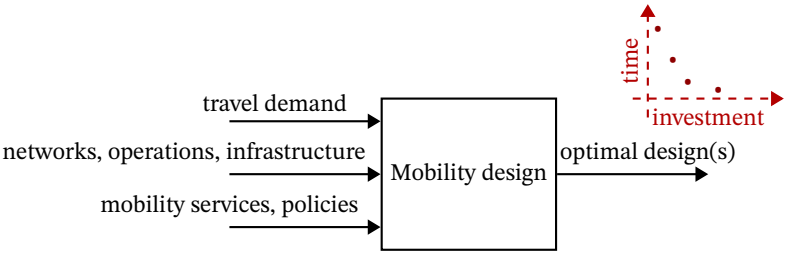


Figure 3: Vision for the "automated designer" in the context of mobility systems.

energy consumption and the cumulated errors while executing the prescribed task (Fig. 2).

Example 1.2 (Mobility)**.** Similarly, when designing a mobility system, we are given a certain demand (i.e., people at different locations, willing to move to certain destinations, with certain preferences for their trips), principles of network science, operations research, and infrastructure design, and a set of mobility services to design (e.g., fleet size, fleet distribution), or policies to be chosen (e.g., taxes on a certain mobility service). Again, the design process should produce a set of optimal design choices, in terms of of e.g., investments and average travel time one can reach in an entire city (Fig. 3).

1.2 Desiderata and challenges

How should the "automated designer" implement the vision detailed in Section 1.1? We will introduce the term "co-design", and describe it through the need for the design process to exhibit certain properties, listed in the following. Note that some are partially echoed in previous literature [14]–[16].

Formal Complex systems are made of heterogeneous components, and the abstraction one chooses to formulate the design exercise must transcend particular domains (i.e., it must be cross-domain [14]). For instance, when tasked with designing a fleet of AVs providing mobility services in a city, we require a toolbox to describe principles of autonomy, operations research, transportation policy, and economics. At the same time, to be tangible, the abstraction has to be mathematically precise, avoiding vague/controversial statements about the problem at hand. Typically, we also want to characterize all the objectives of the

design problem, without sacrifices.

Compositional, hierarchical This means a number of things. The first meaning, has to do with composition:

> co-design = design everything together

We use the word "co-design" to refer to any decision procedure that has to do with making simultaneous choices about the components of a system to achieve system-level goals. This includes the choice of components, the interconnection of components, and the configuration of components. We will see that in most cases, choices that are made at the level of components without looking at the entire system are doomed to be *suboptimal*.

Slightly modifying a quote from Howard Aiken[2]

> *A system is composed of components;*
> *a component is something you understand*,

we choose the following as our quote:

> *A system is composed of components;*
> *a component is something you understand* ***how to design***.

We refer to this first composition idea as *horizontal composition*.

The second meaning is given by the principle "your system is a component of somebody else's system". For the example of the fleet of AVs, one could think about designing a single AV, considering interconnected hardware components (actuators, sensors, computers) and software ones (control, perception, planning). When designing an entire mobility system, this interconnection of components can be considered a component, embedded into the larger set of interconnected components at the city level (e.g., including the design of a fleet of such vehicles, design of public transit, design of the network, etc.). We refer to this kind of composition as *vertical composition*.

Collaborative

> co-design = design everything, together

There are two types of collaborations. First, there is the collaboration between human and machine, in the definition and solution of design problems. Second, and most importantly, is the collaboration among different experts or teams of experts in the design process.

The typical situation is that the system design is suboptimal because every expert only knows one component and there are rigid interfaces/contracts designed

[2] Aiken was an american physicist and a pioneer in computing, being the original designer behind IBM's Harvard Mark I computer.

early on. The problem here is sharing of knowledge across teams, specifically, knowledge about the design of systems.

In this case, this is the slogan:

> *A system is composed of components;*
> *a component is something that* ***somebody*** *understands* ***how to design****.*

There is a tight link between the "composition" and "collaboration" aspects.

As Conway[3] first observed for software systems:

> *Organizations which design systems [...] are constrained to produce designs which are copies of the communication structures of these organizations.*

This "mirroring" hypothesis between system and organization was explored formally and found to hold [17]. The ultimate reason is that "the organization's governance structures, problem-solving routines and communication patterns constrain the space in which it searches for new solutions". This appears to be true for generic systems in addition to software.

In the end, civilization is about dividing up the work, and so we must choose where one's work ends and the other's work begins. But we need to keep talking if we want that everything works together.

Computationally tractable We need to be able to compute solutions to the design problems efficiently. Therefore, we strive to create not only a qualitative modeling for co-design, but also a formal and quantitative description that will be suitable for setting up an optimization problem that can be solved to obtain an optimal design.

Our slogan can be further modified:

> *A system is composed of components;*
> *a component is something that* ***somebody*** *understands* ***how to design well enough to teach a computer****.*

Continuous We look at designs not as something that exists as a single decision in time, but rather as something that continuously exists and evolves, independently from the designer. The designer should be able to smoothly characterize this evolution within a framework of co-design. In the literature, this is often referred to as *temporality* [18].

Manipulable Strictly connected to continuity, is manipulability. Not only we want the designer to be able to specify models for design problems, and to do

[3] John Horton Conway (1937–2020) was a mathematician. Probably the most popular idea of his was the invention of the Game of Life, which inspired countless works on cellular automata.

that over time, but we also want the whole problem "manipulation" process to be simple. For instance, if we might need to ignore certain objectives, or merge others (e.g., merging emissions and costs, via emissions penalties). Another important feature in this sense, is the ability to answer different questions, given the same co-design architecture. More on that in Chapter 3.

Intellectually tractable The design process and its formalisms should not be exclusively accessible to system architects, and specialists: they should be truly collaborative. Oftentimes, when developing design optimization tools, one confuses the *developer's* and the *user's* viewpoints. While we want the co-design formalism to possess the above properties (i.e., being formal, compositional, collaborative, computationally tractable, and continuous), we also want stakeholders of the design exercise to take an active role, and to be able to smoothly interact with the framework. This requires a simple, cross-domain user interface.

1.3 Related work

The literature on engineering design is very broad, and contributions stem from several fields. In this thesis, we will focus on (co-) designing embodied intelligence, ranging from a single platform (e.g., AVs) to mobility systems leveraging such platforms (e.g., within the context of AMoD systems). This kind of problems naturally draws inspiration from several fields, each contributing in various ways to the process of designing new technologies. For instance, in mechanical engineering, design optimization and design automation are part of the discipline of *formal engineering design*. In aeronautics and astronautics, instead, such methodologies fall under the umbrella of *systems engineering*. Notable reviews of the state of the art in these fields are [18]–[20].

Although they feature a vaste literature, these disciplines contributed to some megatrends, as well as to some general design methodologies, which we report briefly in the following.

In mechanical engineering, a large part of design optimization problems relate to topology optimization, which poses the question "how to employ a certain material within a design, to obtain certain structural performances?" See [21] for a survey and [22] for recent work in the area. Problems in this area, relevant for instance to soft robotics as well, are typically solved by either employing standard optimization techniques (e.g., convex optimization) or heuristic-based approaches. This discipline also gave birth to some general design techniques, such as the "A-design approach", by Campbell, Cagan, and Kotovsky [23], [24]. This agent-based methodology contrasts ad-hoc optimization problems formulations by introducing a user-friendly learning framework to input specific criteria in the design process. The framework allows for limited computation in the back-

ground, but it is intuitive on the surface, allowing the user to specify optimization criteria. Other general toolboxes are proposed as the "General design theory", by Yoshikawa [25], [26], and as the "Theory of Technical Systems", by Hubka and Eder [27]. The former framework is formal (based on set theory), and the latter is more conceptual. Finally, it is worth mentioning the "Axiomatic Design theory for Systems" from Suh [28], [29]. This theory provides a notable reference for functional decomposition of complex tasks, and features some sort of formality and computation, even if it remains at the conceptual level for many applications. We will discuss the shortcomings of this theory later in this thesis, in particular when compared to the presented theory of co-design.

In systems engineering, a lot of methodologies have been developed to organize architectures of complex systems. Several examples are provided in [18] and later in this thesis. Design optimization problems are typically approached from the point of view of multi-objective optimization. These are formulated in terms of classic problems (e.g., weighted convex optimization), ad-hoc problems solved with heuristics, and Pareto optimization [30]–[32]. In general, the formulations lack intuitivity and generalizability, and are very difficult to manipulate.

If we consider the fields of electrical engineering and computer science, as well as robotics, the literature on the co-design of complex systems is more prominent, and particular applications to embodied intelligence are more common.

Embodied intelligence

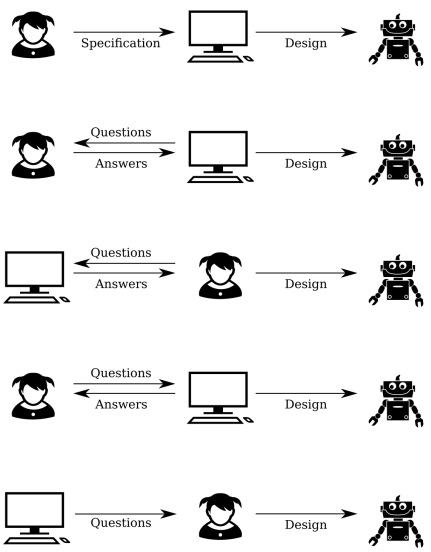


Figure 4: Information flows for computer-aided design of robots [16].

Designing embodied intelligence involves the choice of material parts, such as sensors, computing units, and actuators, and software components, including perception, planning, and control routines. In the last decade, research on autonomous systems has witnessed significant developments. While researchers have mostly focused on specific problems in robotics, there is glaring gap in our understanding of the optimal co-design of autonomous robots as a whole. Traditionally, the design of embodied intelligence has been approached in a compartmentalized manner, hindering interdisciplinary collaboration and design automation. In particular, such compartmentalization has treated the design optimization of components as separate entities, failing to capture the interconnections between *physical* and *software* components. In this context, interconnected designs have broad implications, ranging from power-constrained robot design to the design of AV fleets integrated into urban transportation systems.

Unresolved questions include: What is the simplest sensor that a robot can use to obtain a specific performance in a given task? How much computation is really needed? What control scheme should one choose to solve a specific task? What are the trade-offs between robot safety and task performance? How can one optimally design a robotic platform by minimizing resource usage?

The literature on design automation techniques and the associated challenges has been widely acknowledged and discussed in serveral sources [14]–[16], [33]–[38]. For instance, [14] identifies the unique characteristics of cyber-physical systems, including their hybrid nature (combining computational and physical elements), heterogeneity (including various types of components, different models), distributed networked structure, large-scale complexity, dynamic behavior, and human involvement. Similarly, [39] provides a comprehensive review of existing co-design methodologies for complex cyber-physical systems, and highlights synthesis challenges, such as functional complexity, intricate architectures, component interdependencies, conflicting system objectives, and the presence of humans in the design loop. Furthermore, Nilles et al. emphasize the importance of information flows in the design process (as reported in Fig. 4) and articulate four key challenges for modern robotics: *formalization*, *minimality*, *automation*, and *integration* [16].

The realm of embodied intelligence co-design primarily revolves around the exploration of trade-offs within robotics, encompassing (combinations of) aspects such as sensor and actuator selection, planning and control synthesis, and morphology and motion design.

Trade-offs within the field of robotics have been extensively studied [16], [34], [40]–[54]. [40] introduces a formulation for optimizing the design of serial manipulators, while [34], [53], [54] discuss the role of formal methods when synthesizing specifications and behaviors for autonomous robots performing complex tasks. Resource-performance trade-offs in mobile robotics are explored in [41], energy-efficient design techniques for legged robots are detailed in [42], and [43] delves into trade-offs between robot sensing and actuation for worst-case scenarios. Furthermore, [44], [45] provide insights into performance limits for robotics tasks and their relation to environmental complexity, while [55] examines information requirements for robot tasks through the concept of information invariants. Additionally, [50]–[52] offer a framework for interactive exploration of design trade-offs in manufacturing using CAD models.

In the context of co-designing autonomous systems, the literature predominantly focuses on sensing and control/actuation. While the sensor selection problem often lacks a closed-form solution, specific cases have been shown to exhibit sufficient structure for efficient optimization schemes [56]–[59] [60] investigates sensing-constrained task-driven LQG control, [46], [47], [61], [62] concentrate on perception architectures for AV navigation, and [63]–[65] explore large-scale sensor/actuator networks. Furthermore, while in [66] researchers provide a framework to jointly optimize sensor selection and control, by minimizing the information a sensor needs to acquire, authors of [67], [68] propose techniques for optimal control with communication constraints, and [69] studies a hierarchical multi-rate control architecture for actuation and planning. An

intriguing line of work led by Shell, O'Kane, and their collaborators explores the design of minimal robots capable of solving planning problems, studying various trade-offs in sensing, actuation, and path planning [70]–[76]. The authors of [68] and [77] introduce approaches for the co-design of control algorithms and platforms, applied to lane keeping and HVAC systems, respectively. [39], [78] focus on performance and security of cyber-physical systems, [79] investigates co-learning of task and sensor placement for soft robotics, and [80] studies computation-communication trade-offs and sensor selection

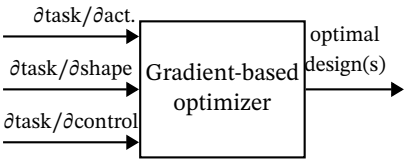


Figure 5: Working principle of gradient-based design co-optimization based on differentiable simulation.

Another prominent line of research stemming from computer graphics, robotics, and computer science, centers around morphology and actuation/motion design for (soft, articulated) robots. Researchers frequently leverage gradient-based optimization algorithms informed by differentiable simulations, typically considering material parameters, shapes, and actuation as influential factors [81]–[83]. These algorithms leverage the differentiable simulator's ability to provide rapid, differentiable relationships between design parameters changes and task performance, as demonstrated in scenarios such as robot swimming [82]. The nature of these simulators allows for the incorporation of such structural insights into gradient-based optimization algorithms, as illustrated in Fig. 5. Similar concepts have been applied to the design of AV architectures [84]. Within this line of research, authors have developed methodologies to design (soft) robots based on multiple objectives in computationally efficient manners [85]–[90], and created end-to-end frameworks, capable of starting from functional specifications and creating new robots from scratch, leveraging formal methods [91]–[93] and evolutionary algorithms [94]–[97]. Furthermore, researchers were able formulate ad-hoc optimization problems for the computational design of robots performing complex tasks [98]–[108].

Another notable research effort revolves around assume-guarantee contracts, and design techniques leveraging them (see [37], [109] for excellent explanations). Such contracts are expressed as formal specifications attached to components in a system, stating what is assumed from the environment, and what is required from the component in case the environment meets the specifications. In the context of designing a complex system, contracts can be embedded in the techniques of platform-based design [110]–[115], contract-based design [109], [116], [117], and compositional behavior generation and verification [118]–[121]. All of these methods have been successfully applied to a wide range of systems, ranging from AV to HVAC systems [77].

Summary

None of the methods described above is driven by the purpose of automating the design of an entire system, as described in Section 1.1. Rather, the developed tools

focus on particular aspects of that purpose, partially addressing the challenges reported in Section 1.2. To visualize this better, we can categorize the methods in the following groups:

- General design techniques, which are mostly conceptual. Let's denote them by (G).
- General design techniques, which are also quite quantitative, such as the "Axiomatic Design Theory", the "General Design Theory", or the "A-design approach" [23], [26], [29]. Let's denote them by (Q);
- Ad-hoc (multi-objective) optimization problems, with classic structure and solution techniques. Typically they require some strong assumptions to fit the problem to the particular structure at hand. Let's denote them by (H);
- Ad-hoc (multi-objective) "exotic" optimization problems, often requiring heuristics/evolutionary approaches to be solved. Typically, there are no guarantees that a solution can be found. Let's denote them by (M);
- Gradient-based methods employed in the context of system design, leveraging differentiable simulation tools. Let's denote them by (D);
- End-to-end approaches, denoted by (E);
- Methods based on contract theory, such as platform-based and contract-based design [109], [110] Let's denote them by (C);

In the spirit of analyzing trade-offs (more on that in Chapter 2), we now want to compare the aforeintroduced macro categories based on some representative desiderata introduced in Section 1.2. We report them in Fig. 6 to Fig. 8.

Intellectual vs. computational tractability When designing complex systems, there exists a significant trade-off to consider: the balance between intellectual tractability (i.e., how easy it is to work with the framework and formulate and solve design problems) and computational tractability (i.e., the efficiency with which the framework can actually address the problem, and the complexities it can handle). Let's explore the extremes in both directions.

On one hand of the spectrum, we have historical and general design techniques which possess inherent intellectual tractability due to their conceptual nature. If these methods involve quantitative aspects, they rely on straightforward principles which are easy to grasp (e.g., minimizing the number of joints in a mechanical system). However, when applied to the type of problems associated with embodied intelligence, these frameworks tend to impose rigid structures that are ill-suited for these challenges both formally and computationally. Often, these methods focus solely on feasibility, rather than minimality, and they don't involve optimization.

On the other end of the spectrum, some aspects of co-design problems are framed

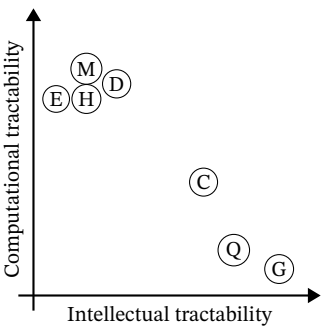


Figure 6: Intellectual vs. computational tractability.

as standard optimization problems (e.g., sensor selection through convex optimization [56]), which offer excellent computational properties. However, this computational efficiency comes at the cost of reduced intellectual tractability, and ease of manipulation. Designers may find themselves constrained in defining performance metrics (e.g., they must adhere to properties like differentiability and convexity), express constraints in non-intuitive ways through various reformulations, and require different tools when altering the problem's parameters.

In the context of co-designing embodied intelligence, end-to-end and heuristic-driven approaches tend to be less intellectually tractable, often excluding the designer from the process. In contrast, differential simulation-driven methods provide more flexibility, while remaining computationally tractable, particularly for simplified instances of the problem.

Lastly, methods based on contract theory aim to strike a balance between both worlds. They maintain intellectual tractability (indeed, understanding set-based contracts is relatively simple) while offering computational efficiency. It's important to note that these methods prioritize feasibility and verification, rather than looking for minimal solutions. Furthermore, the tools around them are often focused on particular types of systems (e.g., embedded systems, dynamical systems, etc.).

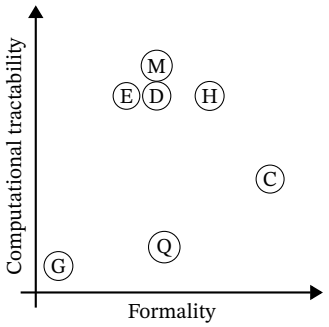


Figure 7: Formality vs. computational tractability.

Formality vs. computational tractability Another significant trade-off to consider relates to the balance between formality and computational tractability. Often, when tackling a design problem, a specific technique is applied, but the process of using it and how to extend its application to future scenarios lacks a formal structure.

For instance, one might design a component in the autonomy stack of a robot by fixing various parameters, defining a reward function, and exploring the design space using a specific heuristic. While this approach may yield interesting solutions, it often falls short in providing insights into their quality, dependencies with other parameters, and how these solutions change when assumptions are slightly adjusted. In such cases, conceptual design frameworks prove to be lacking in both formalism and computational tractability, making them less suitable for such types of problems.

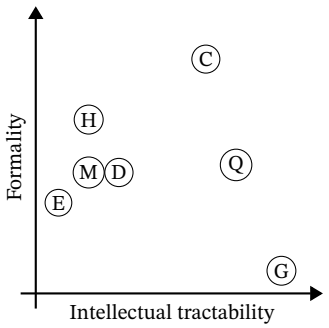


Figure 8: Intellectual tractability vs. formality.

In contrast, contract-based methods offer a high level of generalizability and can be applied across various applications, as evident in the existing literature [37]. On the other hand, standard (multi-objective) optimization techniques have solid formal foundations but often lack formalism and generalizability when employed in design optimization tasks.

Formality vs. intellectual tractability Finally, combining the previous plots one can investigate the trade-off between formality and intellectual tractability. Typically, intellectual tractability comes at the cost of having a framework which is not really formal. This is true in the analyzed intuitive and at times quantitative general design methods. On the other hand, optimization schemes which can be well formalized, are not intuitive to use in the context of co-design, and interfaces might be complicated. Contract-based methods find a good balance between the two desiderata, being both formally clear, but also understandable and manipulable.

All in all, the previous observations highlight a gap in the research of engineering design methods. In particular, a toolbox which is at the same time formal (hence, generalizable to multiple disciplines), computationally tractable, but also easy to use, is missing. In this thesis we will present a monotone theory of co-design, which will address such concerns.

1.4 Outline and contributions

In this thesis, our focus revolves around the intricate process of co-designing comlpex systems. We embark on this journey by delving into the theoretical underpinnings of a monotone theory of co-design, a theory initially developed by Censi [122]. Our objective is to provide a comprehensive understanding of this concept by placing it within a rigorous mathematical framework, complete with illustrative examples elucidating each key concept.

To navigate the features of this framework, we draw upon the tools of category theory, offering insights into features such as compositionality and the application of functorial solution schemes to co-design problems. This approach serves as a powerful lens through which we explore the world of co-design.

The thesis not only equips the reader with a deep comprehension of the theoretical foundation, but also offers a practical roadmap for applying the monotone theory of co-design to a diverse array of real-world problems. These span a wide spectrum, ranging from the co-design of an AV autonomy stack to the task of designing the infrastructure for an intermodal mobility system.

This first-of-a-kind toolbox unifies disparate disciplines under a single modeling framework. It is designed to facilitate the efficient computation of optimal design solutions tailored to specific tasks. Its novelty not only enhances our understanding of co-design processes, but also paves the way for a multitude of future research endeavors, promising to expand the horizons of this field.

Organization of the manuscript

The manuscript is subdivided in three parts. First, we introduce a mathematical theory of co-design (Part A). To do so, we get the reader up to speed with a background on orders and monotonicity, and explain the basics of design problems, providing theoretical and practical examples for all concepts introduced. We then show how one can interconnect different design problems, and provide a formal explanation of why one can do so, via category theory. Finally, we show how to solve co-design problems.

In the second part (Part B), we propose a systematic process for the co-design of complex systems, providing a recipe for the employment of the presented theory. We then show how the proposed framework can deal with co-design problems all the way from a single platform (i.e., an AV) to an entire mobility system leveraging that platform.

The final part features interesting venues for future research, as well as conclusions (Part C).

Hacks to read this thesis

Table 1: Use of colors

sets	$\mathbf{A}, \mathbf{B}$
posets	$\mathbf{P}, \mathbf{Q}$
categories	$\mathbf{C}, \mathbf{D}$
objects	X, Y
morphisms	$f : X \to Y$
functors	$F : \mathbf{C} \to \mathbf{D}$

Use of colors

- We **use colors** to aid in the parsing of formulas and diagrams (Table 1). We also color the operations between these elements. In this way it is easy to see the types at first glance: $f \fatsemi g$, $F \fatsemi G$, etc.
- Color is *not* necessary to infer meaning. The choice of colors is **colorblind-friendly** for red-green color blindness. (The author of this thesis is red-green color blind.) Please let me know if this is not the case.

Proofs To facilitate the reading, most proofs are reported in Chapter 16. Each time this is the case, there is a hyperlink which allows you to navigate to the proof, and then back to the main text.

Sequent notation We will often employ the sequent notation to represent implications and equivalences. For instance, the statement "if A and B, then C" ($A, B \Rightarrow C$) is represented as

$$\frac{A \quad B}{C}.$$

Furthermore, the statement "A if and only if B" ($A \Leftrightarrow B$) is represented as

$$\frac{A}{B}.$$

Book Part of the technical notions are adapted from our work-in-progress book [123]. We will provide related references whenever it is the case.

A Monotone Theory of Co-design | Part A

Background on orders and monotonicity

2

There are no solutions, only trade-offs.
—Thomas Sowell[4]

Engineering is about trade-offs, and when designing a system, there is seldom a best outcome out of all quantities of interest. In this chapter, we introduce concepts to reason about uncomparable attributes. Specifically, we first present the notion of trade-offs (Section 2.1), and make it more precise by presenting partially ordered sets (posets) (Section 2.2). Posets are the mathematical structure to reason about trade-offs, and will be crucial for the understanding of the mathematical theory of co-design we introduce in this part. We will then provide a broad list of examples (Section 2.3), as well as tools to construct new posets starting from old (Section 2.5). Finally, we will present the concepts of monotonicity (Section 2.6), poset bounds (Section 2.7), and lattices (Section 2.8).

2.1 Trade-offs

Trade-offs characterize all engineering disciplines, and can be literally found everywhere.[5]

For instance, during your morning commute, you might be interested in the cost of your trip, and the time needed to get to your destination. Typically, getting a taxi will be more expensive than choosing public transit, but it will generally result in a quicker trip. This is a trade-off. The role of trade-offs in engineering has been widely discussed in the literature.

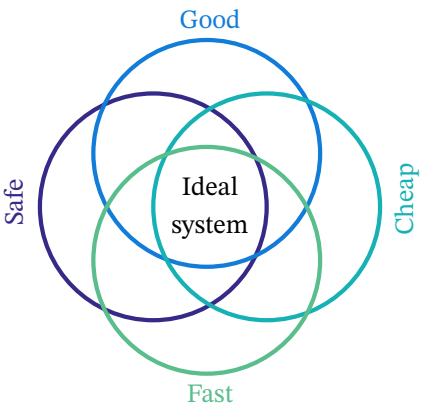
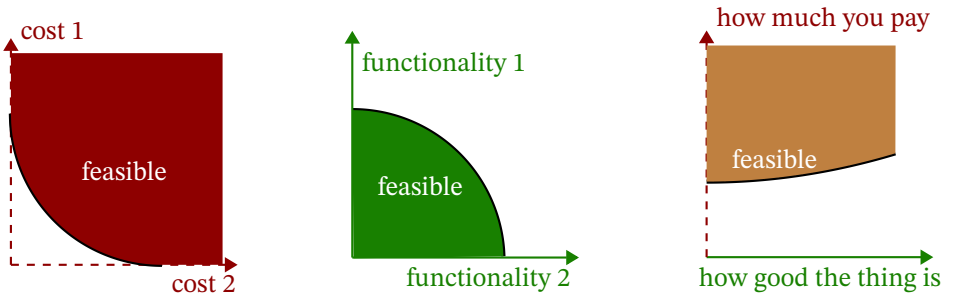


Figure 9: The four quantities of system architects from Rechtin and Maier [124].

Example 2.1 (The Art of Systems Architecting)**.** Rechtin and Maier, in their "Art of Systems Architecting" [124], state that the process of designing any system creates trade-offs between four fundamental quantities: performance (e.g., quality of a product), schedule (e.g., time needed to produce it), cost, and risk (e.g., system's probability of failure). Balancing the conflicts between performance, schedule, and cost, is typically the business of project managers. Usually, core engineers deal with trade-offs in performance and risk. In general, if you want to build something, an ideal system is done well, quickly, is safe, and is cheap. However, in reality, you typically need to trade-off some of these quantities (Fig. 9).

Example 2.2 (The "-ILITIES")**.** De Weck and co-authors present common desired properties of systems one has to trade-off as the "ilities" [18, Chapter 4]. These include quality, reliability, safety, flexibility, robustness, durability, scalability,

[4] Sowell is an american economist, recipient of the National Humanities Medal, and author of over 45 books.
[5] Everywhere in this thesis, everywhere in your field, and everywhere in your daily life.



(a) Costs. **(b)** Functionalities. **(c)** Functionality vs. costs.

Figure 11: Achievable accuracy plots.

adaptability, usability, interoperability, sustainability, maintainability, testability, modularity, resilience, extensibility, agility, manufacturability, repairability, evolvability.

To characterize engineering trade-offs, we will use the mathematical structure of partial orders. But first, let's explore some examples, to better contextualize trade-offs.

Functionality and resources

In this section, we introduce concepts which will be important when talking about theories of co-design. We distinguish semantically between functionalities and requirements/costs. In general, you prefer functionalities to be "large" (Fig. 10b) and requirements/costs to be "small" (Fig. 10a).

(a) Requirements/costs.

(b) Functionality.

Figure 10: Illustration of functionalities and resources.

Throughout this thesis, we will mainly use three kinds of achievable accuracy plots (Fig. 11):

- In Fig. 11a we plot trade-offs in costs and add a "feasibility" curve. Everything above this curve is feasible and will cost more than what is *on* the curve.
- In Fig. 11b we plot trade-offs in functionalities and add a "feasibility" curve. Everything below the curve is feasible, but is below the "standards" required by the curve.
- In Fig. 11c we plot functionality and resource together, representing the trade-off between "how good a product is" and "how much one needs to pay for it". Feasible pairs are represented via the feasibility curve. Everything above the curve will be feasible (by paying more).

It is a good exercise to open any engineering book, find the graphs talking about "achievable" performance and "resources" needed, and classify them into one of the ones reported in Fig. 11. Note that functionalities and requirements are not absolute, and depend from the context.

Trade-offs for the human body

The human body is a great example of trade-offs and adaptability. Consider sports: when looking at different disciplines, various physical abilities are desired and athletes are characterized by trade-offs between them.

For instance, we can think about trade-offs between speed and endurance for humans (Fig. 12). These are functionalities, which different athletes might want to maximize. Consider Usain Bolt, who owns the 100 meters, 200 meters, and 4×100 meters relay world records. Without doubts, in the human speed-endurance trade-off curve he positions himself close to the highest achievable speeds. At the same time, however, Usain Bolt is not among the men with the best endurance in the world. To see the other end of the curve, we need to introduce Eliud Kipchoge, twice Olymptic marathon champion. Similarly to Bolt, he is among the best in his discipline, reaching very high endurance. Again, the speed-endurance trade-off implies that Kipchoge cannot be among the fastest men in the world, if he wants to be among the ones with best endurance. For reference, on the bottom left side of the plot, it's me.

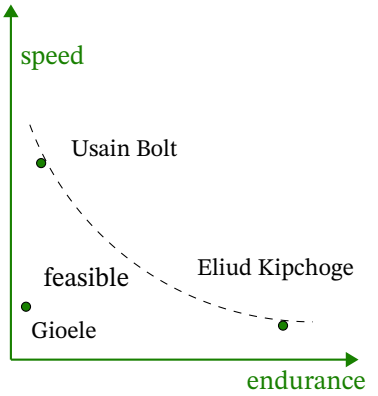


Figure 12: Speed vs. strength trade-off in sports.

In this case, the resource needed to obtain speed or endurance is the amount of training (Fig. 13). If we want to relate the invested training and the resulting endurance reached by the athletes, we will notice that with a lot of training, Kipchoge will improve his results, approaching perfection. On the other hand, the kind of training Bolt undergoes is not optimizing endurance, and therefore his results will be less effective towards maximizing endurance.

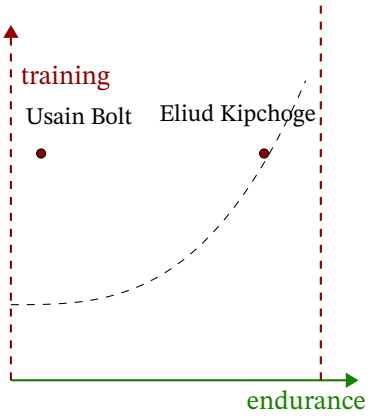


Figure 13: Training vs. strength trade-off.

Protective masks

Orders give us a rich way to describe designs under various lenses. Recently, we all needed to become experts of protective masks. In this section, we will show various ways in which we can order the latter by functionality.

By first thinking about the effectiveness of the mask in protecting the wearer from a virus, we can order masks as in Fig. 14. In general masks are classified following their filter abilities and inward leakages. The FFP1 class filters at least 80 % of airborne particles and allows less than 22 % inward leakage. The FFP2 class filters at least 96 % of airborne particles and allows less than 8 % inward leakage, and the FFP3 class filters at least 99 % of airborne particles and allows less than 2 % inward leakage.

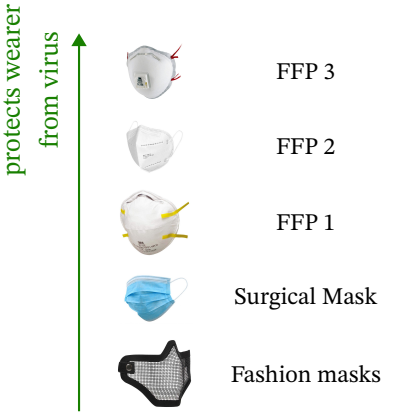


Figure 14: Ordering masks by protection levels.

Obviously, based on the protection level, the most performant in Fig. 14 is FFP3, and the worst is the fashion one. However, this is not the only way in which we can classify masks. If, for instance, we want to consider a functionality "how much does the mask say about the wearer", we can order the masks differently. Arguably, the ordering could look like the one in Fig. 15a.

Indeed, choosing a fashion mask might say that the wearer cares more about aesthetics than safety, and choosing a FFP3 highlights responsible behaviors, care, and research in masks models.

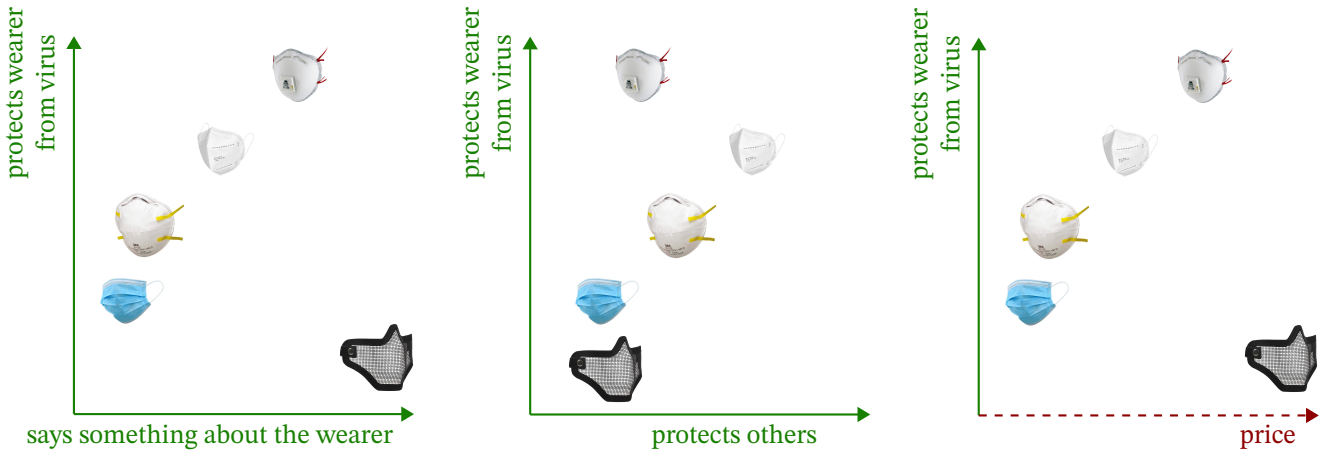
Similarly, we could order masks based on different performance criteria, adding the functionality "how much does it protect others?" (Fig. 15b).

On the other hand, we could think about the trade-offs between the mask performance and its cost, presenting a functionality-resource plot (Fig. 15c).

More performant masks are typically more expensive, and the fashion mask will probably have the least performance and most expensive.

This example once again highlights the flexibility and richness of the "orders approach".

By considering all the aforementioned characteristics together no product dominates another. This is the *law of successful products*. At equilibrium, in an efficient and free market, no product completely dominates another by both functionality and requirements. Otherwise, the dominated product would not sell. Once we *specify the design purpose* and the related constraints, we can (partially) order products.



(a) Self-protection vs. expressitivity.

(b) Self-protection vs. protection of others.

(c) Self-protection vs. price.

Figure 15: Ordering masks by other considerations.

2.2 Ordered sets

So far, the discussion has been purely qualitative. We would like to be able to formally describe preferences/priorities over a number of functionalities and resources. For instance, in the context of a morning commute, one might prefer the cost of the ride over the time needed to reach a destination, and the number

of mode changes (e.g., bus and tram) involved in the trip, and that one might not prefer time over mode changes, or mode changes over time.

In this section, we present the mathematical structures to quantitatively reason about these trade-offs. We introduce pre-orders, partial orders, and total orders. While we report the most important concepts in this text, Davey and Priestley [125] and Roman [126] are excellent reference texts for the subject. This part builds on the notion of relations and their properties, which are covered in our book [123].

We introduce these concepts by adding levels of specificity.

Pre-orders

A pre-order is a set together with a reflexive and transitive relation.

Definition 2.3 (Pre-ordered set)
A *pre-ordered set* is a tuple $\mathbf{P} = \langle \mathbf{P}, \preceq_{\mathbf{P}} \rangle$, where $\mathbf{P}$ is a set, called the *carrier set* or *underlying set*, together with a relation $\preceq_{\mathbf{P}}$ that is reflexive and transitive.

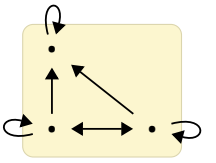


Figure 16: A pre-order represented as a graph.

An example of a pre-ordered set represented as a graph is shown in Fig. 16. In the graph representation of a pre-order $\mathbf{P}$, we draw an arrow between x and y if $x \preceq_{\mathbf{P}} y$.

Example 2.4. The reachability relationship in any directed graph (potentially including cycles) is a pre-order. The pre-order $\mathbf{P}$ is defined as follows. The set $\mathbf{P}$ is the set of nodes of the graph. Take any two nodes $x, y \in \mathbf{P}$. One has $x \preceq_{\mathbf{P}} y$ if and only if there is a path from x to y in the directed graph. There is always a path from a node to itself (reflexivity), and given a path from x to y, and one from y to z, we know that there is a path from x to z (transitivity).

Partial orders

By adding the condition of *antisymmetry* to a pre-order, we obtain a partially-ordered set.

Definition 2.5 (Partially ordered set)
A pre-ordered set $\mathbf{P} = \langle \mathbf{P}, \preceq_{\mathbf{P}} \rangle$ is a *partially-ordered set* (poset) if the relation $\preceq_{\mathbf{P}}$ is antisymmetric.

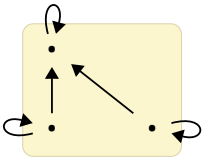


Figure 17: A partial order represented as a graph.

An example of a poset represented as a graph is shown in Fig. 17. By comparing this with Fig. 16, we notice that the double-headed arrow is not allowed anymore (indeed, its existence would imply that source and target of the arrow are the same element in the poset).

Example 2.6. The reachability relationship in any directed graph does not define a poset. As a simple counterexample, take a graph with nodes $\{x, y, z\}$ and paths x to y, y to z, and z to x. From transitivity, one has $x \preceq z$, but from reachability we also have $z \preceq x$. Therefore, per antisymmetry one should have $x = z$, but these are actually distinct nodes. To make things work, one needs to consider only *acyclic* graphs.

Example 2.7. The following defines a partial order $\preceq$ on the set of natural numbers $\mathbb{N}$. Define, for all $x, y \in \mathbb{N}$,

$$x \preceq y \quad \text{if, and only if} \quad x \text{ divides } y.$$

By definition, a natural number x divides another natural number y if there exists some natural number z such that $xz = y$. The notation for "x divides y" is $x|y$.

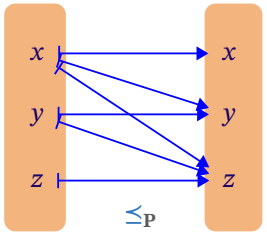
Figure 18: A total order.

Total order

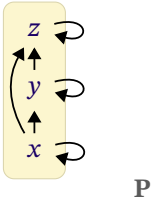
Definition 2.8 (Totally ordered set)
A partially ordered set $\mathbf{P} = \langle \mathbf{P}, \preceq_{\mathbf{P}} \rangle$ is a *totally ordered set* if the relation $\preceq_{\mathbf{P}}$ is total.

An example of a totally ordered set represented as a graph is reported in Fig. 18.

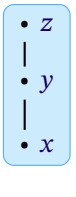
Example 2.9 (Reals)**.** The real numbers $\mathbb{R}$ form a totally ordered poset $\langle \mathbb{R}, \leq \rangle$ with order relation given by the usual ordering.



(a) $\mathbf{P}$ as a relation.



(b) $\mathbf{P}$ as a graph.



(c) $\mathbf{P}$ as a Hasse diagram.

Figure 19: Three different representations for a poset.

Hasse diagrams

We can represent partial orders in various ways. We now take a proxy poset and represent it using different conventions. Consider $\mathbf{P} = \langle \mathbf{P}, \preceq_{\mathbf{P}} \rangle$, where $\mathbf{P} = \{x, y, z\}$ and $x \preceq_{\mathbf{P}} y$, $y \preceq_{\mathbf{P}} z$. First, we could represent this using the standard visualization for relations (Fig. 19a).

However, this is quite heavy, and does not exploit the fact that partial orders are *endorelations*. Therefore, we could think to only draw the carrier set once (Fig. 19b).

However, both the reflexivity arrows and the arrow from x to z is implicit in partial orders, because of transitivity.

A Hasse[6] diagram is an economical (in terms of arrows) way to visualize a poset. In a Hasse diagram elements are points, and if $x \preceq_{\mathbf{P}} y$ then x is drawn lower than y and with an edge connected to it, if no other point is in between (Fig. 19c). Hasse diagrams are directed graphs.

[6] Named after the German mathematician Helmut Hasse, who according to Garret Birkhoff (an American mathematician working on lattice theory), made efficient use of them.

Example 2.10 (Discrete posets)**.** We can see every set $\mathbf{P}$ as a *discrete poset* $\mathbf{P} = \langle \mathbf{P}, = \rangle$ using equality as the partial order. Notice that equality is symmetric, transitive, and antisymmetric. When visualized as a Hasse diagram, discrete posets are a collection of points.

Example 2.11 (**Bool**)**.** The set of booleans $\mathbf{Bool} = \{\bot, \top\}$ can be made into a poset by choosing the order $\bot \preceq_{\mathbf{Bool}} \top$. This is equivalent to using "$\Rightarrow$" as a relation. We obtain the poset $\mathbf{Bool} := \langle \mathbf{Bool}, \Rightarrow \rangle$.

2.3 Examples of posets

Power poset

A classic poset is the one on the power set of a set (i.e., the set of subsets of the set). There is a natural order on subsets, given by set inclusion.

Definition 2.12 (Power poset)
Given a set $\mathbf{A}$, define the *power poset* $\mathsf{Pow}\,\mathbf{A} = \langle \mathsf{Pow}\,\mathbf{A}, \subseteq \rangle$ by ordering the subsets in its power set $\mathsf{Pow}\,\mathbf{A}$ by inclusion.

A subset $\mathbf{S}$ precedes $\mathbf{T}$ if $\mathbf{S} \subseteq \mathbf{T}$:

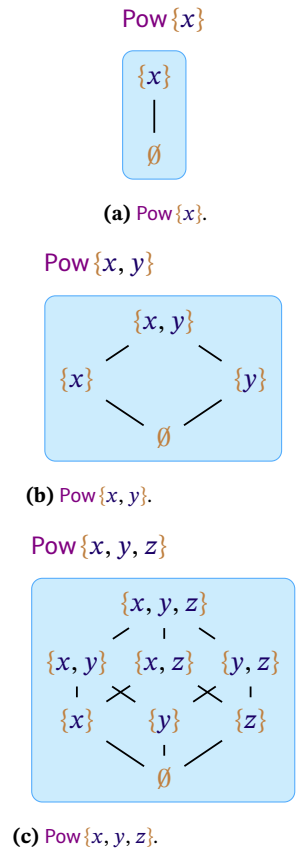
$$\frac{\mathbf{S} \preceq_{\mathsf{Pow}\,\mathbf{A}} \mathbf{T}}{\mathbf{S} \subseteq \mathbf{T}}.$$

This is illustrated in Fig. 20 for sets of 1, 2, 3 elements. One can formally check that the power poset is a set. Consider a set $\mathbf{A}$. Clearly, given $\mathbf{S} \in \mathsf{Pow}\,\mathbf{A}$, we have $\mathbf{S} \subseteq \mathbf{S}$. Furthermore, given also $\mathbf{T} \in \mathsf{Pow}\,\mathbf{A}$, we have

$$\frac{\mathbf{S} \subseteq \mathbf{T} \quad \mathbf{T} \subseteq \mathbf{S}}{\mathbf{S} = \mathbf{T}}.$$

Finally, given also $\mathbf{U} \in \mathsf{Pow}\,\mathbf{A}$, we have

$$\frac{\mathbf{S} \subseteq \mathbf{T} \quad \mathbf{T} \subseteq \mathbf{U}}{\mathbf{S} \subseteq \mathbf{U}}.$$



(a) $\mathsf{Pow}\{x\}$.

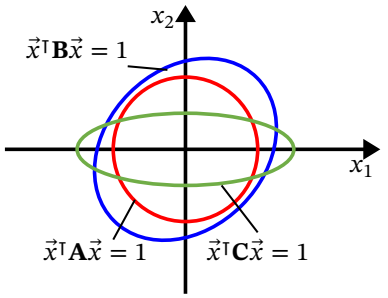
(b) $\mathsf{Pow}\{x, y\}$.

(c) $\mathsf{Pow}\{x, y, z\}$.

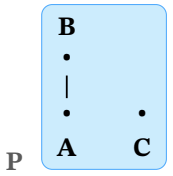
Figure 20: Power set as a poset.

Positive definite matrices

Definition 2.13 (Positive (semi-) definite matrix)
A symmetric matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$ is *positive definite* if $\boldsymbol{x}^\intercal \mathbf{M} \boldsymbol{x} > 0$ for all non-zero $\boldsymbol{x} \in \mathbb{R}^n$. It is *positive semi-definite* if $\boldsymbol{x}^\intercal \mathbf{M} \boldsymbol{x} \geq 0$ for all $\boldsymbol{x} \in \mathbb{R}^n$. We call the set of all positive definite matrices $\mathsf{PDM}(n)$, and the one of all positive



(a) Ellipses representing positive definite matrices.



(b) Order between positive definite matrices.

Figure 21: Poset of positive (semi-) definite matrices.

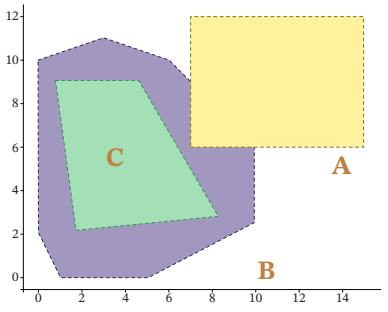


Figure 22: Three different polyhedra.

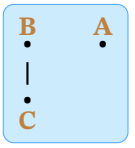


Figure 23: Poset of polyhedra.

semi-definite matrices $\mathsf{PSDM}(n)$.

Positive definite matrices have real, positive eigenvalues, which can be interpreted as axes lengths of ellipsoids. Any matrix $\mathbf{A} \in \mathsf{PDM}(n)$ describes an ellipsoid, which can be written as a quadratic equation:

$$\boldsymbol{x}^\intercal \mathbf{A}\boldsymbol{x} = 1, \quad \boldsymbol{x} \in \mathbb{R}^n.$$

We can define a partial order on $\mathsf{PDM}(n)$ as

$$\frac{\mathbf{A} \preceq_{\mathsf{PDM}(n)} \mathbf{B}}{\boldsymbol{x}^\intercal \mathbf{A}\boldsymbol{x} \leq \boldsymbol{x}^\intercal \mathbf{B}\boldsymbol{x} \quad \forall \boldsymbol{x} \in \mathbb{R}^n}.$$

This can be interpreted as an inclusion of ellipsoids. Take for instance the matrices

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 3/4 & -1/8 \\ -1/8 & 3/4 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1/2 & 0 \\ 0 & 2 \end{bmatrix}.$$

The order $\mathbf{P}$ on the set $\{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$ is reported in Fig. 21b, and it is easily explained via Fig. 21a. The ellipse representing $\mathbf{A}$ (in red) is included by the one representing matrix $\mathbf{B}$ (in blue), but not by the one representing matrix $\mathbf{C}$ (in green). Furthermore, the one representing $\mathbf{B}$ includes the one representing $\mathbf{C}$.

Convex sets

Ordering ellipsoids via inclusion can be made more precise, by defining posets of convex sets.

Definition 2.14 (Convex set)
A set $\mathbf{A} \subseteq \mathbb{R}^n$ is *convex* if, for every $x, y \in \mathbf{A}$, and $\theta \in [0, 1]$:

$$\theta \boldsymbol{a} + (1 - \theta)\boldsymbol{b} \in \mathbf{A}.$$

Examples of convex sets include polyhedra, affine spaces, ellipsoids, etc. We can create a poset of convex sets, by taking the set of all convex sets $\mathbf{C} \subseteq \mathbb{R}^n$ as the carrier set, and the inclusion $\subseteq$ as the order. Let's look at a particular example when considering polyhedra, i.e., sets of the form

$$\mathbf{A} = \{\boldsymbol{x} \in \mathbb{R}^n : \mathbf{A}\boldsymbol{x} \preceq^m \boldsymbol{y}, \mathbf{A} \in \mathbb{R}^{m \times n}, \boldsymbol{y} \in \mathbb{R}^m\},$$

where $\preceq^m$ is the vector ordering (point wise) on m-dimensional vectors.

To visualize this kind of orders, consider for instance three different polyhedra (Fig. 22), and the order they create (Fig. 23).

Sensor/algorithm pairs

Another nice example of partial orders is related to the need of ordering sensor-algorithms pairs (by preference) in the context of autonomous systems design (e.g., an AV). In general, a robot is equipped with sensors, which produce observations from which one can detect obstacles (through algorithms) in the scene to be explored. It has already been observed that sensors can be ordered by their ability to discriminate states of the robot [127]. Here, instead, we characterize sensors-algorithm pairs by the so-called "sensing performance" curves, expressed in terms of a false positives map $\mathrm{FP} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{[0,1]}$ (i.e., given an environment, the probability of detecting an obstacle at distance d, if there is no obstacle), a false negatives map $\mathrm{FN} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{[0,1]}$ (i.e., assuming the presence of an obstacle at distance d, the probability of not detecting it), and an accuracy map $\mathrm{ACC} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, denoting the sensing accuracy (range) as a function of distance from the obstacles. These curves can be obtained through various sensor benchmarking techniques, which we detail in [128] and references therein. As an example, we report some of these maps in Fig. 24, for different sensor-algorithm pairs.

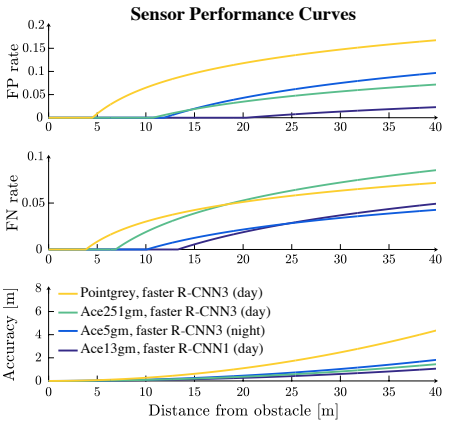


Figure 24: Sensor performance curves, in terms of false positives, true positives, and accuracy rates.

To order these, one can leverage a poset of maps, assessing the point-wise dominance of functions.

Definition 2.15 (Poset of maps)
Consider posets $\mathbf{P}, \mathbf{Q}$, and consider the set of functions $\mathbf{P} \to \mathbf{Q}$, denoted by $\mathbf{Q}^{\mathbf{P}}$. Given any two functions $f, g : \mathbf{P} \to \mathbf{Q}$, we define

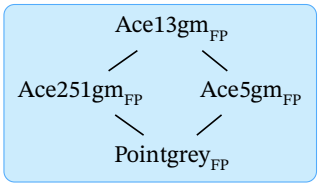
$$f \preceq_{\mathbf{Q}^{\mathbf{P}}} g \Leftrightarrow f(p) \preceq_{\mathbf{Q}} g(p), \quad \forall p \in \mathbf{P}.$$

Lemma 2.16. Def. 2.15 indeed defines a poset.

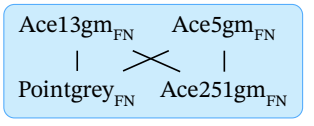
See proof on page 203.

Given this poset, we can order the curves presented in Fig. 24 for each of the three quantities of interest, false positives, false negatives, and accuracy (Fig. 25)[7]. For instance, when looking at false positives, the camera Pointgrey is dominated by Ace5gm, Ace251gm, and, by transitivity, by Ace13gm. Instead, Ace251gm and Ace5gm are not comparable.

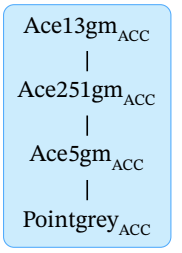
Interestingly, the posets are different for the three quantities, meaning that to make a decision on which sensor-algorithm pair is the best one, one will have to choose a particular way of combining posets into a new one (more on that in Section 2.5).



(a) False positives.



(b) False negatives.



(c) Accuracy.

Figure 25: Hasse diagrams for the sensor performance curves.

[7] For simplicity, we refer here only to the sensor names, omitting the algorithm names, since we have four different sensors.

2.4 Chains and Antichains

There are two special types of subsets of a poset: chains and antichains. Their definitions are dual.

Definition 2.17 (Chain and antichain in a poset)
Given a poset $\mathbf{P} = \langle \mathbf{P}, \leq_{\mathbf{P}} \rangle$, a *chain* is a subset $\mathbf{S} \subseteq \mathbf{P}$ such that any two elements of $\mathbf{S}$ are comparable:

$$\frac{x, y \in \mathbf{S}}{(x \leq_{\mathbf{P}} y) \vee (y \leq_{\mathbf{P}} x)}.$$

An *antichain* is a subset $\mathbf{S}$ of a poset where *no* two distinct elements are comparable:

$$\frac{x, y \in \mathbf{S} \quad x \leq_{\mathbf{P}} y}{x = y}.$$

We denote the set of antichains of a poset $\mathbf{P}$ by $\mathsf{Anti}\,\mathbf{P}$.

Example 2.18 (Chains and antichains in a power poset)**.** Consider the power poset on $\{x, y, z\}$. Examples of chains are

$$\{\emptyset, \{x\}\} \quad \text{and} \quad \{\emptyset, \{y\}, \{y, z\}, \{x, y, z\}\},$$

depicted in Fig. 26a and Fig. 26b, respectively.

Examples of antichains are

$$\{\{x\}, \{y\}\} \quad \text{and} \quad \{\{x, y\}, \{x, z\}, \{y, z\}\},$$

depicted in Fig. 26c and Fig. 26d, respectively.

Example 2.19. Imagine you need to choose a battery based on its mass and cost. Ideally, you want both to be small (Fig. 27). The black markers represent an antichain of choices

$$\{\langle \text{cheap, heavy} \rangle, \langle \text{expensive, light} \rangle\}.$$

Note that in multi-objective optimization, the standard terminology for this is Pareto[8] front. It is a set of pairs because they do not dominate each other: one is cheaper, but is heavier, and the other is more expensive, but lighter, making them incomparable. If a battery with the properties as the red marker existed (very expensive, between light and heavy), that would be an element that cannot be part of the antichain, since it would be dominated by ⟨expensive, light⟩.

[8] From Vilfredo Pareto, italian polymath.

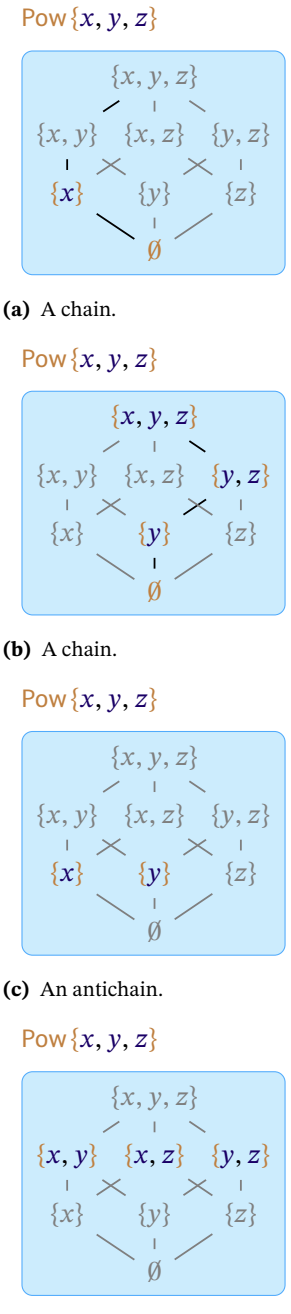


Figure 26: Examples of chains (a-b) and antichains (c-d) in the poset $\mathsf{Pow}\{x, y, z\}$.

Similarly, we could think of a continuous law which relates battery cost and mass. Assume that cheap means 10 CHF, expensive means 20 CHF, light means 250 g, and heavy means 500 g. For instance, consider the antichain given by mass $=$ $500 - 25 \cdot$ cost, with maximum possible cost 20 CHF (Fig. 28).

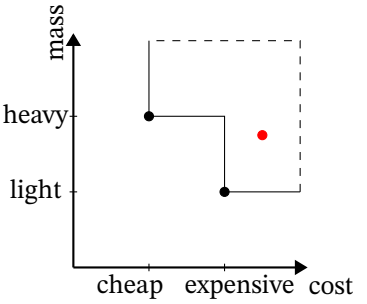


Figure 27: Example of discrete antichains.

2.5 Poset constructions

In this section, we look at a few standard recipes on how we can construct posets from other posets.

Product of posets

Just like the product of sets, we can construct the product of posets. That is a poset with the underlying set being the product of the underlying sets.

Definition 2.20 (Product of posets)
Given posets $\mathbf{P} = \langle P, \preceq_P \rangle$ and $\mathbf{Q} = \langle Q, \preceq_Q \rangle$, the *product poset*

$$\mathbf{P} \times \mathbf{Q} = \langle P \times Q, \preceq_{P \times Q} \rangle,$$

is the set $P \times Q$ equipped with the order $\preceq_{P \times Q}$ given by

$$\frac{\langle p_1, q_1 \rangle \preceq_{P \times Q} \langle p_2, q_2 \rangle}{(p_1 \preceq_P p_2) \wedge (q_1 \preceq_Q q_2)}.$$

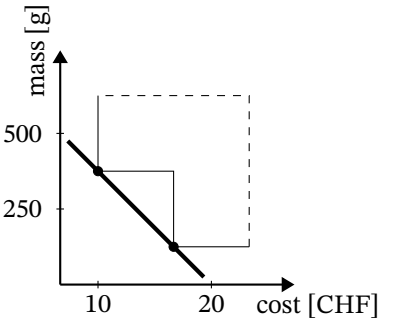
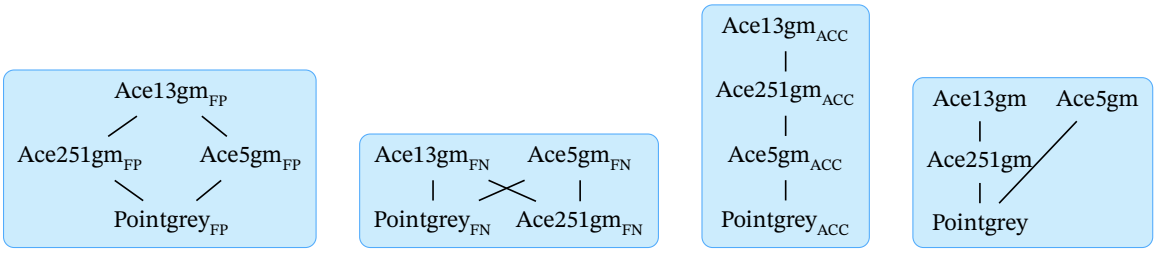


Figure 28: Example of continuous antichains.

Example 2.21. Recall the example of sensor performance curves, presented in Section 2.3. By taking the product of the three posets characterizing false positives, false negatives, and accuracy, one obtains the poset reported in Fig. 29. Here, you can see that Ace13gm and Ace5gm dominate all the other sensors, forming an antichain (they are not comparable).



(a) False positives. **(b)** False negatives. **(c)** Accuracy. **(d)** Product poset.

Figure 29: Hasse diagrams for the sensor performance curves posets, and their product.

Opposite of a poset

Definition 2.22 (Opposite of a poset)
The *opposite of a poset* $\mathbf{P} = \langle P, \leq_P \rangle$ is the poset denoted $\mathbf{P}^{\mathrm{op}} = \langle P, \leq_P^{\mathrm{op}} \rangle$. It has the same elements as $\mathbf{P}$, but is equipped with the reverse ordering, in the sense that, for all $x, y \in P$,

$$\frac{x \leq_P y}{y \leq_P^{\mathrm{op}} x}.$$

For a given $x \in P$, we will sometimes write x^* do denote its corresponding copy in $\mathbf{P}^{\mathrm{op}}$, in order to emphasize that x and x^* belong to distinct posets. However, often we will not be so pedantic with our notation.

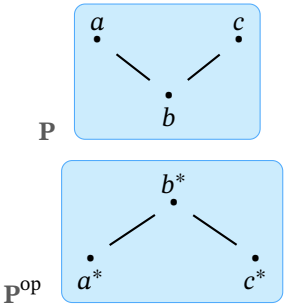


Figure 30: Opposite of a poset.

Example 2.23 (Credit and debt)**.** Let us define the set

$$P = \{0.00, 0.01, 0.02, ...\} \subseteq \mathbb{R}$$

of all CHF monetary quantities approximated to the cent. From this set we can define two posets, $\mathbf{P}^+ = \langle P, \leq \rangle$ and $\mathbf{P}^- = \langle P, \geq \rangle$, that are the opposite of each other. If the context is that, given two quantities 1 CHF and 2 CHF, we prefer 1 CHF to 2 CHF (for example because it is a cost to pay to acquire a component), then we are working in $\mathbf{P}^+$, otherwise we are working in $\mathbf{P}^-$ (for example because it represents the price at which we are selling our product).

2.6 Monotonicity

Monotone maps

A monotone map is the generalization to posets of a "non-decreasing" function on real numbers. The function $x \mapsto \max(0, 42x)$ is non-decreasing on the real numbers because

$$\frac{x \leq y}{\max(0, 42x) \leq \max(0, 42y)}.$$

Note that we use "$\leq$" and not "$<$". "Non-decreasing" is a weaker condition than "increasing".

The definition of monotone function on a poset is the direct generalization of this concept; the only change is that we use the partial orders at hand, rather than the total order on the reals.

Definition 2.24 (Monotone map)
A *monotone map* between two posets $\mathbf{P} = \langle P, \leq_P \rangle$ and $\mathbf{Q} = \langle Q, \leq_Q \rangle$ is a

function $f : \mathbf{P} \to \mathbf{Q}$ that is compatible with the partial-orderings on its source and target in the sense that

$$\frac{x \preceq_{\mathbf{P}} y}{f(x) \preceq_{\mathbf{Q}} f(y)}.$$

Example 2.25 (The identity is monotone)**.** Given a poset $\mathbf{P}$, the identity function $\mathrm{id}_{\mathbf{P}} : \mathbf{P} \to \mathbf{P}$ is a monotone map, since if $x \preceq_{\mathbf{P}} y$, then $\mathrm{id}_{\mathbf{P}}(x) = x \preceq_{\mathbf{P}} y = \mathrm{id}_{\mathbf{P}}(y)$.

Example 2.26 (Constant functions)**.** Every constant function is a monotone map.

Example 2.27 (Cardinality map)**.** Consider the power poset (Def. 2.12) $\mathsf{Pow}\,\mathbf{A}$ of a finite set $\mathbf{A}$. The cardinality map

$$\mathsf{card} : \mathsf{Pow}\,\mathbf{A} \to \mathbb{N}$$

is monotone when considered as a map from the poset $\mathsf{Pow}\,\mathbf{A}$ to the poset $\langle \mathbb{N}, \leq \rangle$. Figure 31 shows a visualization of this map for the set $\mathbf{A} = \{x, y, z\}$. To prove this, recall that in the power poset subsets are ordered by inclusion. Therefore, we need to show that

$$\frac{\mathbf{S} \subseteq \mathbf{T}}{\mathsf{card}(\mathbf{S}) \leq \mathsf{card}(\mathbf{T})}.$$

It is easy to see that, because all elements of $\mathbf{S}$ are also in $\mathbf{T}$, the cardinality of $\mathbf{S}$ cannot be more than the cardinality of $\mathbf{T}$. Monotonicity depends on the partial order used on the domain and the codomain. To indicate that a map is monotone, we write it indicating the two posets as the domain/codomain:

$$\mathsf{card} : \langle \mathsf{Pow}\,\mathbf{A}, \subseteq \rangle \to \langle \mathbb{N}, \leq \rangle.$$

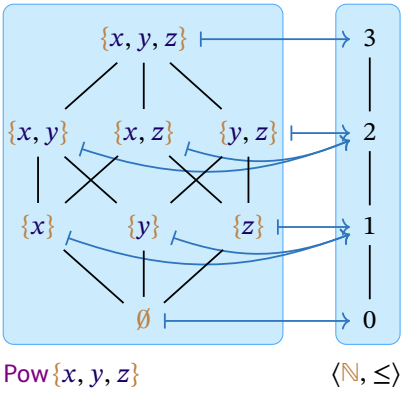


Figure 31: card is a monotone map.

Lemma 2.28. Any map $f : \mathbf{P} \to \mathbf{Q}$ is monotone, when $\mathbf{P}$ is a discrete poset.

Antitone maps

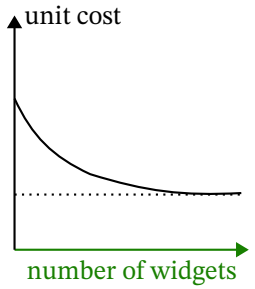
Dually to monotone functions, we can define antitone maps as order *reversing* functions.

Definition 2.29 (Antitone map)
An *antitone map* between two posets $\mathbf{P} = \langle \mathbf{P}, \preceq_{\mathbf{P}} \rangle$ and $\mathbf{Q} = \langle \mathbf{Q}, \preceq_{\mathbf{Q}} \rangle$ is a

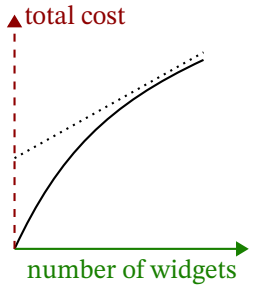
map f that reverses the ordering, in the sense that

$$\frac{x \preceq_{\mathbf{P}} y}{f(x) \succeq_{\mathbf{Q}} f(y)}.$$

Example 2.30 (Unit cost, total cost)**.** Assume that you want to produce some widgets, and that the manufacturing cost depends on the number of widgets. The function describing the total cost $t : \mathbb{N} \to \mathbb{R}_{\geq 0}$ is a map between the ordered sets $\mathbb{N}$ and $\mathbb{R}_{\geq 0}$, and maps each quantity of widgets to a total manufacturing cost (Fig. 32b). Clearly, t is a monotone function. Conversely, the unit cost function $u : \mathbb{N} \to \mathbb{R}_{\geq 0}$ is antitone (Fig. 32a).



(a) Unit cost vs. number of widgets.



(b) Total cost vs. number of widgets.

Figure 32: Unit and total costs vs. number of widgets.

It is easy to see that an antitone map $f : \mathbf{P} \to \mathbf{Q}$ is the same thing as a monotone map $f : \mathbf{P}^{\mathrm{op}} \to \mathbf{Q}$.

Lemma 2.31. An antitone map $f : \mathbf{P} \to \mathbf{Q}$ is a monotone map $f : \mathbf{P}^{\mathrm{op}} \to \mathbf{Q}$ and a monotone map $f : \mathbf{P} \to \mathbf{Q}^{\mathrm{op}}$.

Monotonicity is a compositional property: the composition of two monotone maps is monotone.

Lemma 2.32. Given posets $\mathbf{P}, \mathbf{Q}, \mathbf{R}$ and two monotone maps. $f : \mathbf{P} \to \mathbf{Q}$ and $g : \mathbf{Q} \to \mathbf{R}$, the composite map $f \fatsemi g : \mathbf{P} \to \mathbf{R}$ is monotone as well.

See proof on page 203.

Remark 2.33 (Order on monotone maps)**.** Fixed two posets $\mathbf{P}$ and $\mathbf{Q}$, the set of monotone maps $\mathbf{P} \to \mathbf{Q}$ form a poset themselves. We can order them point wise, using the same order we defined for general maps. The example related to the ordering of sensor-algorithm performance curves is an example of this order.

2.7 Poset bounds

In co-design, it will be important to identify poset bounds.

Minimal and maximal elements

You know already the operators min/max that give the minimum/maximum values of a set of real numbers. If the set is finite, the minimum and maximum always exist. But for infinite sets, the minimum and maximum might not exist. For example, consider the set of real numbers contained between 0 and 1, excluding the boundaries:

$$\mathbf{A} = \{x \in \mathbb{R} : 0 < x < 1\}.$$

This set does not have a minimum or a maximum.

For a total order, if the minimum and maximum exist, then they are unique. In a partial order, this is not the case. We introduce the operators Min and Max that are the generalization to partial orders of min / max.

Definition 2.34 (Minimal and maximal elements)
Min : $\mathsf{Pow}\,\mathbf{P} \to \mathsf{Anti}\,\mathbf{P}$ is the map that sends a subset $\mathbf{S}$ of a poset to the *minimal elements* of that subset (those elements $a \in \mathbf{S}$ such that $a \preceq_{\mathbf{P}} b$ for all $b \in \mathbf{S}$). In formulas:

$$\begin{aligned} \mathrm{Min} : \mathsf{Pow}\,\mathbf{P} &\to \mathsf{Anti}\,\mathbf{P}, \\ \mathbf{S} &\mapsto \left\{ c \in \mathbf{S} : \frac{d \in \mathbf{S} \quad d \preceq_{\mathbf{P}} c}{c = d} \right\}. \end{aligned}$$

Max : $\mathsf{Pow}\,\mathbf{P} \to \mathsf{Anti}\,\mathbf{P}$ is the map that sends a subset $\mathbf{S}$ of a poset to the *maximal elements* of that subset (those elements $a \in \mathbf{S}$ such that $a \succeq_{\mathbf{P}} b$ for all $b \in \mathbf{S}$). In formulas:

$$\begin{aligned} \mathrm{Max} : \mathsf{Pow}\,\mathbf{P} &\to \mathsf{Anti}\,\mathbf{P}, \\ \mathbf{S} &\mapsto \left\{ c \in \mathbf{S} : \frac{d \in \mathbf{S} \quad d \succeq_{\mathbf{P}} c}{c = d} \right\}. \end{aligned}$$

Note that Min($\mathbf{S}$) and Max($\mathbf{S}$) could be empty.

Upper/lower bounds

Definition 2.35 (Upper bounds in a poset)
The *upper bounds* of a subset $\mathbf{S}$ of a poset $\mathbf{P}$ are, if they exist, the elements of $\mathbf{P}$ which dominate all elements in $\mathbf{S}$. In other words, the upper bounds of $\mathbf{S}$ are the elements of the set

$$\mathsf{UppB}\,\mathbf{S} := \{ y \in \mathbf{P} \mid \forall x \in \mathbf{S} : \; x \preceq_{\mathbf{P}} y \}.$$

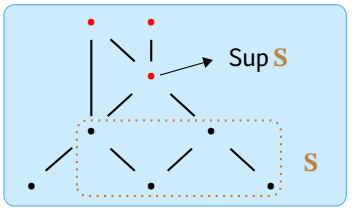


Figure 33: Example of upper bounds and least upper bound for $\mathbf{S}$.

Definition 2.36 (Least upper bound / join / supremum)
A *least upper bound* of $\mathbf{S} \subseteq \mathbf{P}$, if it exists, is the least element among the upper bounds of $\mathbf{S}$. It is denoted $\vee\mathbf{S}$ or $\mathsf{Sup}\,\mathbf{S}$, and also called the *join* or *supremum* of $\mathbf{S}$.

So, given $\mathbf{S} \subseteq \mathbf{P}$ and $y \in \mathbf{P}$, $y = \vee\mathbf{S}$ if and only if

1. $x \preceq_{\mathbf{P}} y$, $\forall x \in \mathbf{S}$, and
2. $x \preceq_{\mathbf{P}} z$, $\forall x \in \mathbf{S} \Rightarrow y \preceq_{\mathbf{P}} z$.

Lemma 2.37. Let $\mathbf{P}$ be a poset and $\mathbf{S} \subseteq \mathbf{P}$ a subset of the underlying set of $\mathbf{P}$. If $\bigvee \mathbf{S}$ exists, then it is unique.

See proof on page 203.

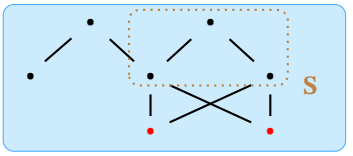


Figure 34: Example of lower bounds of $\mathbf{S}$.

Example 2.38. Consider the poset $\mathbf{P}$ and its subset $\mathbf{S}$ depicted in Fig. 33. The red markers • represent the upper bound of $\mathbf{S}$. For this specific case, there is *a single* least upper bound.

Example 2.39. Least upper bounds need not necessarily exist even in total orders. For instance, the subset

$$\mathbb{R}_{>0} = \{x \in \mathbb{R} : x > 0\}$$

of the poset $\mathbb{R}$ (with the usual ordering) does not have a least upper bound.

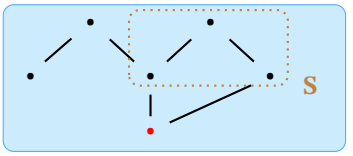


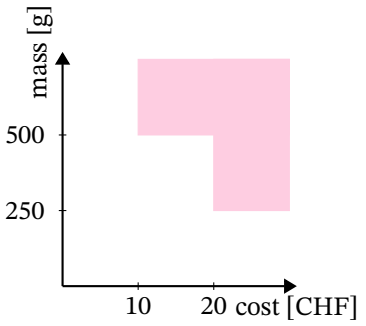
Figure 35: Example of lower bounds and greatest lower bounds of $\mathbf{S}$.

Analogously to the case of (least) upper bounds, we can define lower bounds and greatest lower bounds.

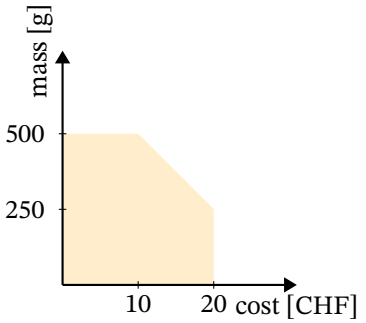
Definition 2.40 (Lower bounds in a poset)
The *lower bounds* of a subset $\mathbf{S}$ of a poset $\mathbf{P}$ are, if they exist, the elements which are dominated by all elements in $\mathbf{S}$. In other words, the lower bounds of $\mathbf{S}$ are the elements of the set

$$\mathsf{LowB}\,\mathbf{S} := \{y \in \mathbf{P} \mid \forall x \in \mathbf{S} : y \leq_{\mathbf{P}} x\}.$$

Definition 2.41 (Greatest lower bound / meet / infimum)
The *greatest lower bound*, if it exists, is the greatest among the lower bounds of $\mathbf{S}$. This is denoted $\bigwedge \mathbf{S}$ or $\mathsf{Inf}\,\mathbf{S}$ and also called the *meet* or *infimum* of $\mathbf{S}$.



(a)



(b)

Figure 36: Examples of upper and lower sets.

Example 2.42. It is easy to come up with an example of a subset $\mathbf{S}$ of a poset $\mathbf{P}$ which has lower bounds but no greatest lower bound.

In Fig. 34 you find an example of a subset $\mathbf{S}$ of a poset $\mathbf{P}$ which has incomparable lower bounds. In Fig. 35 instead, there is a greatest lower bound.

Upper and lower sets

Definition 2.43 (Upper and lower set)
An *upper set* $\mathbf{U}$ is a subset of a poset $\mathbf{P}$ such that, if $x \in \mathbf{U}$, then all elements of $\mathbf{P}$ that are above x are also in $\mathbf{U}$. In other words:

$$\frac{x \in \mathbf{U} \quad x \leq_{\mathbf{P}} y}{y \in \mathbf{U}}.$$

A *lower set* $\mathbf{L}$ is a subset of a poset $\mathbf{P}$ such that, if $x \in \mathbf{L}$, then all elements of $\mathbf{P}$ that are below x are also in $\mathbf{L}$. In other words:

$$\frac{x \in \mathbf{L} \quad y \preceq_{\mathbf{P}} x}{y \in \mathbf{L}}.$$

We call $\mathsf{USets}\,\mathbf{P}$ the set of upper sets of $\mathbf{P}$ and $\mathsf{LSets}\,\mathbf{P}$ the set of lower sets of $\mathbf{P}$.

Given the battery choices $\{\langle 10\,\text{CHF}, 500\,\text{g}\rangle, \langle 20\,\text{CHF}, 250\,\text{g}\rangle\}$, we can represent an upper set as in Fig. 36a. The upper set can be interpreted as all the potential battery choices which are dominated by at least one of the two choices we have (in case we want to minimize mass and cost). Similarly, the lower set in Fig. 36b can be interpreted as all the potential battery choices which dominate at least one of the choices we have. Here when considering "the choices we have" in Fig. 36b, we not only consider the two choices directly presented to us, but also any convex combination of them.

Upper and lower closure

Definition 2.44 (Upper closure operator)
The *upper closure operator* $\uparrow$ maps a subset to the smallest upper set that includes it:

$$\begin{aligned} \uparrow : \mathsf{Pow}\,\mathbf{P} &\to \mathsf{USets}\,\mathbf{P}, \\ \mathbf{P} &\mapsto \{y \in \mathbf{P} \mid \exists x \in \mathbf{P} : x \preceq_{\mathbf{P}} y\}. \end{aligned}$$

Remark 2.45. Note that, by definition, an upper set is closed to upper closure.

Lemma 2.46. For any $\mathbf{S} \in \mathsf{Pow}\,\mathbf{P}$, $\uparrow \mathbf{S}$ is in fact an upper set.

See proof on page 203.

Lemma 2.47. The upper closure operator $\uparrow$ is an antitone map.

See proof on page 203.

In the example of battery choices (in the numerical case), first, consider the upper closure of a single element of the poset, for instance $p_1 = \langle 10\,\text{CHF}, 500\,\text{g}\rangle$ (Fig. 37, left). Second, we can look at the upper closure when we add the choice $p_2 = \langle 20\,\text{CHF}, 250\,\text{g}\rangle$ (Fig. 37, center).

Note that the upper set of the subset formed by the two elements is the union of the upper sets of the single elements. Finally, we can also define the set

$$\mathbf{S} = \{\langle \text{cost}, \text{mass}\rangle \mid \text{mass} = 750 - 25 \cdot \text{cost}, \forall\ \text{cost} \in [0, 20]\},$$

and find its upper closure (Fig. 37, right).

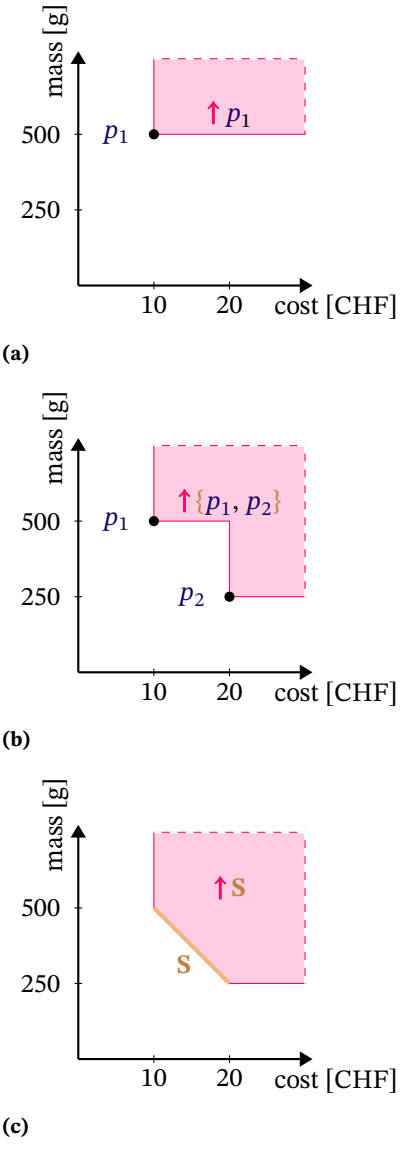
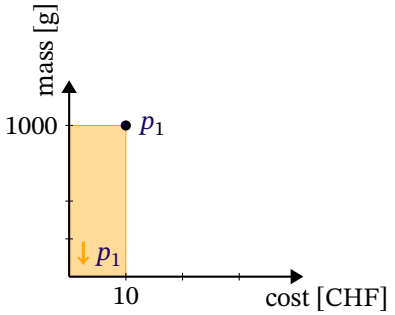
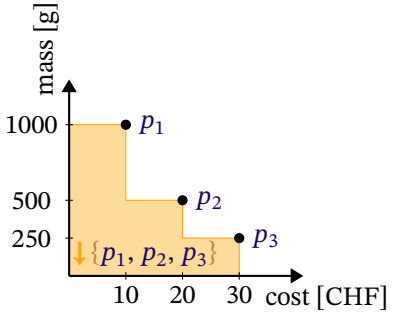


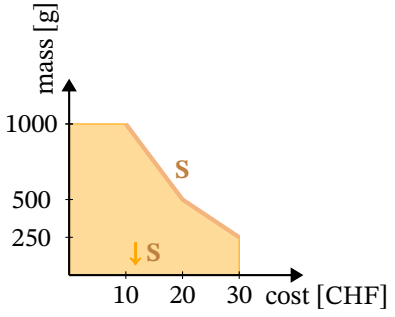
Figure 37: Example of upper closure for different sets of battery choices.



(a)



(b)



(c)

Figure 38: Example of lower closure for different sets of battery choices.

Definition 2.48 (Lower closure operator)
The *lower closure operator* $\downarrow$ maps a subset to the smallest lower set that includes it:

$$\begin{aligned} \downarrow : \operatorname{Pow} \mathbf{P} &\to \operatorname{LSets} \mathbf{P}, \\ \mathbf{S} &\mapsto \{y \in \mathbf{P} \mid \exists x \in \mathbf{S} : y \leq_{\mathbf{P}} x\}. \end{aligned}$$

Lemma 2.49. The lower closure operator $\downarrow$ is a monotone map.

See proof on page 203.

Consider the battery example, and the antichain given by the battery models $p_1 = \langle 10\,\text{CHF}, 1000\,\text{g}\rangle$, $p_2 = \langle 20\,\text{CHF}, 500\,\text{g}\rangle$, and $p_3 = \langle 30\,\text{CHF}, 250\,\text{g}\rangle$ (Fig. 38, left). The lower closure operator $\downarrow\{p_1, p_2, p_3\}$ represents all the battery models which, if existing, would dominate $\{p_1, p_2, p_3\}$. We could instead consider linear maps between the points getting a poset $\mathbf{P}$, and obtain the lower closure depicted in Fig. 38 on the right.

Antichains and upper sets

Lemma 2.50. Let $\mathbf{A}$ and $\mathbf{B}$ be subsets of $\mathbf{P}$ that are antichains. Then

$$\frac{\uparrow \mathbf{A} = \uparrow \mathbf{B}}{\mathbf{A} = \mathbf{B}}.$$

See proof on page 204.

Definition 2.51 (Downward and upward closed sets)
An upper set $\mathbf{S}$ is *downward-closed* in a poset $\mathbf{P}$ if

$$\mathbf{S} = \uparrow \operatorname{Min} \mathbf{S}.$$

A lower set $\mathbf{S}$ is *upward-closed* in a poset $\mathbf{P}$ if

$$\mathbf{S} = \downarrow \operatorname{Max} \mathbf{S}.$$

The set of downward-closed upper sets of $\mathbf{P}$ is denoted $\overline{\operatorname{UpSets}}\,\mathbf{P}$, and the one of upward-closed lower sets of $\mathbf{P}$ is denoted $\overline{\operatorname{LowSets}}\,\mathbf{P}$.

2.8 Lattices

Definition 2.52 (Lattice)
A *lattice* is a poset $\mathbf{P} = \langle \mathbf{P}, \leq_{\mathbf{P}} \rangle$ with the additional property that, for any two-element subset $\{x, y\} \subseteq \mathbf{P}$, both the join $\vee\{x, y\}$ and the meet $\wedge\{x, y\}$

exist. Usually these are written using infix notation as $x \vee y$ and $x \wedge y$, respectively.

Definition 2.53 (Top and bottom)
If there is a least upper bound for the entire lattice $\mathbf{P}$, it is called the *top* ($\top$). If a greatest lower bound exists, it is called the *bottom* ($\bot$).

Definition 2.54 (Bounded lattices)
If both a top and a bottom exist, we call the lattice *bounded*, and denote it by $\mathbf{P} = \langle \mathbf{P}, \leq_{\mathbf{P}}, \vee, \wedge, \bot, \top \rangle$.

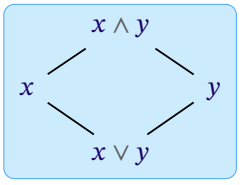


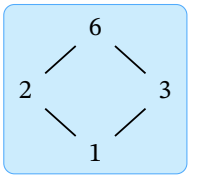
Figure 39: Lattice structure.

Example 2.55. In Def. 2.12 we presented the poset arising from the power set $\mathrm{Pow}\,\mathbf{A}$ of a set $\mathbf{A}$ and ordered via subset inclusion. This is a lattice, bounded by $\mathbf{A}$ and by the empty set $\emptyset$. Note that this lattice possesses two (dual) monoidal structures $\langle \mathrm{Pow}\,\mathbf{A}, \subseteq, \emptyset, \cup \rangle$ and $\langle \mathrm{Pow}\,\mathbf{A}, \subseteq, \mathbf{A}, \cap \rangle$.

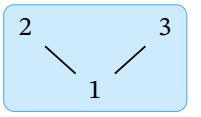
Example 2.56. Consider the poset **Bool**, in which $b_1 \leq_{\mathbf{Bool}} b_2$ iff $b_1 \Rightarrow b_2$, that is, in addition to the operation

$$\Rightarrow : \mathbf{Bool} \times \mathbf{Bool} \to \mathbf{Bool},$$

called *implication*, there are also the familiar *and* ($\wedge$) and *or* ($\vee$) operations. Note that $\wedge$ and $\vee$ are commutative ($b \wedge c = c \wedge b$, $b \vee c = c \vee b$), whereas $\Rightarrow$ is not. Furthermore, $\wedge$ and $\vee$ correspond to the meet and the join, respectively.



(a) A lattice.



(b) Not a lattice.

Figure 40: Examples of a lattice and a non-lattice.

Example 2.57. Consider the set $\{1, 2, 3, 6\}$ ordered by divisibility. For instance, since 2 divides 6, we have $2 \leq 6$. This is a lattice. However, the set $\{1, 2, 3\}$ ordered by divisibility is not, since 2 and 3 lack a meet (Fig. 40).

Lemma 2.58. $U\mathbf{P}$ is a bounded lattice (Def. 2.52) with

$$\leq_{U\mathbf{P}} := \supseteq,\ \bot_{U\mathbf{P}} := \mathbf{P},\ \top_{U\mathbf{P}} := \emptyset,\ \wedge_{U\mathbf{P}} := \cap,\ \vee_{U\mathbf{P}} := \cup.$$

See proof on page 204.

Lemma 2.59. $L\mathbf{P}$ is a bounded lattice (Def. 2.52) with:

$$\leq_{L\mathbf{P}} := \subseteq,\ \bot_{L\mathbf{P}} := \emptyset,\ \top_{L\mathbf{P}} := \mathbf{P},\ \wedge_{L\mathbf{P}} := \cup,\ \vee_{L\mathbf{P}} := \cap.$$

See proof on page 205.

a	b	$a \leq b$	$a \wedge b$	$a \vee b$
$\top$	$\top$	$\top$	$\top$	$\top$
$\top$	$\bot$	$\bot$	$\bot$	$\top$
$\bot$	$\top$	$\top$	$\bot$	$\top$
$\bot$	$\bot$	$\top$	$\bot$	$\bot$

Table 2: Properties of the **Bool** poset. Note that $\leq \equiv \Rightarrow$.

Co-design 3

One of the first rules of science is if somebody delivers a secret weapon to you, you better use it.
—Herbert A. Simon

In this chapter we start introducing the theory of co-design which will be leveraged in the rest of the thesis. Starting from basic concepts of formal engineering design (Section 3.1), we will then talk about queries in design, first in a general context (Section 3.2), then precisely, in the context of co-design problems with implementations (Section 3.3 and Section 3.4). We will then introduce the notion of "co-design problems with implementations", showing how one can interconnect multiple design problems (Section 3.5).

3.1 Basic concepts of formal engineering design

In this section, we introduce some basic concepts of formal engineering design.

Functionality and functional requirements You are an engineer in front of an empty whiteboard, ready to start designing the next product. The first question to ask is: What is the *purpose* of the product to be designed [18]? The purpose of the product is expressed by the *functional requirements*, sometimes called *functional specifications* (e.g., in formal methods), *desired behavior* (e.g., in robotics), *objectives* (e.g., in optimization), *guarantees* (e.g., in contract theory), *conclusions* (e.g., in proof theory), or simply *function*.

Unfortunately, the word "function" conflicts with the mathematical concept. Therefore, we will talk about *functionality*.

Example 3.1. These are a few examples of functional requirements:

- A car must be able to transport at least $n \geq 4$ passengers.
- A battery must store at least 100 kJ of energy.
- An autonomous vehicle should reach at least 40 km/h while guaranteeing safety.
- A refrigerator must maintain a certain temperature [29].

Resources and resource constraints We call *resources* what we need to pay to realize the given functionality. In some contexts, these are better called *costs*, *requirements*, *dependencies*, or *assumptions*.

Example 3.2. These are a few examples of resource constraints:

- ▹ A car should not cost more than 15,000 CHF.
- ▹ A battery should not weigh more than 1 kg.
- ▹ A process should not take more than 10 s.

Duality of functionality and resources There is an interesting duality between functionality and resources. When designing systems, one is given functional requirements, as a *lower bound* on the functionality to provide, and one is given resource constraints, which are an *upper bound* on the resources to use.

As far as design objectives go, most can be understood as either *minimize resource usage* or *maximize functionality provided*.

This duality between functionality and resources will be at the center of the co-design formalization.

Non-functional requirements Functionality and resources do not cover all the requirements– there is, for example, a large class of *non-functional requirements* [18] such as the extensibility and the maintainability of the system. Nevertheless, functionality and resources can express most of the requirements which can be quantitatively evaluated, at least prior to designing, assembling, and testing the entire system.

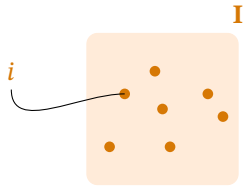


Figure 41: An *implementation* i is a particular point in the implementation space $\mathbf{I}$.

Implementation space The *implementation space*, also known as the *design space*, is the set of all possible design choices that could be chosen; by *implementation*, or the word "design", used as a noun, we mean one particular set of choices. Implementations are also known as *plans*, *blueprints*, or *decision variables*. The implementation space $\mathbf{I}$ is the set over which we are optimizing; an implementation $i \in \mathbf{I}$ is a particular point in that set (Fig. 41).

The interconnection between functionality, resources, and implementation spaces is as follows. We will assume that, given one implementation, we can evaluate it to know the functionality and the resources spaces (Fig. 42).

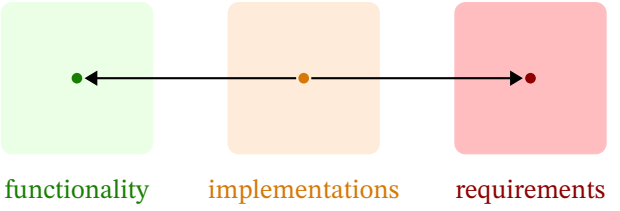


Figure 42: Evaluation of specific implementations to get functionality and resources spaces.

Functional interfaces and interconnection Components are *interconnected* to create a system. This implies that we have defined the *interfaces* of components,

which have the dual function of delimiting when one component ends and another begins, and also to describe exactly what is the nature of their interaction.

We will develop a formalism in which the functionality and resources are the interfaces used for interconnection: two components are connected if the resources required by the first correspond to the functionality provided by the second.

Abstraction By *abstraction*, we mean that it is possible to "zoom out", in the sense that a system of components can be seen as a component itself, which can be part of larger systems. For instance, imagine the components composing a complex electronic circuit building a vision sensor. Now, imagine the components of the autonomy stack of a robot (e.g., perception, planning, control). In this context, the vision sensor is an interconnection of small components, and at the same time a component in the autonomy stack.

Compositionality A *compositional* property is a property that is preserved by interconnection and abstraction; assuming each component in a system satisfies that property, also the system as a whole satisfies the property. For instance, one can compose two electronic circuits by joining their terminals to obtain another electronic circuit. We would say that the property of being an electronic circuit is compositional.

3.2 Queries in design

Suppose that we have a model with a functionality space $\mathbf{F}$, a requirements space $\mathbf{R}$, and an implementation space $\mathbf{I}$.

There are several queries we can ask of a model. They all look at the same phenomenon from different angles, so they look similar; however the computational cost of answering each one might be very different.

The first kind of query is one that asks if the design is feasible when fixed all variables.

Problem (Feasibility problem)**.** Given a triplet of implementation $i \in \mathbf{I}$, functionality $f \in \mathbf{F}$, requirements $r \in \mathbf{R}$, determine if the design is feasible.

The second type of query is that which fixes the boundary conditions of functionality and requirements, and asks to find a solution.

Problem (Find implementation)**.** Given a pair of minimal requested functionality $f \in \mathbf{F}$ and maximum allowed requirements $r \in \mathbf{R}$, determine if there is an implementation $i \in \mathbf{I}$ that is feasible.

A different type of query is the one in which the design objective (the functionality) is fixed, and we ask what are the least resources necessary.

Problem (FixFunMinRes). Given a certain functionality $f \in \mathbf{F}$, find the set of "minimal" resources in $\mathbf{R}$ that are needed to realize it (along with the implementations), or provide a proof that there are none (a certificate of infeasibility).

Dually, we can ask, fixed the resources available, what are the functionalities that can be provided.

Problem (FixResMaxFun). Given a certain requirement $r \in \mathbf{R}$, find the set of "maximal" functionalities that can realize it (along with the implementations), or provide a proof that there are none (a certificate of infeasibility).

It is very natural to talk about the "minimal" requirements and "maximal" functionalities; after all, we always want to minimize costs and maximize performance. In the next chapter we start to put more mathematical scaffolding in place, starting from defining functionality and requirements as posets.

3.3 Design Problems with Implementation

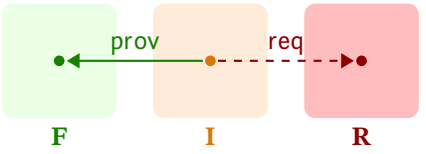


Figure 43: Design problems with implementation.

We start by defining a "design problem with implementation", which is a tuple of "functionality space", "implementation space", and "resources space", together with two maps that describe the feasibility relations between these three spaces (Fig. 43).

Definition 3.3 (Design problem with implementation)
A *design problem with implementation* (DPI) is a tuple

$$\langle \mathbf{F}, \mathbf{R}, \mathbf{I}, \mathsf{prov}, \mathsf{req} \rangle,$$

where:

- $\mathbf{F}$ is a poset, called *functionality space*;
- $\mathbf{R}$ is a poset, called *requirements space*;
- $\mathbf{I}$ is a set, called *implementation space*;
- $\mathsf{prov} : \mathbf{I} \to \mathbf{F}$ maps an implementation to the functionality it provides;
- $\mathsf{req} : \mathbf{I} \to \mathbf{R}$ maps an implementation to the resources it requires.

Example 3.4. Suppose we need to choose a motor for a robot from a given set. The functionality of a motor can be parametrized by torque and speed. The resources could include the cost [USD], the mass [g], the input voltage [V], and the input current [A]. The map $\mathsf{prov} : \mathbf{I} \to \mathbf{F}$ assigns to each motor its functionality, and the map $\mathsf{req} : \mathbf{I} \to \mathbf{R}$ assigns to each motor the resources it needs (Fig. 44).

Graphical notation

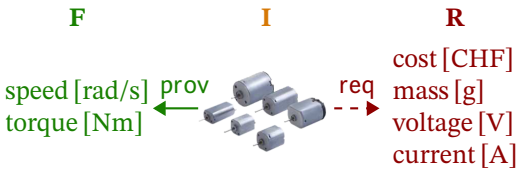


Figure 44: Design problem of an eletric motor.

A graphical notation will help reasoning about composition. A DPI is represented as a box with n_f green edges and n_r red edges (Fig. 45). This means that the functionality and resources spaces can be factorized in n_f and n_r components:

$$\mathbf{F} = \prod_{i=1}^{\mathrm{n}_f} \mathrm{pr}_i \mathbf{F}_i, \quad \mathbf{R} = \prod_{j=1}^{\mathrm{n}_r} \mathrm{pr}_j \mathbf{R}_j,$$

where "pr_i" represents the projection to the i-th component. If there are no green (respectively, red) edges, then n_f (respectively, n_r) is zero, and $\mathbf{F}$ (respectively, $\mathbf{R}$) is equal to $\mathbf{1} = \{\langle\,\rangle\}$, the set containing one element, the empty tuple $\langle\,\rangle$.

These *co-design diagrams* are not to be confused with signal flow diagrams, in which the boxes represent oriented systems and the edges represent signals. More on that in Section 8.3.

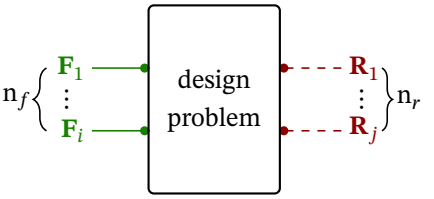


Figure 45: Graphical notation for design problems.

Examples of design problems

We now present a list of design problems for different disciplines, to showcase the universality of the approach. Throughout the manuscript, we will often introduce new examples.

Example 3.5 (Electric motor)**.** An electric motor provides speed and torque, and requires cost, mass, voltage, and current. The implementations are given by different motor models/technologies, for instance available in catalogues.

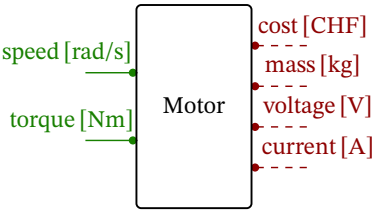


Figure 46: Design problem for an electric motor.

Example 3.6 (Gearbox)**.** A gearbox (Fig. 47) provides a certain output torque τ_o and speed ω_o, given a certain input torque τ_i and speed ω_i. For an ideal gearbox with a reduction ratio $r \in \mathbb{Q}_+$ and efficiency ratio γ, $0 < \gamma < 1$, the constraints among those quantities are $\omega_i \geq r\,\omega_o$ and $\tau_i\omega_i \geq \gamma\,\tau_o\omega_o$. With this simple model, the set of implementations are given by the possible values of reduction and efficiency ratio.

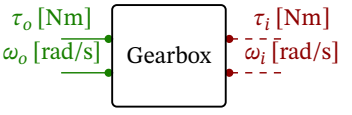


Figure 47: Design problem for a gearbox.

Example 3.7 (Congested roads)**.** In classic transportation science, when designing a road, one leverages a key relationship between number of vehicles which access the road, the capacity of the road, and the travel time needed to traverse it (Fig. 48). A common way of relating these quantities is via the Bureau of Public

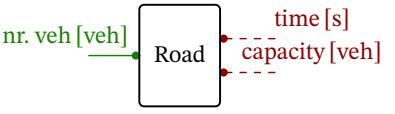


Figure 48: Design problem for the design of a road.

Roads (BPR) function

$$t(x) = t_{\text{nom}}\left(1 + \alpha\left(\frac{x}{k}\right)^{\beta}\right),$$

where t is the time needed to traverse the road, t_{nom} is the nominal time in case of no congestion, k is the capacity of the road, x is the number of vehicles on the road (congestion), and α, β are positive parameters, which, for instance, could define different implementations.

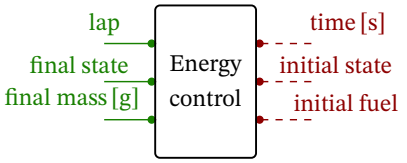


Figure 49: Design problem for the energy management of a formula 1 car.

Example 3.8 (Formula 1 hybrid electric power unit)**.** When designing the hybrid electric power unit of a formula 1 car, one can think of a design problem to represent the energy management for completing one lap. In this context, the energy management must ensure a minimum battery state at the *end* of the lap, as well as a minimum mass. In racing scenarios, one cares about minimizing lap time, initial battery state and fuel (Fig. 49). The implementations are given by different control strategies to manage the energy during the lap, and relationships can be obtained by solving optimization problems [129].

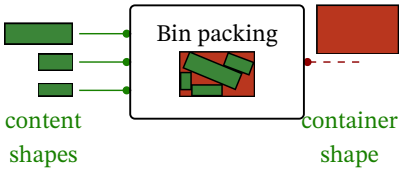


Figure 50: Design problem for bin packing

Example 3.9 (Bin packing)**.** Suppose that each internal component occupies a volume bounded by a parallelepiped, and that we must choose the minimal enclosure in which to place all components (Fig. 50). What is the minimal size of the enclosure? This is a variation of the *bin packing* problem, which is in NP for both 2D and 3D [130]. It is easy to see that the problem is monotone, by noticing that, if one the components shapes increases, then the size of the enclosure cannot shrink. The implementations, in this case, are the configurations which one can choose to place all components in the container (one of the possible configurations is shown in the picture).

Figure 51: Design problem for SLAM.

Example 3.10 (SLAM)**.** One issue with particle-filter-based estimation procedures, such as the ones used in the popular GMapping [131] suite, is that the filter might diverge if there aren't enough particles. Although the relation might be hard to characterize, there is a monotone relation between the robustness (1 - probability of failure), the accuracy, and the number of particles (Fig. 51). Here, the implementation space contains the other choices of parameters for the filter: fixed the number of particles, by changing the tuning of the filter, we can explore the trade-off of accuracy and robustness.

Figure 52: Design problem for progressive stereo reconstruction.

Example 3.11 (Stereo reconstruction)**.** Progressive reconstruction systems [132], which start with a coarse approximation of the solution that is progressively refined, are described by a smooth relation between the resolution and the latency to obtain the answer (Fig. 52). A similar relation characterizes any anytime algorithms in other domains, such as robot motion planning. Here, the implementations could be a list of different algorithms (or algorithm parameters) to

perform the task at hand, and the specific relationships might be obtained by benchmarking.

Example 3.12. In a multi-robot system there is always a trade-off between the number of robots and the capabilities of the single robot. Suppose we need to create a swarm of agents whose functionality is to sweep an area. If the functionality is fixed, one expects a three-way trade-off between the three resources: number of agents, the speed of a single agent, and the execution time (Fig. 53). For example, if the time available decreases, we have to increase either the speed of an agent or the number of agents (Fig. 53b).

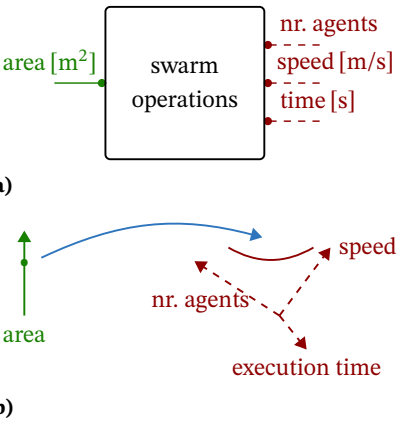


Figure 53: Design problem for swarm operations and sketch of functionality-resources trade-off.

Example 3.13. The trivial model of a CPU is as a device that provides computation, measured in flops, and requires power (in W). Clearly there is a monotone relation between the two (Fig. 54).

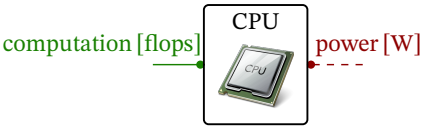


Figure 54: Design problem for a CPU.

Example 3.14. Svorenova *et al.* [41] consider a joint sensor scheduling and control synthesis problem, in which a robot can decide to not perform sensing to save power, given performance objectives on the probability of reaching the target and the probability of collision. The method outputs a Pareto frontier of all possible operating points. This can be cast as a design problem with functionality equal to the probability of reaching the target and (the inverse of) the collision probability, and with resources equal to the actuation power, sensing power, and sensor accuracy (Fig. 55).

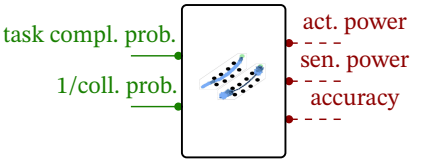


Figure 55: Design problem for joint sensor scheduling and control synthesis problem.

Example 3.15. Nardi *et al.* [133] describe a benchmarking system for visual SLAM that provides the empirical characterization of the monotone relation between the accuracy of the visual SLAM solution, the throughput [frames/s] and the energy for computation [J/frame]. The implementation space is the product of algorithmic parameters, compiler flags, and architecture choices, such as the number of GPU cores active. This is an example of a design problem whose functionality-resources map needs to be experimentally evaluated (Fig. 56).

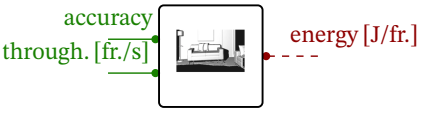


Figure 56: Design problem for SLAM benchmarking.

3.4 Queries, more precisely

A DPI is a model to which we can associate a family of optimization problems. While in previous examples we covered the problem "feasibility", we still miss FixFunMinRes, FixResMaxFun, and FeasibleImp.

The first can be translated to "Given a lower bound on the functionality f, what are the implementations that have minimal resource usage?" (Fig. 57).

Problem (FixFunMinRes)**.** Given $f \in \mathbf{F}$, find the implementations in $\mathbf{I}$ that realize the functionality f (or higher) with minimal resources, or provide a proof

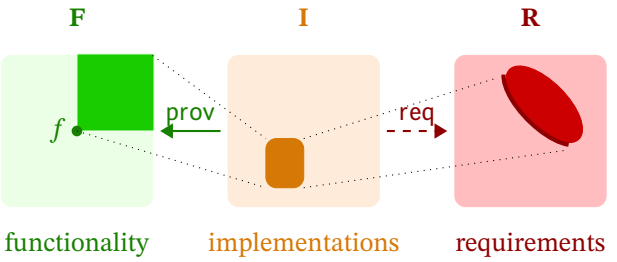


Figure 57: Graphical representation of FixFunMinRes.

that there are none:

$$\begin{cases} \text{using} & i \in \mathbf{I}, \\ \text{Min}_{\preceq_\mathbf{R}} & r, \\ \text{s.t.} & r = \mathsf{req}(i), \\ & f \preceq_\mathbf{F} \mathsf{prov}(i). \end{cases} \tag{1}$$

Remark 3.16 (Minimal vs. least solutions)**.** Note the use of $\text{Min}_{\preceq_\mathbf{R}}$ in (1), which indicates the set of minimal (non-dominated) elements according to $\preceq_\mathbf{R}$, rather than $\min_{\preceq_\mathbf{R}}$, which would presume the existence of the least element. In all problems in this paper, the goal is to find the optimal trade-off of resources ("Pareto front"). So, for each f, we expect to find an antichain $R \in \text{Anti}\,\mathbf{R}$. We will see that this formalization allows an elegant way to treat multi-objective optimization problems. The algorithm to be developed will directly solve for the set R, without resorting to techniques such as *scalarization*, and therefore is able to work with arbitrary posets, possibly *discrete*.

In an entirely symmetric fashion, we could fix an upper bound on the resource usage, and then maximize the functionality provided (Fig. 58). The formulation is entirely dual, in the sense that it is obtained from (1) by swapping Min with Max, **F** with **R**, and prov with req.

Problem (FixResMaxFun)**.** Given $r \in \mathbf{R}$, find the implementations in **I** that requires r (or lower) and provide the maximal functionality, or provide a proof that there are none:

$$\begin{cases} \text{using} & i \in \mathbf{I}, \\ \text{Max}_{\preceq_\mathbf{F}} & f, \\ \text{s.t.} & f = \mathsf{prov}(i), \\ & r \succeq_\mathbf{R} \mathsf{req}(i). \end{cases}$$

Another type of query is: "Given a lower bound on the functionality f and an upper bound on the costs f, what are the feasible implementations?

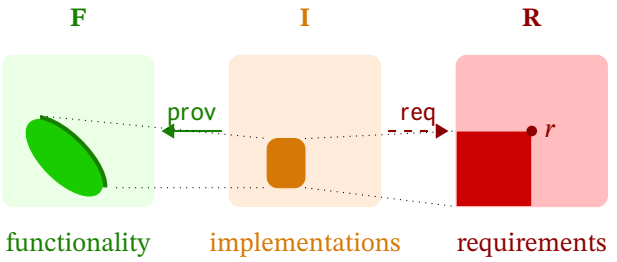


Figure 58: Graphical representation of FixResMaxFun.

Problem (FeasibleImp). Given $f \in \mathbf{F}$ and $r \in \mathbf{R}$, find the implementations in $\mathbf{I}$ that requires r (or lower) and provide f (or higher)

$$\begin{cases} \text{using} & i \in \mathbf{I}, \\ \text{s.t.} & f \preceq_{\mathbf{F}} \mathsf{prov}(i), \\ \text{s.t.} & \mathsf{prov}(i) \preceq_{\mathbf{R}} \mathsf{req}(i), \end{cases} \tag{2}$$

Another variation is to find only whether there are feasible solutions or not.

Problem (Feasibility). Given $f \in \mathbf{F}$ and $r \in \mathbf{R}$, find if (2) is feasible.

3.5 Co-design problems

A "co-design problem" is defined as a *multigraph* of design problems.

Definition 3.17 (Co-design problem with implementation)
A *co-design problem with implementation* (CDPI) is a tuple $\langle \mathbf{F}, \mathbf{R}, \langle \mathbf{V}, \mathbf{A} \rangle \rangle$, where $\mathbf{F}$ and $\mathbf{R}$ are two posets, and $\langle \mathbf{V}, \mathbf{A} \rangle$ is a multigraph of DPIs. Each node $\mathbf{d} \in \mathbf{V}$ is a DPI $\mathbf{d} = \langle \mathbf{F}_{\mathbf{d}}, \mathbf{R}_{\mathbf{d}}, \mathbf{I}_{\mathbf{d}}, \mathsf{prov}_{\mathbf{d}}, \mathsf{req}_{\mathbf{d}} \rangle$. An edge is a tuple $\langle \langle \mathbf{d}_1, i_1 \rangle, \langle \mathbf{d}_2, j_2 \rangle \rangle$, where $\mathbf{d}_1, \mathbf{d}_2 \in \mathbf{V}$ are two nodes and i_1 and j_2 are the indices of the components of the functionality and resources to be connected, and it holds that $\pi_{i_1} \mathbf{R}_{\mathbf{d}_1} = \pi_{j_2} \mathbf{F}_{\mathbf{d}_2}$ (Fig. 59).

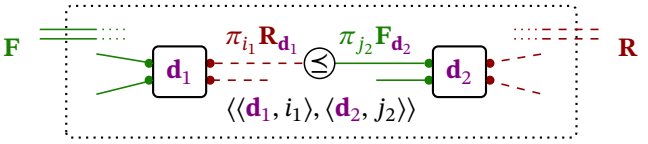


Figure 59: Co-design problem as a multigraph of design problems.

A CDPI is equivalent to a DPI with an implementation space $\mathbf{I}$ that is a subset of the product of implementation spaces of each design problem $\prod_{\mathbf{d} \in \mathbf{V}} \mathbf{I}_{\mathbf{d}}$, and contains only the tuples that satisfy the co-design constraints. An implementation tuple $i \in \prod_{\mathbf{d} \in \mathbf{V}} \mathbf{I}_{\mathbf{d}}$ belongs to $\mathbf{I}$ iff it respects all functionality–resources constraints on

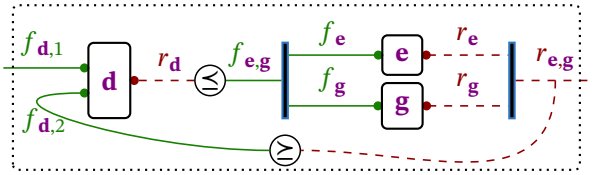


Figure 60: Example of interconnection of 3 DPs

the edges, in the sense that, for all edges $\langle\langle \mathbf{d}_1, i_1\rangle, \langle \mathbf{d}_2, j_2\rangle\rangle$ in $\mathbf{A}$, it holds that

$$\pi_{i_1}\mathsf{req}_{\mathbf{d}_1}(\pi_{\mathbf{d}_1} i) \preceq \pi_{j_2}\mathsf{prov}_{\mathbf{d}_2}(\pi_{\mathbf{d}_2} i).$$

The posets $\mathbf{F}, \mathbf{R}$ for the entire CDPI are the products of the functionality and resources of the nodes that remain *unconnected*. For a node $\mathbf{d}$, let $\mathsf{UF}_{\mathbf{d}}$ and $\mathsf{UR}_{\mathbf{d}}$ be the set of unconnected functionalities and resources. Then $\mathbf{F}$ and $\mathbf{R}$ for the CDPI are defined as the product of the unconnected functionality and resources of all DPIs: $\mathbf{F} = \prod_{\mathbf{d}\in\mathbf{V}} \prod_{j\in\mathsf{UF}_{\mathbf{d}}} \pi_j \mathbf{F}_{\mathbf{d}}$ and $\mathbf{R} = \prod_{\mathbf{d}\in\mathbf{V}} \prod_{i\in\mathsf{UR}_{\mathbf{d}}} \pi_i \mathbf{R}_{\mathbf{d}}$. The maps prov and req return the values of the unconnected functionality and resources:

$$\mathsf{prov} : i \mapsto \prod_{\mathbf{d}\in\mathbf{V}} \prod_{j\in\mathsf{UF}_{\mathbf{d}}} \pi_j \mathsf{prov}_{\mathbf{d}}(\pi_{\mathbf{d}} i),$$
$$\mathsf{req} : i \mapsto \prod_{\mathbf{d}\in\mathbf{V}} \prod_{i\in\mathsf{UR}_{\mathbf{d}}} \pi_i \mathsf{req}_{\mathbf{d}}(\pi_{\mathbf{d}} i).$$

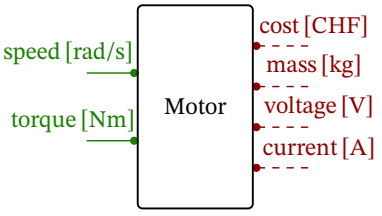


Figure 61: Design problem for the electric motor.

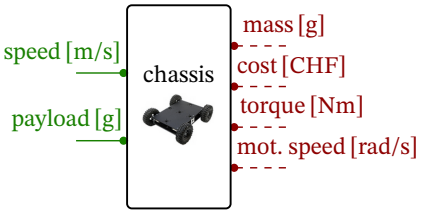


Figure 62: Design problem for the chassis.

Example 3.18. The CDPI in Fig. 60 is the interconnection of 3 DPs $\mathbf{d}, \mathbf{e}, \mathbf{g}$. The implementation space is a subset of the product

$$\mathbf{I}_{\mathbf{d}} \times \mathbf{I}_{\mathbf{e}} \times \mathbf{I}_{\mathbf{g}}.$$

The elements $\langle i_{\mathbf{d}}, i_{\mathbf{e}}, i_{\mathbf{g}}\rangle$ that are feasible are the ones that respect the following constraints:

1. Functionality and resources of each DPI are given by their implementation:

$$r_{\mathbf{d}} = \mathsf{req}(i_{\mathbf{d}}), \quad f_{\mathbf{d}} = \mathsf{prov}(i_{\mathbf{d}}),$$
$$r_{\mathbf{e}} = \mathsf{req}(i_{\mathbf{e}}), \quad f_{\mathbf{e}} = \mathsf{prov}(i_{\mathbf{e}}),$$
$$r_{\mathbf{g}} = \mathsf{req}(i_{\mathbf{g}}), \quad f_{\mathbf{g}} = \mathsf{prov}(i_{\mathbf{g}}).$$

2. Wiring constraints:

$$\langle f_{\mathbf{d}_1}, f_{\mathbf{d}_2}\rangle = f_{\mathbf{d}},$$
$$r_{\mathbf{e},\mathbf{g}} = \langle r_{\mathbf{e}}, r_{\mathbf{g}}\rangle,$$
$$f_{\mathbf{e},\mathbf{g}} = \langle f_{\mathbf{e}}, f_{\mathbf{g}}\rangle.$$

3. Co-design constraints:

$$r_{\mathbf{e},\mathbf{g}} \preceq f_{\mathbf{d}_2},$$
$$r_{\mathbf{d}} \preceq f_{\mathbf{e},\mathbf{g}}.$$

Recursive constraints

Consider the co-design of chassis plus motor. The design problem for a motor has speed and torque as the provided functionality (what the motor must provide), and cost, mass, voltage, and current as the required resources (Fig. 61).

For the chassis (Fig. 62), the provided functionality is parameterized by the mass of the payload and the cost, total mass, and what the chassis needs from its motor(s), such as speed and torque.

The two design problems can be connected at the edges for torque and speed, as in Fig. 63. The semantics is that the chassis needs *at least* the torque and speed provided by the motor.

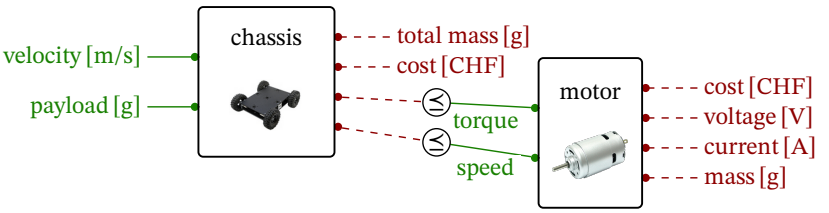


Figure 63: Series interconnection of chassis and motor design problems.

Resources can be summed together using a trivial design problem corresponding to the relation:

$$f_1 + f_2 \preceq r.$$

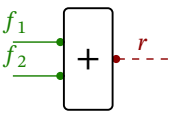


Figure 64: Sum design problem.

A co-design problem might contain recursive co-design constraints. For example, if we set the payload to be transported to be the sum of the motor mass plus some extra payload, a cycle appears in the graph (Fig. 66).

Abstraction This formalism makes it easy to abstract away the details in which we are not interested. Once a diagram like Fig. 66 is obtained, we can draw a box around it and consider the abstracted problem (Fig. 65). The idea of interconnections and abstraction will be made more precise in Chapter 5.

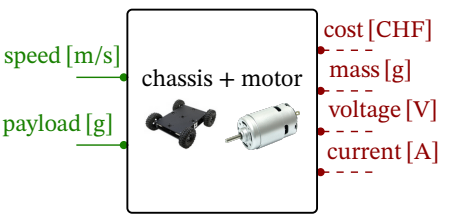


Figure 65: Abstraction of "chassis plus motor" design problem.

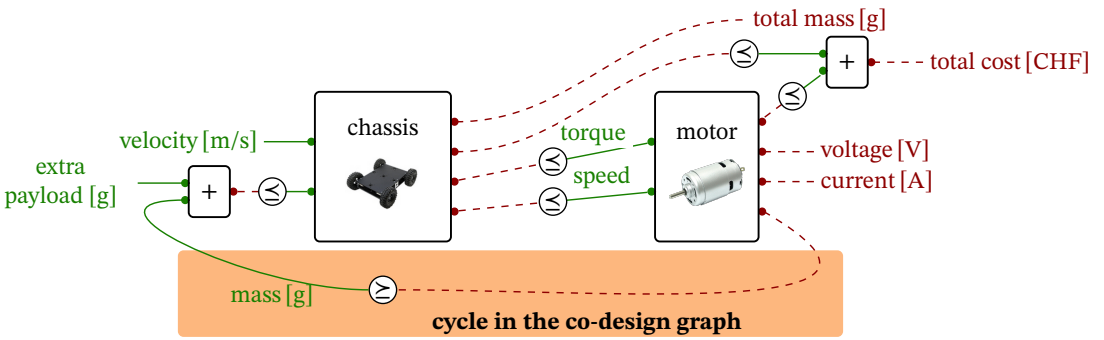


Figure 66: Example of cycle in a co-design diagram.

Feasibility 4

Nature uses as little as possible for anything.
—Johannes Kepler[9]

In the previous chapter we have introduced *design problems with implementations*. Those describe relations among three quantities: functionalities, resources, and implementations.

For the sake of computation, one can simplify the model and directly obtain a *feasibility* relation between functionality and resources. In this chapter, we first present the concept of design problems as *monotone* maps (Section 4.1), and then show how one can practically populate the related feasibility relations (Section 4.2). We close the chapter with a series of exampels from different domains.

4.1 Design problems as monotone maps

A DPI (Def. 3.3) describes a relation between three entities: $\mathbf{F}$, $\mathbf{R}$, $\mathbf{I}$. In this chapter, we show how one can consider a DPI just in the relation between $\mathbf{F}$ and $\mathbf{R}$ (a "DP").

Recall how the problem Feasibility was defined in Section 3.4. Given a particular functionality f and resource r, we would like to know whether they are feasible.

This is a function from $\mathbf{F} \times \mathbf{R}$ to **Bool**:

$$g : \mathbf{F} \times \mathbf{R} \to \mathbf{Bool}.$$

The value $g(f, r)$ is the answer to the question "is the functionality f feasible with resources r?". Due to how the problem is defined, we know that

1. If f is feasible with r, then any $f' \preceq_{\mathbf{F}} f$ is feasible with r.
2. If f is feasible with r, then f is feasible with any $r' \succeq_{\mathbf{R}} r$.

Therefore, we can conclude that g is monotone in the second argument r, and antitone in the first argument f.

It is going to be convenient to have functions that are monotone in both arguments, and not mixed monotone/antitone. Instead of considering a map from $\mathbf{F} \times \mathbf{R}$ to **Bool**, we can turn things around and look at a map $\mathbf{d}$ from $\mathbf{F}^{\mathrm{op}} \times \mathbf{R}$ to **Bool**, defined as $\mathbf{d}(f, r) = g(f^*, r)$. Because we use $\mathbf{F}^{\mathrm{op}}$ rather than $\mathbf{F}$, the map $\mathbf{d}$ is monotone.

The feasibility map $\mathbf{d}$ has now forgotten everything about implementations;

[9] Kepler was a german astronomer, mathematician, astrologer, and philosopher, and a key figure of the 17th century Scientific Revolution.

however, it does contain all the information we need to solve co-design feasibility problems.

Definition 4.1 (Design Problem)
A *design problem* (DP) is a tuple $\langle \mathbf{F}, \mathbf{R}, \mathbf{d} \rangle$, where $\mathbf{F}$, $\mathbf{R}$ are posets and $\mathbf{d}$ is a monotone map of the form

$$\mathbf{d} : \mathbf{F}^{\mathrm{op}} \times \mathbf{R} \to_{\mathbf{Pos}} \mathbf{Bool}.$$

The notation $\to_{\mathbf{Pos}}$ means "a morphism in **Pos**", and will be clear in Chapter 6.

We will also use the notation $\mathbf{d} : \mathbf{F} \nrightarrow \mathbf{R}$ for design problems, in order to emphasize how we think of them as morphisms in a category. This will be explained in Chapter 6.

Remark 4.2. Given a DPI $\langle \mathbf{F}, \mathbf{R}, \mathbf{I}, \mathsf{prov}, \mathsf{req} \rangle$ it is always possible to obtain the following DP:

$$\begin{aligned} \mathbf{d} : \mathbf{F}^{\mathrm{op}} \times \mathbf{R} &\to_{\mathbf{Pos}} \mathbf{Bool}, \\ \langle f^*, r \rangle &\mapsto \exists i \in \mathbf{I} : (f \preceq_{\mathbf{F}} \mathsf{prov}(i)) \wedge (\mathsf{req}(i) \preceq_{\mathbf{R}} r). \end{aligned}$$

Evaluating this DP is the same as asking whether the set

$$\{i \in \mathbf{I} : (f \preceq_{\mathbf{F}} \mathsf{prov}(i)) \wedge (\mathsf{req}(i) \preceq_{\mathbf{R}} r)\}$$

is empty or not.

Lemma 4.3. Given any monotone map $g : \mathbf{F} \to_{\mathbf{Pos}} \mathbf{R}$, we can turn it into a design problem

$$\mathbf{d}_g : \mathbf{F}^{\mathrm{op}} \times \mathbf{R} \to_{\mathbf{Pos}} \mathbf{Bool}$$

via the following recipe. Set

$$\mathbf{d}_g(f^*, r) = \top \quad \text{if and only if} \quad g(f) \preceq r.$$

$\mathbf{d}_g$, as defined above, is indeed a design problem when g is a monotone map.

See proof on page 206.

DPs as upper sets

Definition 4.4 (Feasible set of a design problem)
We define the *feasible set* $\mathbf{K}_{\mathbf{d}}$ of a design problem

$$\mathbf{d} : \mathbf{F}^{\mathrm{op}} \times \mathbf{R} \to_{\mathbf{Pos}} \mathbf{Bool}$$

as the subset of $\mathbf{F}^{\mathrm{op}} \times \mathbf{R}$ for which $\mathbf{d}$ is the *indicator function*, that is

$$\mathbf{K}_{\mathbf{d}} = \{\langle f^*, r\rangle \in \mathbf{F}^{\mathrm{op}} \times \mathbf{R} \mid \mathbf{d}(f^*, r) = \top\}.$$

Note that the feasibility set $\mathbf{K}_{\mathbf{d}}$ of a design problem $\mathbf{d} : \mathbf{F}^{\mathrm{op}} \times \mathbf{R} \to_{\mathbf{Pos}} \mathbf{Bool}$ is a binary relation $\mathbf{K}_{\mathbf{d}} \subseteq \mathbf{F}^{\mathrm{op}} \times \mathbf{R}$. There is a one-to-one correspondence between functions $g : \mathbf{A} \times \mathbf{B} \to \mathbf{Bool}$ and binary relations $R : \mathbf{A} \to \mathbf{B}$.

An analogous correspondence holds in the context of design problems:

Lemma 4.5. There is a one-to-one correspondence between DPs $\mathbf{d} : \mathbf{F}^{\mathrm{op}} \times \mathbf{R} \to_{\mathbf{Pos}} \mathbf{Bool}$ and upper sets $\mathbf{K}_{\mathbf{d}} \subseteq \mathbf{F}^{\mathrm{op}} \times \mathbf{R}$.

See proof on page 206.

The Boolean-valued design problems we are considering in this section do not distinguish between particular implementations: they only tell us if *any* implementation or solution exists for given functionality and resources.

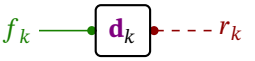


Figure 67: Diagrammatic representation of a design problem.

Diagrammatic notation We represent design problems using a diagrammatic notation. One design problem $\mathbf{d} : \mathbf{F} \nrightarrow \mathbf{R}$ is represented as a box with functionality $\mathbf{F}$ on the *left* and resources $\mathbf{R}$ on the *right* (Fig. 67).

Querying design problems

Equation Remark 4.2 on the one hand, and Remark 4.2 on the other hand, give two perspectives on the mathematical definition of what we are calling a *design problem*. When thinking about querying design problems, a correspondent definition will be important.

Let $\mathbf{d} : \mathbf{F}^{\mathrm{op}} \times \mathbf{R} \to_{\mathbf{Pos}} \mathbf{Bool}$ be a design problem. We can represent the query FixFunMinRes of this design problem as a monotone function

$$\mathbf{F} \to \langle \mathsf{USets}(\mathbf{R}), \supseteq \rangle,$$

and the query FixResMaxFun as a monotone function

$$\mathbf{R} \to \langle \mathsf{LSets}(\mathbf{F}), \subseteq \rangle.$$

Here $\mathsf{USets}(\mathbf{R})$ denotes the set of upper sets of $\mathbf{R}$ and $\mathsf{LSets}(\mathbf{F})$ denotes the set of lower sets of $\mathbf{F}$. The queries, and the associated optimization problems, will be made more precise in Chapter 7.

4.2 Populating feasibility relations

Feasibility relations offer a formidable tool to reason about design. But how can we populate them? We think of mainly three classes of models:

- Population via *catalogues* of off-the-shelf designs;
- Population via *first principles*, analytical relations;
- Population via *data-driven* methods, on-demand.

We now provide some intuition for each of them.

Catalogues

Catalogues are a very intuitive way of populating feasibility relations. Given a discrete set of components, one can identify functionalities and resources of interest, and consider the feasibility relations provided by all the components.

For instance, consider a catalogue of drones produced by DJI, as the one reported in Table 3

Table 3: Selection of drones produced by DJI.

	Spark	Phantom 3 Std	Phantom 4 Adv	Phantom 4 Pro	Mavic	Inspire
Flight time [min]	16.0	25.0	30.0	30.0	27.0	27.0
Top speed [km/h]	50.0	58.0	72.0	72.0	65.0	94.0
Range [km]	2.0	1.0	7.0	7.0	7.0	7.0
Takeoff weight [kg]	0.33	1.2	1.4	1.4	0.74	4.0
Price [USD]	499.0	499.0	1,349	1,499	999,0	2,999

We now want to come up with a design problem, and therefore need to classify the different quantities as functionalities or resources. Depending on the context, there might be multiple classifications which make sense. In this case, we want a drone to provide flight time, at a certain top speed, with a certain range, by requiring a certain takeoff weight and price. In other words, we can come up with a design problem **d** as in Fig. 68, where each feasibility relation between functionality and resources is reported in the catalogue. In particular, we can query the design problem for combinations of functionalities and resources. For instance:

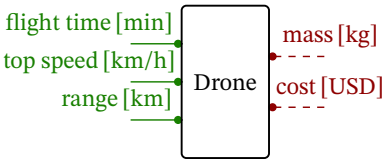


Figure 68: Design problem for a drone.

$$\mathbf{d}(\langle 20\,\text{min}, 50\,\text{km/h}, 0.5\,\text{km}\rangle^*, \langle 1.0\,\text{kg}, 1,000\,\text{USD}\rangle) = \bot,$$

since no model satisfies the properties. For instance, by increasing the allowed takeoff weight, one has:

$$\mathbf{d}(\langle 20\,\text{min}, 50\,\text{km/h}, 0.5\,\text{km}\rangle^*, \langle 1.5\,\text{kg}, 1,000\,\text{USD}\rangle) = \top,$$

since now Spark and Mavic satisfy the requirements.

First principles

Sometimes one does not just have a discrete list of components with some properties, but rather an analytical law governing the properties. In engineering this is particularly the case with physical laws. For instance, when designing a drone, we know that:

$$\text{mission energy} \geq \text{mission duration} \cdot \text{power consumption}.$$

Therefore, one could come up with a design problem to represent the energy consumption calculation, as in Fig. 69.

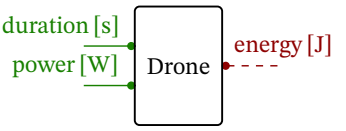


Figure 69: Design problem for the energy consumption of a drone.

First principle relations also include closed-form solutions of complex problems. In Section 4.3, we show how to populate a feasibility relation in the context of designing control systems.

Data-driven

Finally, an important, an extremely timely approach to populate feasibility relations is the data-driven one. In practice, one creates a catalogue, but not one of existing off-the-shelf designs, but rather experiments, black-box simulations, or solutions of optimization problems. In the course of this manuscript, we will show various examples for which this procedure is adequate.

4.3 Example: Linear Quadratic Gaussian Control

When controlling linear systems, a classical tool is Linear Quadratic Gaussian (LQG) control. In this example, adapted from our work [134], we will show how one can embed this popular method in the language of co-design. As we will see in the remainder of the section, LQG problems can be solved in closed-form, by solving specific equations (the so-called Riccati equations), making the arising co-design problems a case of the "first principles" ones.

Background on continuous-time LQG control

First, we recall the definition of infinite-horizon LQG control.

Definition 4.6 (Continuous-time LQG control)
Given the continuous-time stochastic dynamics

$$\begin{aligned} \mathrm{d}\boldsymbol{x}_t &= \mathbf{A}\boldsymbol{x}_t \mathrm{d}t + \mathbf{B}\boldsymbol{u}_t \mathrm{d}t + \mathbf{E}\mathrm{d}\boldsymbol{w}_t \\ \mathrm{d}\boldsymbol{y}_t &= \mathbf{C}\boldsymbol{x}_t \mathrm{d}t + \mathbf{G}\mathrm{d}\boldsymbol{v}_t, \end{aligned} \tag{3}$$

where $\boldsymbol{v}_t$ and $\boldsymbol{w}_t$ are two standard Brownian processes, let $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{E}, \mathbf{G}$ be matrices of compatible dimensions and $\mathbf{W} = \mathbf{E}\mathbf{E}^*$, $\mathbf{V} = \mathbf{G}\mathbf{G}^*$ be the effective noise covariances. The continuous-time infinite-horizon *LQG problem* consists of finding a control law $\boldsymbol{u}^\star$ minimizing the quadratic cost

$$J = \lim_{T\to\infty} \frac{1}{T}\mathbb{E}\{\int_0^T \left((\boldsymbol{x}_t^\intercal \mathbf{Q}\boldsymbol{x}_t) + (\boldsymbol{u}_t^\intercal \mathbf{R}\boldsymbol{u}_t)\right) \mathrm{d}t\},$$

with $\mathbf{Q} \succeq \mathbf{0}$, $\mathbf{R} \succ \mathbf{0}$ (i.e., positive semi-definite and positive definite matrices, respectively).

Hereafter, we use the poset of Hermitian matrices introduced in Section 2.3.

Lemma 4.7. The optimal control law for the LQG problem in Def. 4.6 is $\boldsymbol{u}_t^\star = -\mathbf{K}\hat{\boldsymbol{x}}_t = -\mathbf{R}^{-1}\mathbf{B}^*\bar{\mathbf{S}}\hat{\boldsymbol{x}}_t$, where $\hat{\boldsymbol{x}}_t$ is the unbiased minimum-variance estimate of $\boldsymbol{x}_t$ given previous measurements and $\bar{\mathbf{S}} \succ \mathbf{0}$ solves the Riccati equation

$$\mathbf{S}\mathbf{A} + \mathbf{A}^*\mathbf{S} - \mathbf{S}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^*\mathbf{S} + \mathbf{Q} = \mathbf{0}. \tag{4}$$

The minimum cost $J^\star$ achieved by the optimal control is[10]:

$$\begin{aligned} J^\star &= \mathsf{Tr}\left(\bar{\mathbf{S}}\bar{\mathbf{\Sigma}}\mathbf{C}^*\mathbf{V}^{-1}\mathbf{C}\bar{\mathbf{\Sigma}} + \bar{\mathbf{\Sigma}}\mathbf{Q}\right) \\ &= \mathsf{Tr}\left(\bar{\mathbf{\Sigma}}\bar{\mathbf{S}}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^*\bar{\mathbf{S}} + \bar{\mathbf{S}}\mathbf{W}\right), \end{aligned}$$

where $\bar{\mathbf{\Sigma}} \succ \mathbf{0}$ is the solution of the Riccati equation

$$\mathbf{A}\mathbf{\Sigma} + \mathbf{\Sigma}\mathbf{A}^* - \mathbf{\Sigma}\mathbf{C}^*\mathbf{V}^{-1}\mathbf{C}\mathbf{\Sigma} + \mathbf{W} = \mathbf{0}. \tag{5}$$

Co-design formalization

To formalize the LQG control problem as a design problem we first define two performance metrics. From a continuous-time LQG problem, we define the stationary *tracking error* P_{track} and *control effort* P_{effort}:

$$\begin{aligned} P_{\text{track}} &= \lim_{t\to\infty} \mathbb{E}\{\boldsymbol{x}_t^\intercal \mathbf{Q}\boldsymbol{x}_t\}, \\ P_{\text{effort}} &= \lim_{t\to\infty} \mathbb{E}\{\boldsymbol{u}_t^\intercal \mathbf{R}\boldsymbol{u}_t\}. \end{aligned}$$

We are now ready to state the first central result of this example.

Theorem 4.8. The LQG problem of Def. 4.6 can be formulated as a design problem with diagrammatic form as in Fig. 70.

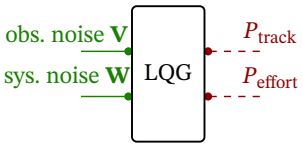


Figure 70: Co-design theorem for continuous-time LQG problems.

[10] Note that [135] contains a typo at p.188 (one extra $\mathbf{V}^{-1}$ factor). Instead, [136] has a cleaner derivation and exposition.

To prove Theorem 4.8, we show that there exists a design problem, relating functionalities $\mathbf{V}$, $\mathbf{W}$ and resources P_{track}, P_{effort}. First, by writing P_{track} and P_{effort} explicitly (Lemma 4.9), we prove their monotonicity with respect to cost weighting (Lemma 4.10). Second, we show the monotone relation characterizing the design problem (Lemma 4.12 and Lemma 4.13).

Lemma 4.9. The metrics P_{track} and P_{effort} can be written as

$$\lim_{t\to\infty} \mathbb{E}\{\boldsymbol{x}_t^\intercal \mathbf{Q}_0 \boldsymbol{x}_t\} = \mathsf{Tr}\left(\mathbf{Q}_0\left(\boldsymbol{\Sigma} + \mathbf{F}\right)\right),$$
$$\lim_{t\to\infty} \mathbb{E}\{\boldsymbol{u}_t^\intercal \mathbf{R}_0 \boldsymbol{u}_t\} = \mathsf{Tr}\left(\mathbf{S}\mathbf{B}^*\mathbf{R}^{-1}\mathbf{R}_0\mathbf{R}^{-1}\mathbf{B}\mathbf{S}\mathbf{F}\right),$$

where $\boldsymbol{\Sigma}$ solves (5), $\mathbf{F}$ solves the Lyapunov equation

$$(\mathbf{A} - \mathbf{B}\mathbf{K})\,\mathbf{F} + \mathbf{F}\,(\mathbf{A} - \mathbf{B}\mathbf{K})^* + \mathbf{L}\mathbf{V}\mathbf{L}^* = \mathbf{0}, \tag{6}$$

$\mathbf{S}$ solves (4) and $\mathbf{L} = \boldsymbol{\Sigma}\mathbf{C}^*\mathbf{V}^{-1}$ is the Kalman gain.

See proof on page 207.

Given explicit forms, we can now show that they are characterized by monotonic relations.

Lemma 4.10. Let $\mathbf{Q}(\alpha) = \alpha\mathbf{Q}_0$ and $\mathbf{R}(\alpha) = \frac{1}{\alpha}\mathbf{R}_0$, $\alpha \in \mathbb{R}_+$. Let $\boldsymbol{u}^\star(\alpha)$ be the solution of the LQG problem with $\mathbf{Q}(\alpha)$ and $\mathbf{R}(\alpha)$. Then, under optimal control one has:

- $P_{\text{track}}(\alpha)$ is decreasing with α increasing.
- $P_{\text{effort}}(\alpha)$ is increasing with α increasing.

See proof on page 207.

Intuitively, by increasing α we increase the penalty for the tracking error of the control. For this reason, P_{track} decreases and P_{effort} increases. We now want to assess the effect of the system and observation noises on the optimal control.

Lemma 4.11 (Lemma 3, [137])**.** Let $\mathbf{A}, \mathbf{B} \succ \mathbf{0}$, $\mathbf{B} - \mathbf{A} \succeq \mathbf{0}$. Then, $\mathbf{A}^{-1}, \mathbf{B}^{-1} \succ \mathbf{0}$ and $\mathbf{A}^{-1} - \mathbf{B}^{-1} \succeq \mathbf{0}$.

Lemma 4.12. The solution of (5) is monotonic in $\mathbf{V}$ and $\mathbf{W}$, i.e., $\langle \mathbf{V}, \mathbf{W}\rangle \preceq \langle \mathbf{V}', \mathbf{W}'\rangle \implies \boldsymbol{\Sigma}(\mathbf{V}, \mathbf{W}) \preceq \boldsymbol{\Sigma}(\mathbf{V}', \mathbf{W}')$.

See proof on page 208.

Lemma 4.13. Consider the situation of Lemma 4.10:

- Fix P_{track}. P_{effort} is monotonic in $\mathbf{W}$ and in $\mathbf{V}$.
- Fix P_{effort}. P_{track} is monotonic in $\mathbf{W}$ and in $\mathbf{V}$.

See proof on page 208.

This shows that the more uncertain the observations and the system dynamics are, the larger the control effort and tracking error will be, and concludes the proof of Theorem 4.8.

The presented DPI precisely assesses the feasibility relation between control effort, tracking error, system noise, and the observation noise. This design problem can be manipulated by taking the "op" of a quantity, and moving it on the other side of the diagram. For instance, we can think of the observation noise as resource, by switching its meaning to information (i.e., from noise to information matrix).

Dealing with delays We now show the ability of our formalism to capture the influence of delays on the system.

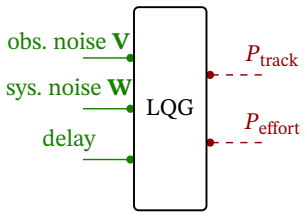


Figure 71: Co-design theorem for continuous-time LQG problems with delays.

Theorem 4.14. A continuous-time LQG problem with observation and computation delays $(d_{\text{obs}}, d_{\text{comp}})$ can be formulated as a design problem with diagram as in Fig. 71.

To establish the effect of a nuisance, we follow what we call the *substitution principle*. If in the case in which the nuisance was "lower" the controller could simulate a "higher" nuisance, then we have monotonicity. If we had a smaller delay, we could simulate a larger one by adding it artificially. Hence, control effort and tracking error of the optimal control strategy cannot decrease with larger delay.

Visualization via Pareto fronts We want to give a visual interpretation of the presented results. For the scalar case of Def. 4.6 we can derive P_{track} and P_{effort} in closed-form:

$$P_{\text{track}}(q_0) = q_0\left(\bar{\sigma} + \frac{(\bar{\sigma}c)^2}{2v\sqrt{a^2 + \alpha^2 q_0 b^2/r_0}}\right),$$

$$P_{\text{effort}}(r_0) = \frac{r_0\bar{\sigma}^2c^2}{2b^2v}\frac{\left(a + \sqrt{a^2 + \alpha^2 b^2 q_0/r_0}\right)^2}{\sqrt{a^2 + \alpha^2 b^2 q_0/r_0}},$$

where variables are in lower case since they represent scalar quantities. We can compute their limits, by fixing v and w:

$$\lim_{\alpha\to 0} P_{\text{track}}(q_0) = q_0\left(\bar{\sigma} + \frac{(\bar{\sigma}c)^2}{2va}\right), \lim_{\alpha\to\infty} P_{\text{track}}(q_0) = q_0\bar{\sigma},$$

$$\lim_{\alpha\to 0} P_{\text{effort}}(r_0) = \frac{2r_0a(c\bar{\sigma})^2}{b^2v}, \lim_{\alpha\to\infty} P_{\text{effort}}(r_0) = \infty.$$

We can plot instances of the Pareto front $\langle P_{\text{track}}, P_{\text{effort}}\rangle$ (Fig. 75). This is the

query in which we "fix functionalities and minimize resources", where the Pareto fronts represent the best achievable control performances, given specific v and w. Alternatively, we could ask to "fix resources and maximize functionalities". This is the query in which we fix P_{track} and P_{effort}, and observe a Pareto front of system and observation noises at which the given performance can be provided. Monotonicity is easy to see. By choosing $\langle v_1, w_1\rangle \preceq \langle v_2, w_2\rangle$ (i.e., $v_1 \leq v_2$ and $w_1 \leq w_2$), we see that both P_{track} and P_{effort} increase, and hence that the Pareto fronts dominate each other. We can derive similar results also for the digital case.

Background on digital LQG

We define the infinite-horizon discrete-time LQG control problem.

Definition 4.15 (Discrete-time LQG control)
Consider the discrete-time stochastic dynamics

$$\begin{aligned}\boldsymbol{x}_k &= \mathbf{A}_{\mathrm{d}}\boldsymbol{x}_k + \mathbf{B}_{\mathrm{d}}\boldsymbol{u}_k + \mathbf{E}_{\mathrm{d}}\boldsymbol{w}_k\\ \boldsymbol{y}_k &= \mathbf{C}_{\mathrm{d}}\boldsymbol{x}_k + \mathbf{G}_{\mathrm{d}}\boldsymbol{v}_k,\end{aligned}$$

where $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ are two standard Brownian processes and $\mathbf{W}_{\mathrm{d}} = \mathbf{E}\mathbf{E}^*$, $\mathbf{V}_{\mathrm{d}} = \mathbf{G}\mathbf{G}^*$ the noise covariances. The *discrete-time* infinite-horizon LQG problem consists of finding a control law $\boldsymbol{u}^\star$ which minimizes the quadratic cost

$$J_{\mathrm{d}} = \lim_{N\to\infty}\frac{1}{N}\mathbb{E}\{\textstyle\sum_{i=0}^{N-1}\left(\boldsymbol{x}_i^{\intercal}\mathbf{Q}_{\mathrm{d}}\boldsymbol{x}_i + \boldsymbol{u}_i^{\intercal}\mathbf{R}_{\mathrm{d}}\boldsymbol{u}_i\right)\},$$

where $\mathbf{Q}_{\mathrm{d}} \succeq \mathbf{0}$, $\mathbf{R}_{\mathrm{d}} \succ \mathbf{0}$.

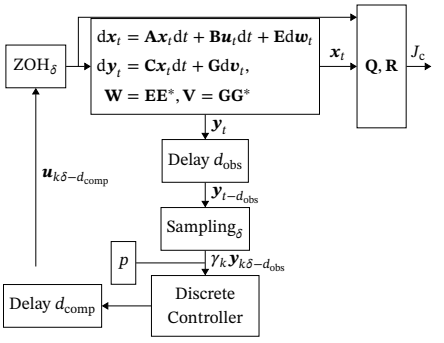


Figure 72: LQG digital control with observation and computation delays, sampling and ZOH.

We want to show that the relations found in the previous section hold for the case of digital LQG control, where we want to control a continuous-time system using a digital controller.

Lemma 4.16. Consider a continuous-time LQG problem, where observations are sampled with period δ and processed by a digital controller to produce a control input (Fig. 72). The input is reconstructed using ZOH with period δ. We can find:

$$\mathbf{A}_{\mathrm{d}} = e^{\mathbf{A}\delta}, \mathbf{B}_{\mathrm{d}} = \int_0^\delta e^{\mathbf{A}s}\mathbf{B}\mathrm{d}s, \mathbf{C}_{\mathrm{d}} = \mathbf{C}, \mathbf{Q}_{\mathrm{d}} = \int_0^\delta e^{\mathbf{A}^{\intercal}s}\mathbf{Q}e^{\mathbf{A}s}\mathrm{d}s,$$
$$\mathbf{R}_{\mathrm{d}} = \int_0^\delta\left(\left(\int_0^s e^{\mathbf{A}t}\mathbf{B}\mathrm{d}t\right)\mathbf{Q}\left(\int_0^s e^{\mathbf{A}t}\mathbf{B}\mathrm{d}t\right)^{\intercal} + \mathbf{R}\right)\mathrm{d}s.$$

such that the optimal cost of this controller coincides with the optimal cost J_{d} in Def. 4.15.

See proof on page 208.

Lemma 4.17. The optimal control law for a digital LQG problem is $\boldsymbol{u}_k^\star = -\mathbf{K}_\mathrm{d}\hat{\boldsymbol{x}}_k = -\left(\mathbf{B}_\mathrm{d}^*\bar{\mathbf{P}}\mathbf{B}_\mathrm{d} + \mathbf{R}_\mathrm{d}\right)^{-1}\mathbf{B}_\mathrm{d}^*\bar{\mathbf{P}}\mathbf{A}_\mathrm{d}\hat{\boldsymbol{x}}_k$, where $\bar{\mathbf{P}} \succ \mathbf{0}$ is the solution of the Riccati equation

$$\mathbf{P} = \mathbf{A}_\mathrm{d}^*\mathbf{P}\mathbf{A}_\mathrm{d} - (\mathbf{A}_\mathrm{d}^*\mathbf{P}\mathbf{B}_\mathrm{d})(\mathbf{R}_\mathrm{d} + \mathbf{B}_\mathrm{d}^*\mathbf{P}\mathbf{B}_\mathrm{d})^{-1}(\mathbf{B}_\mathrm{d}^*\mathbf{P}\mathbf{A}_\mathrm{d}) + \mathbf{Q}_\mathrm{d}. \tag{7}$$

The minimum cost $J_\mathrm{d}^\star$ achieved by the optimal control is[11]:

$$\begin{aligned} J_\mathrm{d}^\star &= \mathsf{Tr}\left(\bar{\mathbf{P}} + \mathbf{Q}_\mathrm{d}\right)\mathbf{L}_\mathrm{d}\left(\mathbf{C}_\mathrm{d}\bar{\mathbf{\Gamma}}\mathbf{C}_\mathrm{d}^* + \mathbf{V}_\mathrm{d}\right)\mathbf{L}_\mathrm{d}^* + \mathbf{Q}_\mathrm{d}\bar{\mathbf{\Gamma}} \\ &= \mathsf{Tr}\left((\mathbf{Q}_\mathrm{d} + \bar{\mathbf{P}})\mathbf{W}_\mathrm{d} + \bar{\mathbf{\Gamma}}\mathbf{K}_\mathrm{d}^*\left(\mathbf{R}_\mathrm{d} + \mathbf{B}_\mathrm{d}^*(\mathbf{Q}_\mathrm{d} + \bar{\mathbf{P}})\mathbf{B}_\mathrm{d}\right)\mathbf{K}_\mathrm{d}\right), \end{aligned}$$

where $\bar{\mathbf{\Gamma}} \succ \mathbf{0}$ is the solution of the Riccati equation

$$\mathbf{\Gamma} = \mathbf{A}_\mathrm{d}\mathbf{\Gamma}\mathbf{A}_\mathrm{d}^* - \mathbf{A}_\mathrm{d}\mathbf{\Gamma}\mathbf{C}^*(\mathbf{C}\mathbf{\Gamma}\mathbf{C}^* + \mathbf{V}_\mathrm{d})^{-1}\mathbf{C}\mathbf{\Gamma}\mathbf{A}_\mathrm{d}^* + \mathbf{W}_\mathrm{d}. \tag{8}$$

Co-design formalization

We are now ready to state the result for digital LQG.

Theorem 4.18. The LQG problem of Def. 4.15 with fixed sampling period δ can be formulated as a design problem with diagrammatic form as in Fig. 71.

To prove the theorem, we proceed as we did for the continuous-time case.

Lemma 4.19. One can write

$$\begin{aligned} \lim_{k\to\infty} \mathbb{E}\{\boldsymbol{x}_k^\intercal\mathbf{Q}\boldsymbol{x}_k\} &= \mathsf{Tr}\,\mathbf{Q}(\mathbf{\Gamma} + \mathbf{F}), \\ \lim_{k\to\infty} \mathbb{E}\{\boldsymbol{u}_k^\intercal\mathbf{R}\boldsymbol{u}_k\} &= \mathsf{Tr}\,\mathbf{K}_\mathrm{d}^\intercal\mathbf{R}\mathbf{K}_\mathrm{d}\mathbf{F}, \end{aligned}$$

where $\mathbf{\Gamma}$ solves (8) and $\mathbf{F}$ solves the Lyapunov equation

$$\mathbf{F} = \mathbf{L}_\mathrm{d}\left(\mathbf{C}\mathbf{\Gamma}\mathbf{C}^* + \mathbf{V}_\mathrm{d}\right)\mathbf{L}_\mathrm{d}^\intercal + (\mathbf{A}_\mathrm{d} - \mathbf{B}_\mathrm{d}\mathbf{K}_\mathrm{d})\mathbf{F}(\mathbf{A}_\mathrm{d} - \mathbf{B}_\mathrm{d}\mathbf{K}_\mathrm{d})^*, \tag{9}$$

with $\mathbf{L}_\mathrm{d} = \mathbf{\Gamma}\mathbf{C}^*(\mathbf{C}\mathbf{\Gamma}\mathbf{C}^* + \mathbf{V}_\mathrm{d})^{-1}$ discrete Kalman gain.

Lemma 4.20. Consider the LQG problem of Lemma 4.16. Let $\mathbf{Q}(\alpha) = \alpha\mathbf{Q}_0$ and $\mathbf{R}(\alpha) = \frac{1}{\alpha}\mathbf{R}_0$, $\alpha \in \mathbb{R}_+$. Let $\boldsymbol{u}^\star(\alpha)$ be the solution of the LQG problem with $\mathbf{Q}(\alpha)$ and $\mathbf{R}(\alpha)$. Then, under optimal control one has:

- $P_{\text{track}}^{\mathrm{d}}(\alpha)$ is decreasing with α increasing.
- $P_{\text{effort}}^{\mathrm{d}}(\alpha)$ is increasing with α increasing.

[11] Note that also [138] contains a typo at p. 476 (- instead of + in Eq. 7.199).

Proof. The proof is analogous to the one of Lemma 4.10. □

Lemma 4.21. Consider the situation of Lemma 4.20. One has:

▷ Fix $P^{\mathrm{d}}_{\mathrm{track}}$. $P^{\mathrm{d}}_{\mathrm{effort}}$ is monotonic in $\mathbf{W}_{\mathrm{d}}$ and in $\mathbf{V}_{\mathrm{d}}$.

▷ Fix $P^{\mathrm{d}}_{\mathrm{effort}}$. $P^{\mathrm{d}}_{\mathrm{track}}$ is monotonic in $\mathbf{W}_{\mathrm{d}}$ and in $\mathbf{V}_{\mathrm{d}}$.

See proof on page 209.

We can further extend our models for the digital case to capture differences in sampling periods and intermittent observations [139], [140].

Definition 4.22 (LQG with intermittent observations)
The *LQG problem with intermittent observations* differs from the original problem by the observations $\boldsymbol{y}'_k = \gamma_k \boldsymbol{y}_k$, where $\gamma_k \in \{0, 1\}$ is a random sequence.

Theorem 4.23. The LQG problem of Def. 4.15 with sampling of the form $\delta = 2^n \delta_0$ and intermittent observations can be formulated as a design problem with diagrammatic form as in Fig. 73.

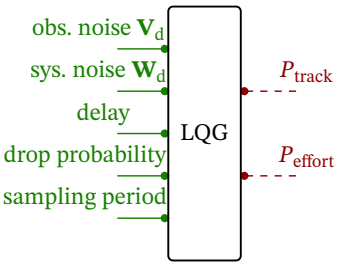


Figure 73: Co-design theorem for digital LQG problems.

To prove all cases, we can use the substitution principle. Assuming a sampling period $\delta = 2^n \delta_0$, P_{track} and P_{effort} are monotonic with n.[12] On the other hand, if the controller is given a set of observations, it can simulate having less (i.e., an higher drop probability p), by artificially deleting selected observations. Therefore, the control effort and tracking error cannot decrease with increasing p.

Practical considerations

In this section, we have seen how one can embed both continuous-time and digital LQG in the language of co-design. In particular, we were able to formulate co-design theorems involving important quantities in control theory, which hold for a generic class of systems. But how do we use these concepts in practice, when populating the respective feasibility relations? Let's see these tools at work with the example of a drone.

Controlling a drone Let's consider the case in which one wants to design a control strategy for a drone which needs to align itself with a goal (Fig. 74). We define the state of the robot as $\langle \theta_t, \omega_t \rangle$, where θ_t is its heading and ω_t its angular speed. The heading of the goal at time t is denoted by θ^{g}_t. The control is

[12] Note that using δ of this form corresponds to the case where the information available is a subset decreasing with n. We are not stating this result in general. This would require a deeper discussion and several assumptions about the system (e.g., about oscillatory behavior). This is an open problem [141], [142].

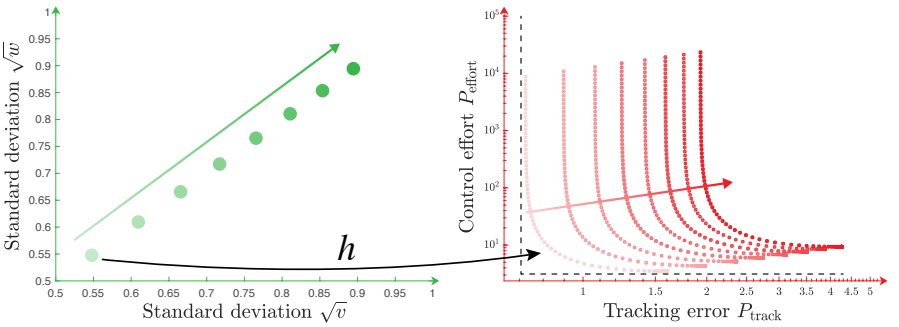


Figure 75: Monotonic relation between functionalities and upper-sets of resources.

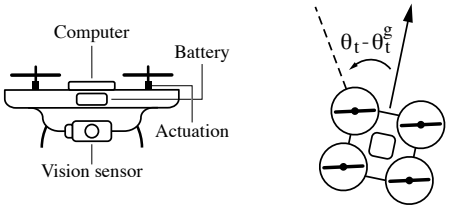


Figure 74: Drone which needs to align with a goal.

the resulting torque τ_t. The dynamics of the heading are given by the differential equations

$$\begin{aligned} \mathrm{d}\theta_t &= \omega_t \mathrm{d}t, \\ \mathrm{d}\omega_t &= \frac{\tau_t}{I}\mathrm{d}t + \mathrm{d}w_t, \end{aligned}$$

where I is the moment of inertia of the drone about its rotational axis, and w_t is a Brownian process with intensity σ_w^2.

We assume Gaussian observations of the relative bearing of drone and goal $y_t = (\theta_t - \theta_t^{\mathrm{g}}) + v_t$, where v_t is white Gaussian noise with intensity σ_v^2, describing measurement uncertainty. Assuming that the goal is far away, θ_t^{g} is approximately constant, and w.l.o.g. we can assume it to be 0 ($y_t = \theta_t + v_t$). We pose the stationary problem of minimizing the objective

$$\lim_{T\to\infty} \frac{1}{T} \int_0^T \left(\alpha q_0 \theta_t^2 + \frac{r_0}{\alpha}\tau_t^2\right) \mathrm{d}t.$$

This is a LQG problem with $\mathbf{Q} = \mathrm{diag}(0, \alpha q_0)$, $\mathbf{R} = r_0/\alpha$, for the continuous-time system given by the system matrices:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \frac{1}{I}\begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix}, \mathbf{W} = \begin{bmatrix} 0 & 0 \\ 0 & \sigma_w^2/I \end{bmatrix}, \quad \mathbf{V} = \sigma_v^2.$$

As shown previously, we can formalize this as a design problem, which provides stability of the system up to a given system noise, requiring observations at a specific frequency and with given precision. The control law has to be implemented at a given frequency, resulting in specific control effort and tracking error. Implementations are given by the different cost weights, parametrized by α. One obtains the relationships between functionalities and resources (i.e., the model) by solving specific Riccati equations via numerical simulations (Fig. 75).

4.4 Example: Convex Optimization Problems

We can formalize the relationships characterizing convex optimization problems as design problems. In this section, we sketch the main idea behind this, and focus on the intuitions. The complete mathematical formulation of the ideas goes beyond the scope of this chapter, and is left out for brevity.

First, recall the definition of convex function:

Definition 4.24 (Convex function)
A function $f : \mathbb{R}^n \to \mathbb{R}$ is *convex* if, for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ and $\theta \in [0,1]$:

$$f(\theta \boldsymbol{x} + (1-\theta)\boldsymbol{y}) \leq \theta f(\boldsymbol{x}) + (1-\theta) f(\boldsymbol{y}).$$

Furthermore, we recall the notion of convex optimization problem.

Definition 4.25 (Convex optimization problem)
A *convex optimization problem* in standard form is given by:

$$\begin{aligned} \underset{\boldsymbol{x} \in \mathbb{R}^n}{\text{minimize}} \quad & f_0(\boldsymbol{x}) \\ \text{subject to} \quad & f_i(\boldsymbol{x}) \leq 0, \quad i = 1, \dots, m \\ & \mathbf{A}\boldsymbol{x} = \boldsymbol{b}, \end{aligned}$$

with $\mathbf{A} \in \mathbb{R}^{p \times n}$, $\boldsymbol{b} \in \mathbb{R}^p$, and $f_0 : \mathbb{R}^n \to \overline{\mathbb{R}}$, $f_i : \mathbb{R}^n \to \mathbb{R}$ convex functions.

So how to relate this to design problems? The key insight will be to define a convex optimization problem as a monotone map between three particular posets.

First, denote by $\mathbf{U}$ the poset of convex functions, where the order is given as:

$$f_1 \preceq_{\mathbf{U}} f_2 \Leftrightarrow f_2(\boldsymbol{x}) \leq f_1(\boldsymbol{x}), \forall \boldsymbol{x} \in \mathbb{R}^n, f_1, f_2 \text{ convex functions.}$$

Furthermore, consider the poset of convex sets $\mathbf{R}$ introduced in Section 2.3 (ordered by inclusion).

Then, one can state the following statement.

Lemma 4.26. A convex optimization problem as in Def. 4.25 is an antitone map

$$\begin{aligned} c : \mathbf{R} \times \mathbf{U} \quad & \to \langle \mathbb{R}, \leq \rangle, \\ \langle \mathbf{D}_f, f_0 \rangle \quad & \mapsto \hat{p}, \end{aligned}$$

where $\hat{p} = \inf_{\boldsymbol{x} \in \mathbf{D}_f}$ and $\mathbf{D}_f = \{\boldsymbol{x} \in \mathbb{R}^n : f_i(\boldsymbol{x}) \leq 0, i \in \{1, ..., m\}, \mathbf{A}\boldsymbol{x} = \boldsymbol{b}\}$.

The intuition is as follows. $\mathbf{D}_f$ is the feasible set of the optimization problem, and lives in the poset of convex sets. f_0 lives in the poset of convex functions,

and represents the objective function. $\hat{p}$ is the minimizer for the problem. By choosing a larger feasible set, one could potentially find a better (and not worse) solution. One can reason analogously for the objective function. We can visualize this explicitly for an example. Let's consider a linear program (a particular case of convex optimization problem):

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \quad & \boldsymbol{c}_i^\intercal \boldsymbol{x} \\ \text{subject to} \quad & \mathbf{G}_i \boldsymbol{x} \preceq \boldsymbol{h}_i, \quad i = 1, 2, \end{aligned}$$

For the sake of the example, consider the following parameters.

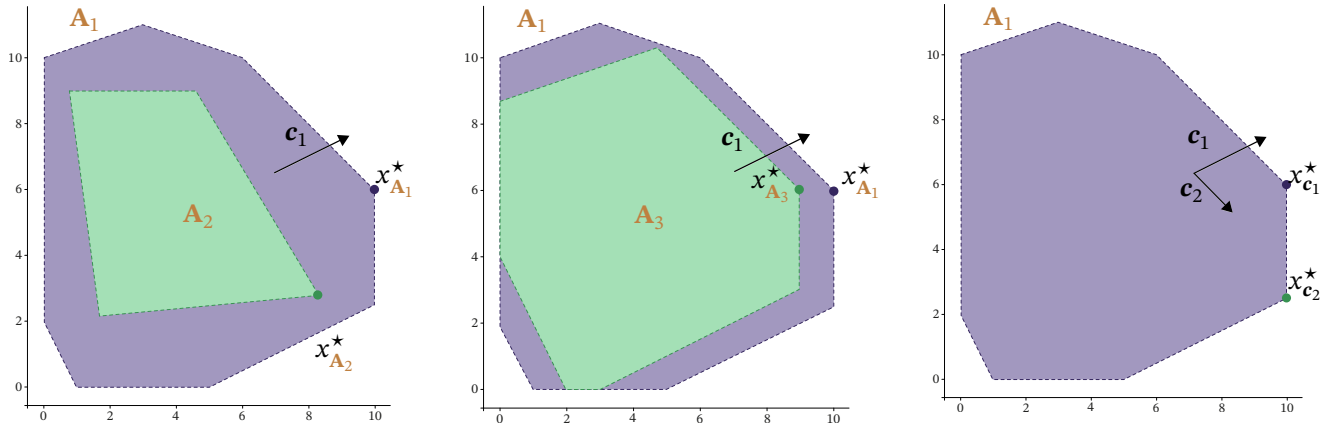
$$\mathbf{G}_1 = \begin{bmatrix} 1 & 1 \\ 1 & 3 \\ 1 & 0 \\ -1 & 3 \\ 1 & -2 \\ -2 & -1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}, \mathbf{G}_2 = \begin{bmatrix} -15 & -2 \\ 1 & -10 \\ 0 & 1 \\ 5 & 3 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}, \boldsymbol{h}_1 = \begin{bmatrix} 16 \\ 36 \\ 10 \\ 30 \\ 5 \\ -2 \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{h}_2 = \begin{bmatrix} -30 \\ -20 \\ 9 \\ 50 \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{h}_3 = \begin{bmatrix} 15 \\ 36 \\ 9 \\ 26 \\ 3 \\ -4 \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{c}_1 = \begin{bmatrix} -2 \\ -1 \end{bmatrix}, \boldsymbol{c}_2 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

The solutions for different combinations of parameters, and the relative size of sets are reported in Fig. 76.

First, we write the feasible sets:

$$\begin{aligned} \mathbf{A}_1 &= \{\boldsymbol{x} \in \mathbb{R}^n \mid \mathbf{G}_1 \boldsymbol{x} \preceq \boldsymbol{h}_1\}, \\ \mathbf{A}_2 &= \{\boldsymbol{x} \in \mathbb{R}^n \mid \mathbf{G}_2 \boldsymbol{x} \preceq \boldsymbol{h}_2\}, \\ \mathbf{A}_3 &= \{\boldsymbol{x} \in \mathbb{R}^n \mid \mathbf{G}_1 \boldsymbol{x} \preceq \boldsymbol{h}_3\}. \end{aligned}$$

Starting from Fig. 76a, we observe $\mathbf{A}_2 \preceq \mathbf{A}_1$. Solving the linear program with $\boldsymbol{c}_1$ in the objective in the two cases, returns -19.4 (using $\mathbf{A}_2$) and -26 (using $\mathbf{A}_1$), showing one aspect of the monotonicity. Moving to Fig. 76b, we observe $\mathbf{A}_3 \preceq \mathbf{A}_1$. Solving the linear program with $\boldsymbol{c}_1$ in the objective in the two cases, returns -24 (using $\mathbf{A}_3$) and -26 (using $\mathbf{A}_1$), showing monotonicity. Finally (Fig. 76c), by consdering the feasible set $\mathbf{A}_1$, and two different objectives parametrized by $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ (with $\boldsymbol{c}_1 \preceq \boldsymbol{c}_2$), obtain -26 and -7.5, respectively, showing the other aspect of monotonicity.



(a) Choose c_1, solve the LP with $\mathbf{A}_1$.

(b) Choose c_1 and $\mathbf{A}_1$, solve the LP with h_1.

(c) Choose $\mathbf{A}_1$ and h_1, solve the LP with c_1 and c_2.

Figure 76: Solutions of different linear programs to showcase monotonicity.

Co-design formalization

Given the above, we can define a design problem

$$\begin{aligned} \mathbf{d}_c : (\mathbf{R} \times \mathbf{U}) \times \langle \mathbb{R}, \leq \rangle \quad &\to_{\mathbf{Pos}} \mathbf{Bool}, \\ \langle \langle \mathbf{D}_f, f_0 \rangle^*, p \rangle \quad &\mapsto c(\mathbf{D}_f, f_0) \leq p. \end{aligned}$$

giving rise to a design problem as in Fig. 77.

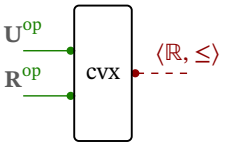


Figure 77: Design problem for a convex optimization problem.

If, in addition, the objective function f_0 is monotone, we can consider objective functions as functionalities.

In this context, implementations can be obtained by solving specific instances of convex optimization problems. This result sets the stage for further investigations, given the abundance of convex optimization problems in engineering, and the ability to embed them as design problems. Furthermore, while not explicitly verified formally, the same reasoning can be applied to other sorts of optimization problems.

4.5 Example: Assume-Guarantee Contracts

This example is based on the literature on assume-guarantee contracts. We will specifically refer to [37], and other relevant references are [115], [117]–[119], [121], [143], [144].

Background on assume-guarantee contracts

Assume-guarantee contracts are specifications for components in complex systems, with typical applications are in cyber-physical systems. Their definition leverages set-based behavioural modeling.

Definition 4.27 (Behavior)
Let $\mathbf{B}$, elements of which are called *behaviors*.

Given a set of behaviors, we can identify *components* and *properties*, as subsets of $\mathbf{B}$. The difference between components and properties lies in their usage:

- Components: sets containing behaviors which a design can display;
- Properties: collection of behaviors displaying a quality of interest.

Components and properties are connected by satisfaction.

Definition 4.28 (Satisfaction)
Given a component $\mathbf{M}$ and a property $\mathbf{P}$, we say that $\mathbf{M}$ *satisfies* $\mathbf{P}$, denoted $\mathbf{M} \vDash \mathbf{P}$, when $\mathbf{M} \subseteq \mathbf{P}$. In words, $\mathbf{M}$ satisfies $\mathbf{P}$ when all of its behaviors have a quality of interest.

Given these notions, one can define the notion of subcomponent.

Definition 4.29 (Subcomponent)
Let $\mathbf{M}_1, \mathbf{M}_2$ be components. We say that $\mathbf{M}_1$ is a *subcomponent* of $\mathbf{M}_2$, denoted $\mathbf{M}_1 \preceq \mathbf{M}_2$, when $\mathbf{M}_1 \subseteq \mathbf{M}_2$, i.e., when $\mathbf{M}_1$ satisfies all the properties of $\mathbf{M}_2$.

Note that this defines a poset on the set of all components.

Definition 4.30 (Composition of components)
Let $\mathbf{M}_1, \mathbf{M}_2$ be components. The *composition* of $\mathbf{M}_1$ and $\mathbf{M}_2$, denoted $\mathbf{M}_1 \parallel \mathbf{M}_2$, is given by

$$\mathbf{M}_1 \parallel \mathbf{M}_2 = \mathbf{M}_1 \cap \mathbf{M}_2.$$

Definition 4.31 (Assume-guarantee contract)
An *assume-guarantee contract* is $\mathcal{C} = \langle \mathbf{A}, \mathbf{G} \rangle$, where $\mathbf{A}, \mathbf{G}$ are properties.

We now introduce the concept of *environment* and *implementation*.

Definition 4.32 (Environment)
We say that a component $\mathbf{E}$ is an *environment* for a contract $\mathcal{C}$, denoted

$\mathbf{E} \vDash^{\mathbf{E}} \mathcal{C}$, if $\mathbf{E} \vDash \mathbf{A}$ (i.e., if the environment satisfies the assumptions of the contract).

Definition 4.33 (Implementation)
We say that a component $\mathbf{M}$ is an *implementation* for a contract $\mathcal{C}$, denoted $\mathbf{M} \vDash^{\mathbf{M}} \mathcal{C}$, if it satisfies the guarantees of the contract, provided that it operates in an environment of the contract, i.e., if:

$$\mathbf{M} \parallel \mathbf{E} \vDash \mathbf{G} \text{ for all } \mathbf{E} \vDash \mathbf{A}.$$

Co-design formalization

Given a contract $\mathcal{C} = \langle \mathbf{A}, \mathbf{G} \rangle$, one can generate a design problem leveraging the partial orders of power sets, introduced in Section 2.3:

$$\begin{aligned} \mathbf{d}_{\mathcal{C}} : \langle \mathsf{Pow}\, \mathbf{G}, \subseteq \rangle^{\mathrm{op}} \times \langle \mathsf{Pow}\, \mathbf{A}, \subseteq \rangle &\to_{\mathbf{Pos}} \mathbf{Bool} \\ \langle \mathbf{G}^*, \mathbf{A} \rangle &\mapsto \exists \mathbf{E}, \mathbf{M} \mid \mathbf{M} \parallel \mathbf{E} \vDash \mathbf{G} \text{ for all } \mathbf{E} \vDash \mathbf{A}. \end{aligned} \tag{10}$$

Since a contract assumes the existence of such a pair of environment/component in the literature, one can technically populare the design problem by taking the upperset in $\langle \mathsf{Pow}\, \mathbf{G}, \subseteq \rangle^{\mathrm{op}} \times \langle \mathsf{Pow}\, \mathbf{A}, \subseteq \rangle$ generated by $\langle \mathbf{G}, \mathbf{A} \rangle$.

(10) provides a valid design problem definition. In general, this is verified intuitively:

- A smaller set of guarantees will not require more assumptions;
- A larger set of assumptions will not provide a smaller set of guarantees.

In practice, take $\mathbf{d}_{\mathcal{C}}(\mathbf{G}, \mathbf{A}) = \top$. Let $\mathbf{G}' \subseteq \mathbf{G}$ and $\mathbf{A}' \supseteq \mathbf{A}$. Then $\mathbf{d}_{\mathcal{C}}(\mathbf{G}', \mathbf{A}) = \top$ and $\mathbf{d}_{\mathcal{C}}(\mathbf{G}, \mathbf{A}') = \top$.

Note that depending on the interpretation of assumptions and guarantees, one might also decide to locate both of them as resources/functionalities.

Practical considerations

For instance, assume-guarantee contracts are used in [117] to characterize the autonomy stack of an AV. The contract for the control of the vehicle involves assumptions about the environment it has to drive in, and provides safety guarantees. The implementations are given by different controllers, or different parameters/executions of the control laws. The specific feasibility relationships can be obtained by leveraging and querying dedicated software, called PACTI, developed by Incer and collaborators.

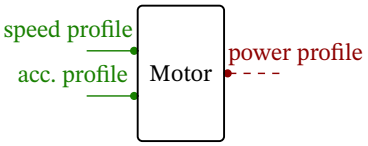


Figure 78: Design problem for the power generation process for an electric vehicle.

4.6 Example: Electric vehicle design

In the context of designing the powertrain for an electric vehicle, a crucial step is to map a particular vehicle type, together with a prescribed drive cycle, to some power request. Typically, the power is then provided by a DC/AC converter [145], [146]. Let's take an exemplary model. At each time instant t the power request can be expressed as the product of torque and angular velocity:

$$P(t) = T(t) \cdot \omega(t).$$

The torque, can be expressed as a relation of force and radius of the wheels:

$$T(t) = F(t) \cdot r,$$

where the force can be computed as

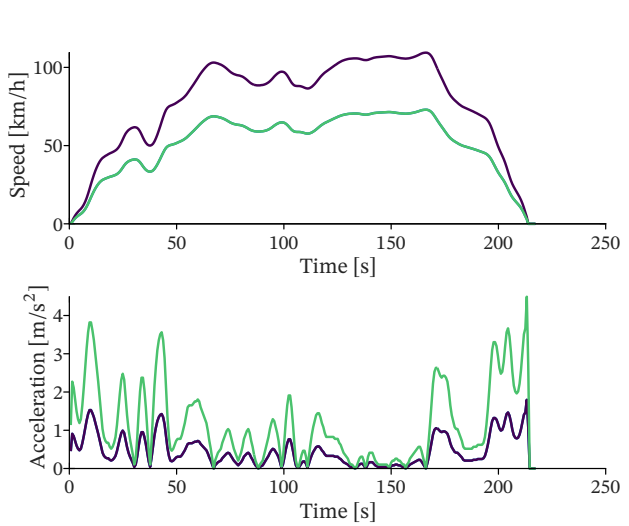
$$F(t) = m \cdot a(t) + m \cdot g \cdot \sin(\alpha(t)) + \frac{1}{2}\rho_{\text{air}} \cdot v(t)^2 \cdot C_{\text{D}} \cdot A_{\text{F}},$$

where m is the mass of the vehicle, and $v(t), a(t)$ represent the speed and acceleration provided by the drive cycle, respectively, g the gravity constant, $\alpha(t)$ the current inclination of the cycle, ρ_{air} the air density, C_{D} the aerodynamic coefficient of the car, and A_{F} the frontal area of the vehicle. Now, given that $\omega(t) = v(t)/r$, one finds the following expression relating power, velocity, and acceleration at each time instant:

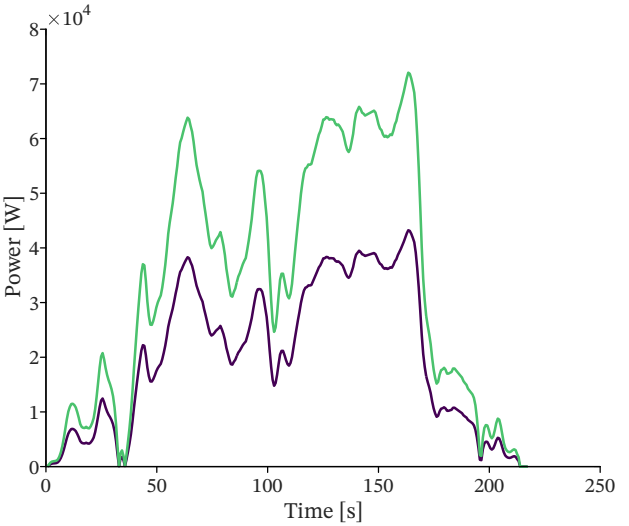
$$P(t) = v(t) \cdot \left(m \cdot (a(t) + g \cdot \sin(\alpha)) + \frac{1}{2}\rho_{\text{air}} \cdot v(t)^2 \cdot C_{\text{D}} \cdot A_{\text{F}} \right).$$

Again, by considering the partial order on maps for the signals P, v, a, one can define a design problem as in Fig. 78.

To visualize this, consider for instance the classic World Light-Duty Test Procedure (WLTP) drive cycle, including portions of urban, exurban, and highway driving. By considering an Astom Martin Cygnet, one might have speed and acceleration profiles as in Fig. 79a. Clearly, dominating profiles (for both speed and acceleration), lead to dominating power profiles (Fig. 79b).



(a) Speed and acceleration profiles of the drive cycle.



(b) Power profile of the drive cycle.

Figure 79: Monotone relationship between acceleration, speed, and power profile in the context of powertrain design.

Interconnecting design problems 5

The internal machinery of life, the chemistry of the parts, is something beautiful. And it turns out that all life is interconnected with all other life.

—Richard P. Feynman[13]

In this chapter, we will define several ways to connect design problems together. We will present series composition (Section 5.1), union and intersection (Section 5.2), parallel composition (Section 5.3), and feedback (Section 5.4). Finally, we will present a case study featuring the co-design of a drone, leveraging the defined interconnection operations (Section 5.5).

5.1 Series composition of design problems

The first and most basic way is series composition. In the rest of this thesis, we will refer to it as just "composition", for reasons that will be clear once we will have presented the categorical structure of design problems (Chapter 6).

Definition 5.1 (Series composition)
Let $\mathbf{d} : \mathbf{P} \nrightarrow \mathbf{Q}$ and $\mathbf{e} : \mathbf{Q} \nrightarrow \mathbf{R}$ be design problems. We define their *series composition* $(\mathbf{d} \fatsemi \mathbf{e}) : \mathbf{P} \nrightarrow \mathbf{R}$ as:

$$\begin{aligned} (\mathbf{d} \fatsemi \mathbf{e}) : \mathbf{P}^{\mathrm{op}} \times \mathbf{R} &\to_{\mathbf{Pos}} \mathbf{Bool}, \\ \langle p^*, r \rangle &\mapsto \bigvee_{q \in \mathbf{Q}} \mathbf{d}(p^*, q) \wedge \mathbf{e}(q^*, r). \end{aligned} \tag{11}$$

The series composition $(\mathbf{d} \fatsemi \mathbf{e})$ judges a pair $\langle p^*, r \rangle$ as feasible if and only if there exists a $q \in \mathbf{Q}$ such that $\mathbf{d}(p^*, q)$ and $\mathbf{e}(q^*, r)$ are feasible.

Given a set $\mathbf{A}$ and a map $s : \mathbf{A} \to \mathbf{Bool}$, we can define the boolean $\bigvee_{a \in \mathbf{A}} s(a)$ by

$$\bigvee_{a \in \mathbf{A}} s(a) := \begin{cases} \top & \text{if there exists } a \in \mathbf{A} \text{ for which } s(a) = \top, \\ \bot & \text{if there exists } \textit{no } a \in \mathbf{A} \text{ for which } s(a) = \top. \end{cases}$$

In (11) we could have written "$\exists_{q \in \mathbf{Q}}$" instead of "$\bigvee_{q \in \mathbf{Q}}$":

$$\exists_{q \in \mathbf{Q}} \mathbf{d}(p^*, q) \wedge \mathbf{e}(q^*, r).$$

Using $\bigvee$ form highlights the connection with an integration operation $\int_q$.

[13] Feynman was an American theoretical physicist, who received the Nobel Prize in Physics in 1965 for his contributions to quantum electrodynamics.

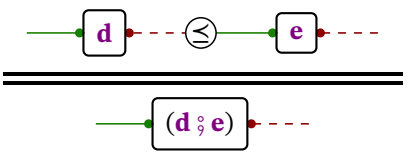


Figure 80: Series composition of design problems.

We use the same diagrammatic notation for DPs as for DPIs. We represent series composition as in Fig. 80.

One can notice the "co-design constraint" $\preceq$, which can be interpreted as follows. The resource required by a component is limited by the functionality produced by another component. This is an intuitive notion, which applies both to physical and non-physical quantities. For instance, in the context of designing a robot, one could think of an electric motor, requiring at most the power provided by a battery. Similarly, an estimator requires data of quality at most as the one provided by the used sensor.

Remark 5.2. When viewing compositions (and larger diagrams) formed from these boxes, it is tempting to interpret the boxes as input-output processes. However, that would be misleading. The arrows do not represent information flow, materials flow, or energy flow. Design problems do not represent input-output processes but rather a calculus of requirements–a requirements flow.

Let us check that, given design problems $\mathbf{d}$ and $\mathbf{e}$, their series composition $(\mathbf{d} \fatsemi \mathbf{e})$ is in fact a design problem.

Lemma 5.3. Series composition as in (11) is monotone in p and r.

See proof on page 209.

Importantly, the "⨟" operation is associative.

Lemma 5.4. The series composition operation as in (11) is associative:

$$(\mathbf{d} \fatsemi \mathbf{e}) \fatsemi \mathbf{g} = \mathbf{d} \fatsemi (\mathbf{e} \fatsemi \mathbf{g}).$$

See proof on page 209.

Because of associativity, we can write $\mathbf{d} \fatsemi \mathbf{e} \fatsemi \mathbf{g}$ without ambiguity (Fig. 81).

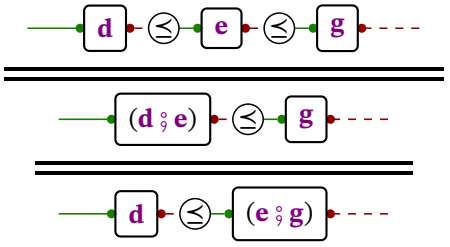


Figure 81: Associativity of the composition of design problems.

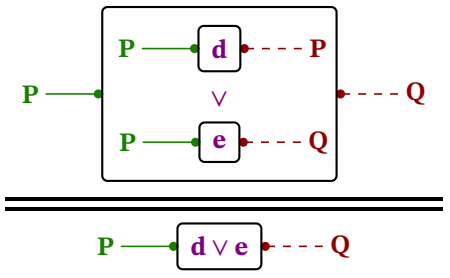


Figure 82: Diagrammatic representation of the union of design problems

.

5.2 Union and intersection of design problems

Union of design problems

Let $\mathbf{d} : \mathbf{P} \nrightarrow \mathbf{Q}$ and $\mathbf{e} : \mathbf{P} \nrightarrow \mathbf{Q}$ be design problems. We define the *union* $\mathbf{d} \vee \mathbf{e}$ to be the design problem which is feasible whenever *either* $\mathbf{d}$ or $\mathbf{e}$ is feasible. This models $\mathbf{d}$ and $\mathbf{e}$ as *interchangeable* technologies. For instance, to provide motion at the cost of energy, you might choose different mobility options.

Definition 5.5 (Union of design problems)
Given two design problems $\mathbf{d} : \mathbf{P} \nrightarrow \mathbf{Q}$ and $\mathbf{e} : \mathbf{P} \nrightarrow \mathbf{Q}$, their *union* $\mathbf{d} \vee$

$\mathbf{e} : \mathbf{P} \nrightarrow \mathbf{Q}$ is defined by

$$\begin{aligned}(\mathbf{d} \vee \mathbf{e}) : \mathbf{P}^{\mathrm{op}} \times \mathbf{Q} &\to_{\mathbf{Pos}} \mathbf{Bool},\\ \langle p^*, q\rangle &\mapsto \mathbf{d}(p^*, q) \vee \mathbf{e}(p^*, q).\end{aligned}$$

The union of design problems is represented as in Fig. 82.

Intersection of design problems

Given two design problems $\mathbf{d}, \mathbf{e} : \mathbf{P} \nrightarrow \mathbf{Q}$, we can define a design problem $\mathbf{d} \wedge \mathbf{e}$ that is feasible if and only if $\mathbf{d}$ and $\mathbf{e}$ are both feasible. We call $\mathbf{d} \wedge \mathbf{e}$ the *intersection* of $\mathbf{d}$ and $\mathbf{e}$. One interpretation of $\mathbf{d} \wedge \mathbf{e}$ is that $\mathbf{d}$ and $\mathbf{e}$ are two slightly different models of the same process, and we want to make sure that the design is conservatively feasible for both models.

Definition 5.6 (Intersection of design problems)
Given design problems $\mathbf{d} : \mathbf{P} \nrightarrow \mathbf{Q}$ and $\mathbf{e} : \mathbf{P} \nrightarrow \mathbf{Q}$, their *intersection* is denoted $(\mathbf{d} \wedge \mathbf{e}) : \mathbf{P} \nrightarrow \mathbf{Q}$, defined by:

$$\begin{aligned}(\mathbf{d} \wedge \mathbf{e}) : \mathbf{P}^{\mathrm{op}} \times \mathbf{Q} &\to_{\mathbf{Pos}} \mathbf{Bool},\\ \langle p^*, q\rangle &\mapsto \mathbf{d}(p^*, q) \wedge \mathbf{e}(p^*, q).\end{aligned}$$

The intersection of design problems is represented as in Fig. 83.

We can directly generalize the intersection $\mathbf{d} \wedge \mathbf{e}$ by allowing $\mathbf{d}$ and $\mathbf{e}$ to have different domain and codomains, $\mathbf{d} : \mathbf{P} \nrightarrow \mathbf{Q}$ and $\mathbf{e} : \mathbf{R} \nrightarrow \mathbf{S}$. We call this putting two design problems "in parallel".

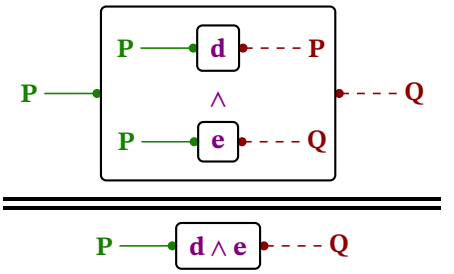


Figure 83: Diagrammatic representation of the intersection of design problems.

5.3 Parallel composition

Definition 5.7 (Parallel composition of DPs)
Given two design problems $\mathbf{d} : \mathbf{P} \nrightarrow \mathbf{Q}$ and $\mathbf{e} : \mathbf{R} \nrightarrow \mathbf{S}$, their *monoidal product* $\mathbf{d} \otimes \mathbf{e} : \mathbf{P} \times \mathbf{R} \nrightarrow \mathbf{Q} \times \mathbf{S}$ is their conjunction:

$$\begin{aligned}\mathbf{d} \otimes \mathbf{e} : (\mathbf{P} \times \mathbf{R})^{\mathrm{op}} \times (\mathbf{Q} \times \mathbf{S}) &\to_{\mathbf{Pos}} \mathbf{Bool},\\ \langle\langle p, r\rangle^*, \langle q, s\rangle\rangle &\mapsto \mathbf{d}(p^*, q) \wedge \mathbf{e}(r^*, s).\end{aligned}$$

We represent the parallel composition as in Fig. 84.

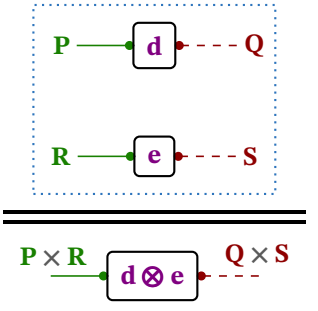


Figure 84: Monoidal product of design problems.

Remark 5.8. For $\mathbf{d} : \mathbf{P} \nrightarrow \mathbf{Q}$ and $\mathbf{e} : \mathbf{R} \nrightarrow \mathbf{S}$, the monoidal product

$$(\mathbf{d} \otimes \mathbf{e})(\langle p^*, r^*\rangle, \langle q, s\rangle)$$

is true if *both* $\mathbf{d}(p^*, q)$ and $\mathbf{e}(r^*, s)$ are true, and false otherwise.

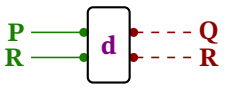


Figure 85: Design problem with a resource and a functionality of the same type.

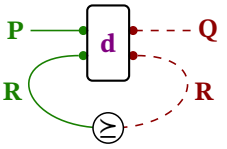


Figure 86: Closing the loop in the design problem.

5.4 Feedback

Suppose that we are given a design problem with a resource and a functionality of the same type **R** (Fig. 85). Can we "close the loop", as in the diagram reported in Fig. 86?

It turns out that we can give a well-defined semantics to this loop-closing operation, which coincides with the notion of a *trace* in category theory (Chapter 6).

The following is the formal definition of the feedback operation for design problems.

Definition 5.9 (Trace of a design problem)
Given a design problem $\mathbf{d} : \mathbf{P} \otimes \mathbf{R} \nrightarrow \mathbf{Q} \otimes \mathbf{R}$, its *trace*

$$\mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{Q}}(\mathbf{d}) : \mathbf{P} \nrightarrow \mathbf{Q}$$

is defined as follows:

$$\mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{Q}}(\mathbf{d}) : \mathbf{P}^{\mathrm{op}} \times \mathbf{Q} \to_{\mathbf{Pos}} \mathbf{Bool},$$
$$\langle p^*, q\rangle \mapsto \bigvee_{r \in \mathbf{R}} \mathbf{d}(\langle p, r\rangle^*, \langle q, r\rangle).$$

Think of drawing a loop as a way of writing down the following requirement: Something that produces **R** should not use up more of **R** than it produces.

5.5 Example: co-design of an autonomous drone

At this point, we have seen what a design problem is, and we have seen how to interconnect multiple design problems. In this section, we want to show the abilities of this structure in practice, in the context of co-designing an autonomous drone. To do so, we will leverage one of the first non-trivial design problems we introduced: the LQG control one (Section 4.3).

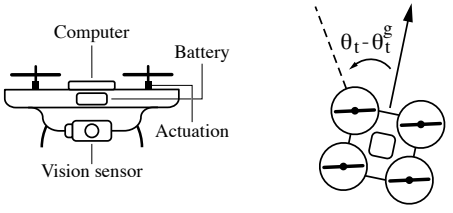


Figure 87: We consider a drone which needs to perform search-and-rescue tasks, and control its alignment with a given goal.

In particular, we want to model an autonomous drone (Fig. 87) capable of performing search-and-rescue missions, where it must efficiently align itself with a predefined goal, such as tracking an object. While we recognize that one key element of this co-design process involves control using LQG techniques, it is important to pinpoint the other critical components and identify their logical connections within the system. In practice, when designing such a system, we need to consider several factors:

- Actuators, which are responsible for lifting and maneuvering the drone;
- A vision sensor, which will give the drone the ability to perceive its environment. It provides essential data to to a feature extraction algorithm, allowing the drone to detect and track targets;

- A controller, bringing the system to the desired state;
- A computing unit, providing the system with the computation needed;
- A battery, powering the entire operation of the robot;
- The actual implementation of all algorithms involved;
- A logic to plan the mission. Parameters such as mission speed, the number of missions to complete, and the allotted mission time all contribute the overall mission strategy.

For each of these components, we must address specific design problems, including formulating them accurately, and determining how to populate them, as detailed in the following.

Actuation The total mass of the system is lifted thanks to *actuation*, which provides lift, control effort and speed by requiring cost, mass and power (Fig. 88).

Obtaining the model: Models can be obtained from catalogues. For instance, power consumption can be modeled as a monotone function of lift and control effort, as in [122].

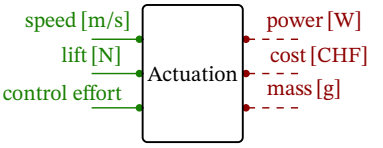


Figure 88: Design problem for the actuation.

Vision sensor The observations required by the drone are provided by a *vision sensor* at a given acquisition frequency and with a specific resolution (typically in px/sterad). Such sensors have a cost, mass and power consumption (Fig. 89).

Obtaining the model: This is obtained from sensor catalogues (e.g., for cameras).

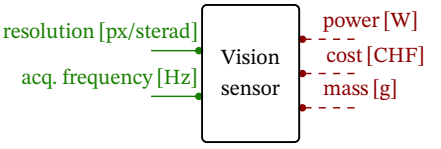


Figure 89: Design problem for the vision sensor.

Feature Extraction Sensor measurements are processed by a *feature extraction* algorithm, providing the LQG control design problem with observations at a certain frequency and accuracy, limited by sensor properties (Fig. 90).

Obtaining the model: This can be obtained by answering photogrammetry questions such as "what resolution is needed to achieve a certain detection accuracy?"

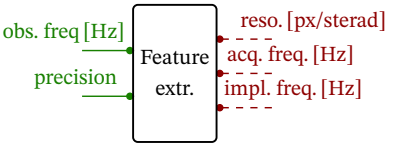


Figure 90: Design problem for feature extraction.

LQG Control As shown in Section 4.3 we can formalize an LQG control problem as a design problem, which provides stability of the system up to a given system noise, requiring observations at a specific frequency and with given precision. The control law has to be implemented at a given frequency, resulting in specific control effort and tracking error (Fig. 91).

Obtaining the model: Given the task of aligning itself with the goal, we define the state of the robot as $\langle \theta_t, \omega_t \rangle$, where θ_t is its heading and ω_t its angular speed. The heading of the goal at time t is denoted by θ_t^{g}. The control is the resulting torque τ_t. The dynamics of the heading are given by the differential equations $\mathrm{d}\theta_t = \omega_t \mathrm{d}t$, $\mathrm{d}\omega_t = \frac{\tau_t}{I}\mathrm{d}t + \mathrm{d}w_t$, where I is the moment of inertia of the drone about its rotational axis, and w_t is a Brownian process with intensity σ_w^2. We assume

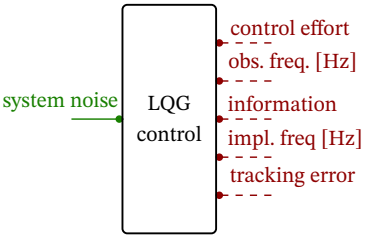


Figure 91: Design problem for LQG control.

Gaussian observations of the relative bearing of drone and goal $y_t = (\theta_t - \theta_t^{\mathrm{g}}) + v_t$, where v_t is white Gaussian noise with intensity σ_v^2, describing measurement uncertainty. Assuming that the goal is far away, θ_t^{g} is approximately constant, and w.l.o.g. we can assume it to be 0 ($y_t = \theta_t + v_t$). We pose the stationary problem of minimizing the objective $\lim_{T\to\infty} \frac{1}{T} \int_0^T \left(\alpha q_0 \theta_t^2 + \frac{r_0}{\alpha} \tau_t^2\right) \mathrm{d}t$. This is a LQG problem with $\mathbf{Q} = \mathrm{diag}(0, \alpha q_0)$, $\mathbf{R} = r_0/\alpha$, for the continuous-time system given by the system matrices:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \frac{1}{I} \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix},$$
$$\mathbf{W} = \begin{bmatrix} 0 & 0 \\ 0 & \sigma_w^2/I \end{bmatrix}, \quad \mathbf{V} = \sigma_v^2.$$

Implementations are given by the different cost weights, parametrized by α. As explained in Section 4.3, the design problem can be obtained by solving specific Riccati equations via numerical simulations.

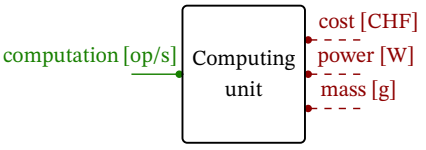


Figure 92: Design problem for the computing unit.

Computing unit The computing unit needs to provide computation required by all the processes we have presented, and requires power and has a mass and a monetary cost (Fig. 92).

Obtaining the model The *computing unit* can be modeled through computer catalogues.

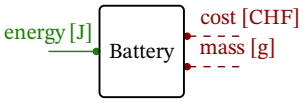


Figure 93: Design problem for the battery.

Battery The *battery* provides energy to the system, requiring cost and mass (Fig. 93).

Obtaining the model: Different battery technologies can be extracted from catalogues. An example is provided in [122].

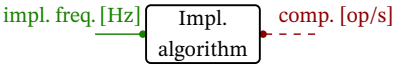


Figure 94: Design problem for the implementation of algorithms.

Algorithms implementation We actually need to implement the control and feature detection algorithms. The related design problems operate at a specific frequency, and require computation (Fig. 94).

Obtaining the model: Models are obtained via catalogues of algorithms, the performance of which can be found via benchmarking. An example is SLAM-Bench [147].

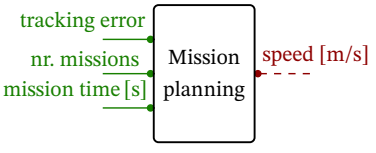


Figure 95: Design problem for mission planning.

Mission planning The *mission planning* design problem evaluates the performance of the system, measured by tracking error, the mission time, the number of missions and the detected drone speed (Fig. 95).

Obtaining the model: This is a simple list of requirements for specific scenarios.

Interconnecting the full diagram

The single components introduced above are interconnected to form a co-design diagram as follows. First, the total power required by the system arises from the sum of the power required by actuation, by the sensors and by the computing unit. Given the mission time, one can determine the energy which needs to be provided by the battery. This is the first feedback loop in the co-design diagram. Second, the computation required by both the control and feature extraction implementations needs to be provided by the computing unit. Third, the mass of the system is given by the masses of the sensors, battery, actuators and computing unit, and determines the lift needed from actuation. This is the second feedback loop in the co-design diagram. The other interconnections arise logically, from the need of the control design problem for observations, produced by the feature extraction design problem, which in turn needs a sensor. Also, the algorithms need to be implemented, requiring computation.

As you can see, here we are using most of the composition operations introduced in this chapter. The interesting thing, is that we are using them "graphically", while meaning it "formally". The next chapter introduces the magic behind this, which ensures that one can work at the graphical level, and things will work out at the formal one. Spoiler alert: it's category theory!

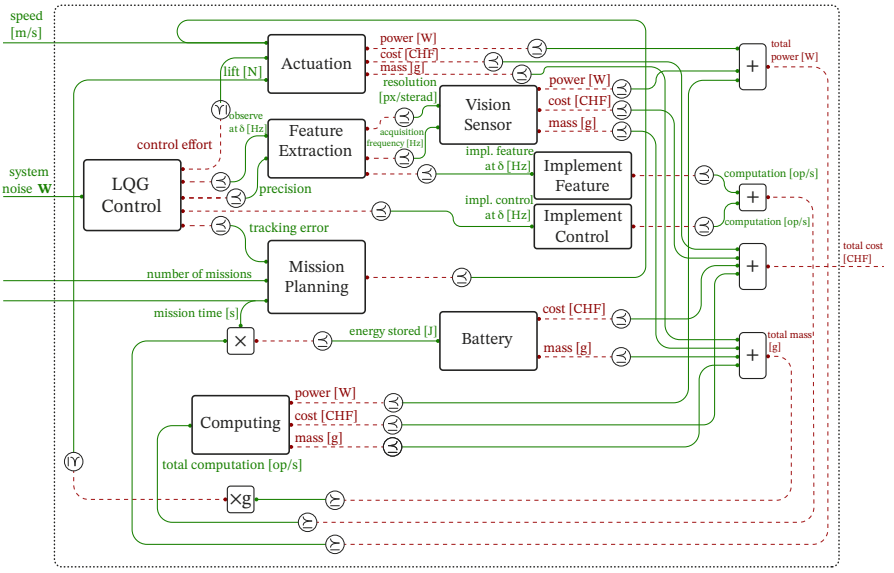


Figure 96: Co-design diagram for the design of an autonomous drone that needs to execute an idealized search-and-rescue mission. The functionalities are task characteristics and the environment. We choose costs as the resources to minimize.

A categorical perspective | 6

Mathematics is the art of giving the same name to different things.
—Henri Poincaré[14]

In this chapter, we make the structure and the interconnections defined in Chapter 4 and Chapter 5 precise mathematically. Category theory allows one to formulate co-design problems elegantly and compactly. In particular, we show that there exists a category of design problems **DP**, which is symmetric monoidal, traced, an locally posetal. This structure will be extremely helpful also when formulating the solution of co-design problems. In this chapter we will provide basic definitions for the various proved properties. For a primer on category theory, as well as the thorough and pedagogical descriptions of the properties proved in this chapter, please refer to our work-in-progress book [123].

Going back to the desiderata, this chapter shows two essential properties of the co-design framework. First, it is at the same time formal and intuitive. On one hand, one can specify and interconnect design problems in various ways at the "graphical" level, and on the other hand such interconnections have a formal meaning, and are not field-specific. Second, it is compositional and hierarchical by nature, able to deal with various forms of composition and abstraction.

6.1 The category of design problems DP

We first briefly recall the definition of category.

Definition 6.1 (Category)
A *category* $\mathbf{C}$ is specified by:

Constituents

1. Objects: A collection $\mathrm{Ob}_{\mathbf{C}}$ whose elements are called *objects*.
2. Morphisms: For every pair of objects X, Y in $\mathrm{Ob}_{\mathbf{C}}$, there is a set called a "hom-set" and indicated as $\mathrm{Hom}_{\mathbf{C}}(X; Y)$, elements of which are called *morphisms* and denoted $f : X \to Y$.

   For such an f, we call X its *source* and Y its *target*.
3. Composition operations: For every three objects X, Y, Z in $\mathrm{Ob}_{\mathbf{C}}$ there is a composition map

$$\fatsemi_{X,Y,Z} : \mathrm{Hom}_{\mathbf{C}}(X; Y) \times \mathrm{Hom}_{\mathbf{C}}(Y; Z) \to \mathrm{Hom}_{\mathbf{C}}(X; Z). \tag{12}$$

[14] Poincaré was a French mathematician, theoretical physicist, engineer, and philosopher of science.

We usually just write $\fatsemi$ instead of $\fatsemi_{X,Y,Z}$:

$$\frac{f : X \to Y \quad g : Y \to Z}{(f \fatsemi g) : X \to Z}. \tag{13}$$

The morphism $f \fatsemi g$ is called the *composition* of f and g.

4. Identity morphisms: For each object X of $\mathbf{C}$, a morphism

$$\mathrm{id}_X : X \to X,$$

called the *identity morphism*.

Conditions

1. Unitality: it holds that

$$\frac{f : W \to X}{f \fatsemi \mathrm{id}_X = f}, \tag{14}$$

and

$$\frac{g : X \to Y}{\mathrm{id}_X \fatsemi g = g}. \tag{15}$$

2. Associativity: it holds that

$$\frac{f : X \to Y \quad g : Y \to Z \quad h : Z \to U}{(f \fatsemi g) \fatsemi h = f \fatsemi (g \fatsemi h)}. \tag{16}$$

Remark 6.2. We denote composition of morphisms using the symbol "$\fatsemi$" (pronounced "then"). This is in contrast to the more common notation for composition, namely $g \circ f$, or simply gf, which reads as "g after f". As usual, f^2 denotes $f \fatsemi f$, f^3 denotes $f \fatsemi f \fatsemi f$, and so on.

Remark 6.3. When we want to emphasize which category we are working with, we will sometimes write

$$f : X \to_{\mathbf{C}} Y \tag{17}$$

to indicate

$$f \in \mathrm{Hom}_{\mathbf{C}}(X; Y). \tag{18}$$

Example 6.4 (Classic categories)**.** Some classic examples of categories are:

- **Set**: the category of sets and functions;
- **Rel**: the category of sets and binary relations;
- **Pos**: the category of posets and monotone functions. We will leverage this one a lot, and will specify a monotone map as a morphism in **Pos**, by writing $f : \mathbf{P} \to_{\mathbf{Pos}} \mathbf{Q}$.

We are now ready to define the category **DP**, which has design problems as morphisms. To do so, we recall the definition of series composition of design problems, and define an identity for it. We define the identity $\mathrm{id}_{\mathbf{P}} : \mathbf{P} \nrightarrow \mathbf{P}$ as follows.

Definition 6.5 (Identity design problem)
For any poset $\mathbf{P}$, the *identity design problem* $\mathrm{id}_{\mathbf{P}} : \mathbf{P} \nrightarrow \mathbf{P}$ is a monotone map

$$\begin{aligned} \mathrm{id}_{\mathbf{P}} : \mathbf{P}^{\mathrm{op}} \times \mathbf{P} &\to_{\mathbf{Pos}} \mathbf{Bool}, \\ \langle p_1^*, p_2 \rangle &\mapsto p_1 \preceq_{\mathbf{P}} p_2. \end{aligned} \tag{19}$$

Figure 97: Identity design problem.

Remark 6.6 (Monotonicity of the identity)**.** Let's consider $p_1' \preceq_{\mathbf{P}} p_1$. If it holds $p_1 \preceq_{\mathbf{P}} p_2$, then it also holds $p_1' \preceq_{\mathbf{P}} p_2$. Similarly, now consider $p_2 \preceq_{\mathbf{P}} p_2'$. If it holds $p_1 \preceq_{\mathbf{P}} p_2$, then it also holds $p_1 \preceq_{\mathbf{P}} p_2'$.

In diagrammatic notation, we represent $\mathrm{id}_{\mathbf{P}}$ as in Fig. 97.

Lemma 6.7. The series composition operation as in (11) satisfies the left and right unit laws (Fig. 98).

See proof on page 210.

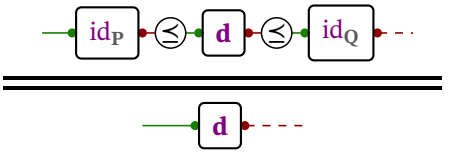


Figure 98: Left and right unitality for **DP**.

We are now ready to define the category of design problems **DP**.

Definition 6.8 (Category of design problems **DP**)
The category of design problems **DP** consists of the following constituents:

1. *Objects*: The objects of **DP** are posets.
2. *Morphisms*: The morphisms of **DP** are design problems (Def. 4.1).
3. *Identity morphism*: The identity morphism $\mathrm{id}_{\mathbf{P}} : \mathbf{P} \nrightarrow \mathbf{P}$ is given by Def. 6.5.
4. *Composition operation*: Given morphisms $\mathbf{d} : \mathbf{P} \nrightarrow \mathbf{Q}$ and $\mathbf{e} : \mathbf{Q} \nrightarrow \mathbf{R}$, their composition $\mathbf{d} \fatsemi \mathbf{e} : \mathbf{P} \nrightarrow \mathbf{R}$ is given by Def. 5.1.

We have already shown that the composition operator "$\fatsemi$" is associative and unital, and that the composition of two design problems is a design problem (closure). Therefore, **DP** is a category.

DP is called **Feas** in [148].

Remark 6.9 (Relation between **DPI** and **DP**)**.** One can come up with a definition of a category of design problems with implementation **DPI**. We have already seen in Remark 4.2 that we can obtain a DP from a DPI. One can make this more formal and say that there exists a forgetful semifunctor from **DPI** to **DP**. Similarly, in the other direction, we can take a DP and find a corresponding DPI, via another semifunctor. For brevity, we refer the reader to [123]

6.2 DP is a symmetric monoidal category

DP is a monoidal category

Let's recall the notion of monoidal category. To do so, we first need to define the notion of functor.

Definition 6.10 (Functor)
Given categories $\mathbf{C}$ and $\mathbf{D}$, a *functor* $F : \mathbf{C} \to \mathbf{D}$ from $\mathbf{C}$ to $\mathbf{D}$ is defined by:

Constituents

1. A map
$$F_\bullet : \mathrm{Ob}_{\mathbf{C}} \to \mathrm{Ob}_{\mathbf{D}}. \tag{20}$$
2. For every pair of objects $X, Y \in \mathrm{Ob}_{\mathbf{C}}$ a map
$$F_\to : \mathrm{Hom}_{\mathbf{C}}(X;Y) \to \mathrm{Hom}_{\mathbf{D}}(F_\bullet(X);F_\bullet(Y)). \tag{21}$$

Conditions

1. Functor application to morphisms is compatible with the respective category composition operations:
$$\frac{f : X \to_{\mathbf{C}} Y \qquad g : Y \to_{\mathbf{C}} Z}{F_\to(f \fatsemi_{\mathbf{C}} g) = F_\to(f) \fatsemi_{\mathbf{D}} F_\to(g)}. \tag{22}$$
2. Functor application is compatible with identities:
$$F_\to(\mathrm{id}_X) = \mathrm{id}_{F_\bullet(X)} \tag{23}$$
for all objects X in $\mathbf{C}$.

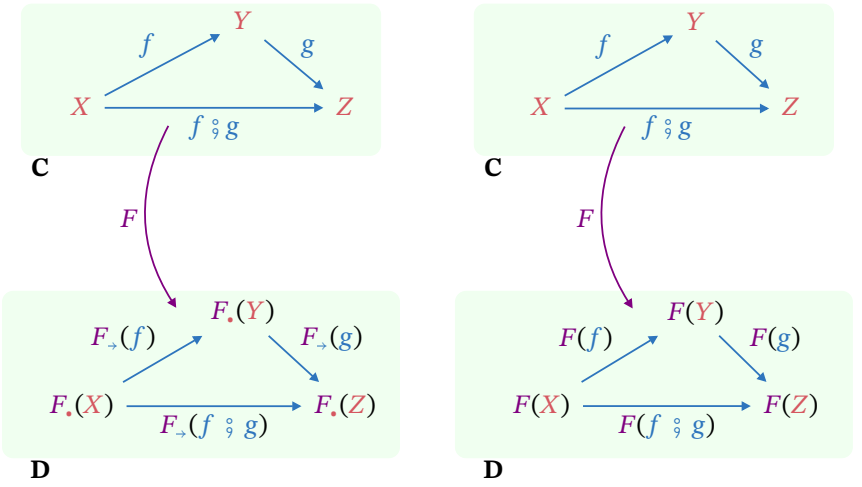
This situation is depicted graphically in Fig. 99a. It is common to overload the notation and use F to mean both $F_\bullet$ and $F_\to$. The diagram with this overloaded "synthetic notation" is in Fig. 99b.

We are now ready to define monoidal categories.

Definition 6.11 (Monoidal category)
A *monoidal structure* on a category $\mathbf{C}$ is specified by:

Constituents

1. A functor $\otimes : \mathbf{C} \times \mathbf{C} \to \mathbf{C}$, called the *monoidal product*.
2. An object $\mathbf{1} \in \mathrm{Ob}_{\mathbf{C}}$, called the *monoidal unit*.
3. A natural isomorphism, called the *associator*, whose components are of



(a) Functor diagram **(b)** Synthetic notation

Figure 99: Commuting diagrams for semifunctors, with verbose notation (left) and synthetic notation (right).

the type

$$\mathsf{as}_{X,Y,Z} : (X \otimes Y) \otimes Z \xrightarrow{\cong} X \otimes (Y \otimes Z) \qquad X, Y, Z \in \mathrm{Ob}_{\mathbf{C}}. \tag{24}$$

4. A natural isomorphism, called the *left unitor*, whose components are of the type

$$\mathsf{lu}_X : \mathbf{1} \otimes X \xrightarrow{\cong} X \qquad X \in \mathrm{Ob}_{\mathbf{C}}. \tag{25}$$

5. A natural isomorphism, called the *right unitor*, whose components are of the type

$$\mathsf{ru}_X : X \otimes \mathbf{1} \xrightarrow{\cong} X \qquad X \in \mathrm{Ob}_{\mathbf{C}}. \tag{26}$$

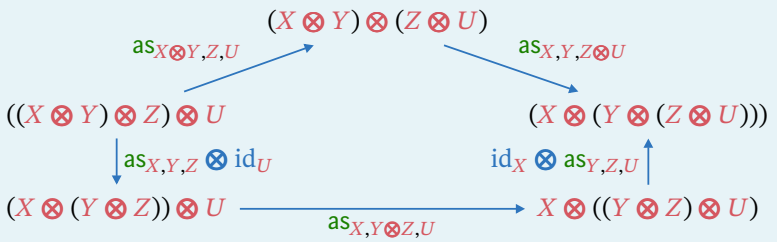
Conditions

For all $X, Y, Z, U \in \mathrm{Ob}_{\mathbf{C}}$, the following diagrams must commute:

1. Triangle identities.

$$\begin{array}{ccc} (X \otimes \mathbf{1}) \otimes Y & \xrightarrow{\mathsf{as}_{X,\mathbf{1},Y}} & X \otimes (\mathbf{1} \otimes Y) \\ & {\scriptstyle \mathsf{ru}_X \otimes \mathrm{id}_Y} \searrow \quad \swarrow {\scriptstyle \mathrm{id}_X \otimes \mathsf{lu}_Y} & \\ & X \otimes Y & \end{array} \tag{27}$$

2. Pentagon identities.



(28)

A category equipped with a monoidal structure is called a *monoidal category*. If the components of the associator, left unitor, and right unitor are all equalities, one calls the category *strict* monoidal.

Remark 6.12. Note that in the constituents listed in Def. 6.11 we specified natural isomorphisms as, lu, and ru simply in terms of their components. You may be wondering: which functors are the respective source and target of these natural transformations? Since it is a mouthful to write, this information is often left to be inferred from the components given. Let us quickly illustrate how to see, from the components, which functors are involved. Take, for example, the left unitor. Its components are

$$\mathsf{lu}_X : \mathbf{1} \otimes X \xrightarrow{\cong} X \qquad X \in \mathrm{Ob}_{\mathbf{C}}, \tag{29}$$

so, if F and G denote the functors which are the source and target of lu, the functor F must act on objects by $F(X) = \mathbf{1} \otimes X$ and G must act by $G(X) = X$. The "obvious" or "canonical" choice then (given that we are considering *any* monoidal category) is that G is the identity functor and that F is the functor which acts on morphisms by mapping $f : X \to Y$ to

$$\mathrm{id}_{\mathbf{1}} \otimes f : \mathbf{1} \otimes X \to \mathbf{1} \otimes Y. \tag{30}$$

Note that the components of the left unitor lu are indexed by one variable $X \in \mathrm{Ob}_{\mathbf{C}}$, while the associator as is indexed by *three* variables! The associator is therefore a natural transformation between two functors of the type

$$\mathbf{C} \times \mathbf{C} \times \mathbf{C} \to \mathbf{C}. \tag{31}$$

Armed with this knowledge, we can now define the monoidal structure on **DP**. In Def. 5.7 we have defined the parallel composition operation for design problems, which we refer to as *monoidal product*, in category theory terms.

Lemma 6.13. The monoidal product $\otimes$ is functorial in **DP**.

See proof on page 211.

Now, we can identify the constituents of the monoidal structure on **DP**, with

monoidal product $\otimes$ and monoidal unit $\mathbf{1}$.

We identify the left and right unitors $\mathsf{lu}_{\mathbf{P}}, \mathsf{ru}_{\mathbf{P}}$, for any $\mathbf{P} \in \mathrm{Ob}_{\mathbf{DP}}$, given by:

$$\begin{aligned} \mathsf{lu}_{\mathbf{P}} : (\mathbf{1} \times \mathbf{P})^{\mathrm{op}} \times \mathbf{P} &\to_{\mathbf{Pos}} \mathbf{Bool}, \\ \langle\langle \bullet, p_1 \rangle^*, p_2 \rangle &\mapsto p_1 \preceq_{\mathbf{P}} p_2, \end{aligned}$$

$$\begin{aligned} \mathsf{ru}_{\mathbf{P}} : (\mathbf{P} \times \mathbf{1})^{\mathrm{op}} \times \mathbf{P} &\to_{\mathbf{Pos}} \mathbf{Bool}, \\ \langle\langle p_1, \bullet \rangle^*, p_2 \rangle &\mapsto p_1 \preceq_{\mathbf{P}} p_2. \end{aligned}$$

Clearly, both the left and right unitors are valid design problems (monotonicity), and hence

$$\mathsf{lu}_{\mathbf{P}} \in \mathrm{Hom}_{\mathbf{DP}}(\mathbf{1} \times \mathbf{P}; \mathbf{P}), \quad \mathsf{ru}_{\mathbf{P}} \in \mathrm{Hom}_{\mathbf{DP}}(\mathbf{P} \times \mathbf{1}; \mathbf{P}).$$

We prove that both these constructions are valid isomorphisms.

Lemma 6.14. The left and right unitors for **DP** are valid isomorphisms.

See proof on page 211.

We now identify the associator $\mathsf{as}_{\mathbf{P},\mathbf{Q},\mathbf{R}}$, for any $\mathbf{P}, \mathbf{Q}, \mathbf{R} \in \mathrm{Ob}_{\mathbf{DP}}$, given by:

$$\begin{aligned} \mathsf{as}_{\mathbf{P},\mathbf{Q},\mathbf{R}} : ((\mathbf{P} \times \mathbf{Q}) \times \mathbf{R})^{\mathrm{op}} \times (\mathbf{P} \times (\mathbf{Q} \times \mathbf{R})) &\to_{\mathbf{Pos}} \mathbf{Bool}, \\ \langle\langle p_1, q_1 \rangle, r_1 \rangle^*, \langle p_2, \langle q_2, r_2 \rangle\rangle &\mapsto (p_1 \preceq p_2) \wedge (q_1 \preceq q_2) \wedge (r_1 \preceq r_2). \end{aligned}$$

Clearly, this is a valid design problem:

$$\mathsf{as}_{\mathbf{P},\mathbf{Q},\mathbf{R}} \in \mathrm{Hom}_{\mathbf{DP}}((\mathbf{P} \times \mathbf{Q}) \times \mathbf{R}; \mathbf{P} \times (\mathbf{Q} \times \mathbf{R})), \quad \mathbf{P}, \mathbf{Q}, \mathbf{R} \in \mathrm{Ob}_{\mathbf{DP}}.$$

Lemma 6.15. The associator for **DP** is a valid isomorphism.

See proof on page 212.

Lemma 6.16. $\langle \mathbf{DP}, \otimes, \mathbf{1}, \mathsf{lu}, \mathsf{ru}, \mathsf{as} \rangle$ is a monoidal category.

See proof on page 213.

DP is a symmetric monoidal category One can define further structure on a category, to make it *symmetric monoidal*.

Definition 6.17 (Braided monoidal category)
A *braided monoidal category* is a monoidal category $\langle \mathbf{C}, \otimes, \mathbf{1}, \mathsf{as}, \mathsf{lu}, \mathsf{ru} \rangle$ equipped with a *braiding*, which is specified by

Constituents

1. A natural isomorphism br, called the *braiding*, whose components are of the type
$$\mathsf{br}_{X,Y} : (X \otimes Y) \xrightarrow{\cong} (Y \otimes X), \quad X, Y \in \mathrm{Ob}_{\mathbf{C}}. \tag{32}$$
Explicitly, this means that for any $f_1 : X_1 \to Y_1$ and $f_2 : X_2 \to Y_2$, the following diagram commutes:
$$\begin{array}{ccc} X_1 \otimes X_2 & \xrightarrow{f_1 \otimes f_2} & Y_1 \otimes Y_2 \\ \downarrow{\scriptstyle \mathsf{br}_{X_1,X_2}} & & \downarrow{\scriptstyle \mathsf{br}_{Y_1,Y_2}} \\ X_2 \otimes X_1 & \xrightarrow[f_2 \otimes f_1]{} & Y_2 \otimes Y_1 \end{array} \tag{33}$$

Conditions

1. *Hexagon identities:* Given any objects $X, Y, Z \in \mathrm{Ob}_{\mathbf{C}}$, the following diagrams must commute.
$$\begin{array}{ccccc} (X \otimes Y) \otimes Z & \xrightarrow{\mathsf{br}_{X,Y} \otimes \mathrm{id}_Z} & (Y \otimes X) \otimes Z & \xrightarrow{\mathsf{as}_{Y,X,Z}} & Y \otimes (X \otimes Z) \\ \downarrow{\scriptstyle \mathsf{as}_{X,Y,Z}} & & & & \downarrow{\scriptstyle \mathrm{id}_Y \otimes \mathsf{br}_{X,Z}} \\ X \otimes (Y \otimes Z) & & \xrightarrow[\mathsf{br}_{X,Y\otimes Z}]{} (Y \otimes Z) \otimes X & \xrightarrow{\mathsf{as}_{Y,Z,X}} & Y \otimes (Z \otimes X) \end{array} \tag{34}$$
$$\begin{array}{ccccc} X \otimes (Y \otimes Z) & \xrightarrow{\mathrm{id}_X \otimes \mathsf{br}_{Y,Z}} & X \otimes (Z \otimes Y) & \xrightarrow{\mathsf{as}^{-1}_{Y,X,Z}} & (X \otimes Z) \otimes Y \\ \downarrow{\scriptstyle \mathsf{as}^{-1}_{X,Y,Z}} & & & & \downarrow{\scriptstyle \mathsf{br}_{X,Z} \otimes \mathrm{id}_Y} \\ (X \otimes Y) \otimes Z & & \xrightarrow[\mathsf{br}_{X\otimes Y,Z}]{} Z \otimes (X \otimes Y) & \xrightarrow{\mathsf{as}^{-1}_{Z,X,Y}} & (Z \otimes X) \otimes Y \end{array} \tag{35}$$

Remark 6.18. If $\langle \mathbf{C}, \otimes, \mathbf{1}, \mathsf{as}, \mathsf{lu}, \mathsf{ru}, \mathsf{br} \rangle$ is a braided monoidal category, we can show that the following diagram commutes for all $X \in \mathrm{Ob}_{\mathbf{C}}$.
$$\begin{array}{ccc} \mathbf{1} \otimes X & \xrightarrow{\mathsf{br}_{\mathbf{1},X}} & X \otimes \mathbf{1} \\ & {\scriptstyle \mathsf{lu}_X} \searrow \quad \swarrow {\scriptstyle \mathsf{ru}_X} & \\ & X & \end{array} \tag{36}$$

Definition 6.19 (Symmetric monoidal category)
A *symmetric monoidal category* is a braided monoidal category $\langle \mathbf{C}, \otimes, \mathbf{1}, \mathsf{as},$

$\mathsf{lu}, \mathsf{ru}, \mathsf{br}\rangle$ for which the braiding satisfies the symmetry condition

$$\mathsf{br}_{X,Y} \fatsemi \mathsf{br}_{Y,X} = \mathrm{id}_{X \otimes Y} \tag{37}$$

for all $X, Y \in \mathrm{Ob}_{\mathbf{C}}$.

Remark 6.20. If br is a natural isomorphism such that it is a candidate to be a braiding on a given monoidal category, and if, additionally, it satisfies (37), then the two hexagon identities are equivalent, and so only one of them needs to be checked.

We now identify the structure for the defined monoidal category to be symmetric. The braiding operator $\mathsf{br}_{\mathbf{P},\mathbf{Q}}$, for any $\mathbf{P}, \mathbf{Q} \in \mathrm{Ob}_{\mathbf{DP}}$ is given by:

$$\begin{aligned} \mathsf{br}_{\mathbf{P},\mathbf{Q}} : (\mathbf{P} \times \mathbf{Q})^{\mathrm{op}} \times (\mathbf{Q} \times \mathbf{P}) &\to_{\mathbf{Pos}} \mathbf{Bool}, \\ \langle p_1, q_2 \rangle^*, \langle q_2, p_2 \rangle &\mapsto p_1 \preceq p_2 \wedge q_1 \preceq q_2. \end{aligned}$$

Clearly, this is a valid design problem: $\mathsf{br}_{\mathbf{P},\mathbf{Q}} \in \mathrm{Hom}_{\mathbf{DP}}(\mathbf{P} \times \mathbf{Q}; \mathbf{Q} \times \mathbf{P})$.

Lemma 6.21. The braiding for **DP** is a valid natural isomorphism.

See proof on page 215.

We are now ready to state that **DP** can be equipped with structure to form a symmetric monoidal category.

Lemma 6.22. $\langle \mathbf{DP}, \otimes, \mathbf{1}, \mathsf{lu}, \mathsf{ru}, \mathsf{as}, \mathsf{br} \rangle$ is a symmetric monoidal category.

See proof on page 215.

6.3 DP is a traced monoidal category

Finally, we introduce more structure and define the notion of *traced monoidal category*.

Definition 6.23 (Traced monoidal category)
We say that a symmetric monoidal category $\langle \mathbf{C}, \otimes, \mathbf{1}, \mathsf{as}, \mathsf{lu}, \mathsf{ru}, \mathsf{br} \rangle$ is *traced* if it is equipped with a family of functions

$$\mathrm{Tr}^{Z}_{X,Y} : \mathrm{Hom}_{\mathbf{C}}(X \otimes Z; Y \otimes Z) \to \mathrm{Hom}_{\mathbf{C}}(X; Y), \tag{38}$$

satisfying the following axioms:

1. *Naturality in* X: For any morphisms $f : X \otimes Z \to Y \otimes Z$ and $g : X' \to X$,

$$\mathrm{Tr}^{Z}_{X',Y}((g \otimes \mathrm{id}_Z) \fatsemi f) = g \fatsemi \mathrm{Tr}^{Z}_{X,Y}(f) \tag{39}$$

2. *Naturality in Y:* For any morphisms $f : X \otimes Z \to Y \otimes Z$ and $g : Y \to Y'$,

$$\mathrm{Tr}^{Z}_{X,Y'}(f \fatsemi (g \otimes \mathrm{id}_Z)) = \mathrm{Tr}^{Z}_{X,Y}(f) \fatsemi g \tag{40}$$

3. *Dinaturality in Z:* For any morphisms $f : X \otimes Z \to Y \otimes Z'$ and $g : Z' \to Z$,

$$\mathrm{Tr}^{Z}_{X,Y}(f \fatsemi (\mathrm{id}_Y \otimes g)) = \mathrm{Tr}^{Z'}_{X,Y}((\mathrm{id}_X \otimes g) \fatsemi f). \tag{41}$$

4. *Vanishing I:* For any morphisms $f : X \otimes \mathbf{1} \to Y \otimes \mathbf{1}$ in $\mathbf{C}$,

$$\mathrm{Tr}^{\mathbf{1}}_{X,Y}(f) = \mathsf{ru}^{-1}_X \fatsemi f \fatsemi \mathsf{ru}_Y. \tag{42}$$

5. *Vanishing II:* For any morphism $f : (X \otimes Z) \otimes U \to (Y \otimes Z) \otimes U$ in $\mathbf{C}$,

$$\mathrm{Tr}^{Z}_{X,Y}\left(\mathrm{Tr}^{U}_{X\otimes Z, Y\otimes Z}(f)\right) = \mathrm{Tr}^{Z\otimes U}_{X,Y}(\mathsf{as}_{X,Z,U} \fatsemi f \fatsemi \mathsf{as}^{-1}_{Y,Z,U}). \tag{43}$$

6. *Superposing:* For any morphism $f : X \otimes Z \to Y \otimes Z$ in $\mathbf{C}$,

$$\mathrm{Tr}^{Z}_{V\otimes X, V\otimes Y}(\mathsf{as}_{V,X,Z} \fatsemi \mathrm{id}_V \otimes f \fatsemi \mathsf{as}^{-1}_{V,Y,Z}) = \mathrm{id}_V \otimes \mathrm{Tr}^{Z}_{X,Y}(f). \tag{44}$$

7. *Yanking:*

$$\mathrm{Tr}^{Z}_{Z,Z}(\mathsf{br}_{Z,Z}) = \mathrm{id}_Z. \tag{45}$$

DP is actually a traced monoidal category, with the trace being the "feedback" operator defined in Section 5.4.

Lemma 6.24. $\langle \mathbf{DP}, \otimes, \mathbf{1}, \mathsf{lu}, \mathsf{ru}, \mathsf{as}, \mathsf{br} \rangle$ is a traced monoidal category.

See proof on page 217.

6.4 More structure

First, DPs can be ordered, and their hom-sets form a bounded lattice.

Ordering DPs

Definition 6.25 (Order on **DP**)
Suppose that **P** and **Q** are posets, and that $\mathbf{d}, \mathbf{e} : \mathbf{P} \nrightarrow \mathbf{Q}$ are design problems. We define the order as follows:

$$\frac{\mathbf{d} \preceq_{\mathbf{DP}} \mathbf{e}}{\mathbf{d}(p^*, q) \preceq_{\mathbf{Bool}} \mathbf{e}(p^*, q) \text{ for all } p \in \mathbf{P}, q \in \mathbf{Q}.},$$

Figure 100: The design problem **d** implies the design problem **e**.

Remark 6.26. Recall that design problems are monotone functions, and note that the order defined in Def. 6.25 is just the usual order on monotone functions.

We diagrammatically represent the relation $\mathbf{d} \preceq_{\mathbf{DP}} \mathbf{e}$ as in Fig. 100.

Lattice structure of sets of DP

Given the definitions of $\wedge$ and $\vee$ in the previous sections, we can prove that hom-sets of **DP** have a lattice structure.

This lattice is bounded by a "true" and a "false" DP.

Definition 6.27 (False and true DPs)
Given any two partial orders $\mathbf{P}, \mathbf{Q}$, we can define a *false* DP as

$$\begin{aligned} \bot_{\mathbf{P},\mathbf{Q}} : \mathbf{P}^{\mathrm{op}} \times \mathbf{Q} &\to_{\mathbf{Pos}} \mathbf{Bool}, \\ \langle p^*, q \rangle &\mapsto \bot. \end{aligned}$$

We can define a *true* DP as

$$\begin{aligned} \top_{\mathbf{P},\mathbf{Q}} : \mathbf{P}^{\mathrm{op}} \times \mathbf{Q} &\to_{\mathbf{Pos}} \mathbf{Bool}, \\ \langle p^*, q \rangle &\mapsto \top. \end{aligned}$$

For any functionality-resource pair $\mathbf{P}, \mathbf{Q}$, these represent the design problem which is never (respectively always) feasible.

Lemma 6.28. $\mathrm{Hom}_{\mathbf{DP}}(\mathbf{P}; \mathbf{Q})$ is a bounded lattice with union $\vee$ as join, intersection $\wedge$ as meet, top $\top_{\mathbf{P},\mathbf{Q}}$ and bottom $\bot_{\mathbf{P},\mathbf{Q}}$.

See proof on page 221.

We show that a **DP** hom-set is a *complete lattice*.

Definition 6.29 (Complete Lattice)
A poset $\mathbf{P} = \langle P, \leq_{\mathbf{P}} \rangle$ is a *complete lattice* if every subset S of P has both a *greatest lower bound* (often referred to as the *infimum, meet*) and a *least upper bound* (often referred to as the *supremum, join*).

Example 6.30. Consider the power set of any given set, ordered by inclusion. The supremum of any two subsets is given by their union. The infimum of any two subsets is given by their intersection.

Lemma 6.31 (**DP** hom-sets are complete lattices)**.** Hom-sets of **DP** are complete lattices.

See proof on page 221.

Definition 6.32 (Distributive Lattice)
A lattice $\mathbf{P} = \langle \mathbf{P}, \wedge, \vee \rangle$ is a *distributive lattice* if for all $x, y, z \in \mathbf{P}$:

$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z). \tag{46}$$

Remark 6.33. Note that condition (46) is equivalent to its dual:

$$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z),$$

for all $x, y, z \in \mathbf{P}$.

Lemma 6.34. Consider $\mathbf{d}, \mathbf{e}, \mathbf{g} \in \mathrm{Hom}_{\mathbf{DP}}(\mathbf{P}; \mathbf{Q})$. We have

$$(\mathbf{d} \wedge \mathbf{e}) \vee \mathbf{g} = (\mathbf{d} \vee \mathbf{g}) \wedge (\mathbf{e} \vee \mathbf{g}).$$

See proof on page 222.

Lemma 6.35. Consider $\mathbf{d}, \mathbf{e}, \mathbf{g} \in \mathrm{Hom}_{\mathbf{DP}}(\mathbf{P}; \mathbf{Q})$. We have

$$(\mathbf{d} \vee \mathbf{e}) \wedge \mathbf{g} = (\mathbf{d} \wedge \mathbf{g}) \vee (\mathbf{e} \wedge \mathbf{g}).$$

See proof on page 222.

Lemma 6.36 (**DP** hom-sets are distributive lattices)**.** Hom-sets of **DP** are distributive lattices.

Proof. Either Lemma 6.34 or Lemma 6.35 prove the statement. □

Interaction with composition

Furthermore, we show that all composition operations preserve joins, and all composition operations except trace preserve meets.

Lemma 6.37. Consider $\mathbf{d}, \mathbf{e} \in \mathrm{Hom}_{\mathbf{DP}}(\mathbf{P}; \mathbf{Q})$ and $\mathbf{g} \in \mathrm{Hom}_{\mathbf{DP}}(\mathbf{Q}; \mathbf{R})$. We have

$$(\mathbf{d} \vee \mathbf{e}) \fatsemi \mathbf{g} = (\mathbf{d} \fatsemi \mathbf{g}) \vee (\mathbf{e} \fatsemi \mathbf{g}).$$

This is diagrammatically represented in Fig. 101.

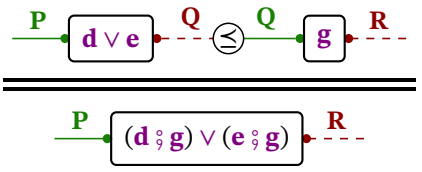


Figure 101: Diagrammatic statement.

See proof on page 223.

Remark 6.38. Consider $\mathbf{d}, \mathbf{e} \in \mathrm{Hom}_{\mathbf{DP}}(\mathbf{P}; \mathbf{Q})$ and $\mathbf{g}, \mathbf{h} \in \mathrm{Hom}_{\mathbf{DP}}(\mathbf{Q}; \mathbf{R})$. In general, we have:

$$(\mathbf{d} \vee \mathbf{e}) \fatsemi (\mathbf{g} \vee \mathbf{h}) \neq (\mathbf{d} \fatsemi \mathbf{g}) \vee (\mathbf{e} \fatsemi \mathbf{h}).$$

Indeed, consider $\mathbf{d} = \top_{\mathbf{P},\mathbf{Q}}$, $\mathbf{e} = \bot_{\mathbf{P},\mathbf{Q}}$, $\mathbf{g} = \bot_{\mathbf{Q},\mathbf{R}}$, and $\mathbf{h} = \top_{\mathbf{Q},\mathbf{R}}$. Clearly:

$$\begin{aligned}((\mathbf{d} \vee \mathbf{e}) \fatsemi (\mathbf{g} \vee \mathbf{h}))(p^*, r) &= \bigvee_{q \in \mathbf{Q}} (\mathbf{d} \vee \mathbf{e})(p^*, r) \wedge (\mathbf{g} \vee \mathbf{h})(q^*, r) \\ &= \top,\end{aligned}$$

but

$$\begin{aligned}&((\mathbf{d} \fatsemi \mathbf{g}) \vee (\mathbf{e} \fatsemi \mathbf{h}))(p^*, r) \\ =&\left(\bigvee_{q \in \mathbf{Q}} \mathbf{d}(p^*, q) \wedge \mathbf{g}(q^*, r)\right) \vee \left(\bigvee_{q \in \mathbf{Q}} \mathbf{e}(p^*, q) \wedge \mathbf{h}(q^*, r)\right) \\ =&\bot \vee \bot \\ =&\bot.\end{aligned}$$

Lemma 6.39. Consider $\mathbf{d}, \mathbf{e} \in \mathrm{Hom}_{\mathbf{DP}}(\mathbf{P}; \mathbf{Q})$ and $\mathbf{g} \in \mathrm{Hom}_{\mathbf{DP}}(\mathbf{Q}; \mathbf{R})$. We have

$$(\mathbf{d} \wedge \mathbf{e}) \fatsemi \mathbf{g} = (\mathbf{d} \fatsemi \mathbf{g}) \wedge (\mathbf{e} \fatsemi \mathbf{g}).$$

This is diagrammatically represented in Fig. 102.

See proof on page 223.

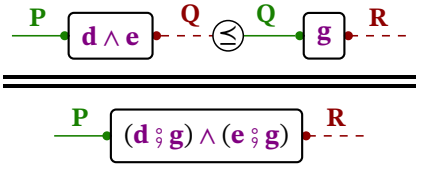


Figure 102: Diagrammatic statement.

DP is locally posetal

Let's recall the definition of enriched category.

Definition 6.40 (**V**-enriched category)
Let $\langle \mathbf{V}, \otimes, \mathbf{1}, \mathsf{as}, \mathsf{lu}, \mathsf{ru} \rangle$ be a monoidal category. A **V**-enriched category **E** is:

Constituents

1. A collection of objects $\mathrm{Ob}_{\mathbf{E}}$.
2. For every pair of objects $\langle X, Y \rangle$ of **E**, a *hom-object*

$$\mathbf{E}(X, Y) \in \mathrm{Ob}_{\mathbf{V}} \tag{47}$$

3. For every triple of objects $\langle X, Y, Z \rangle$ of **E**, a morphism of **V**

$$\mathsf{co}_{X,Y,Z} : \mathbf{E}(X, Y) \otimes \mathbf{E}(Y, Z) \to_{\mathbf{V}} \mathbf{E}(X, Z), \tag{48}$$

called *composition morphism*.

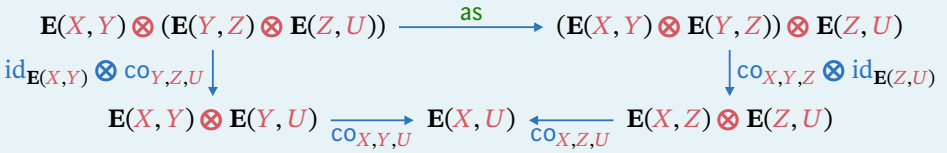
4. For each object X of **E**, a morphism of **V**

$$\mathsf{ic}_X : \mathbf{1} \to_{\mathbf{V}} \mathbf{E}(X, X), \tag{49}$$

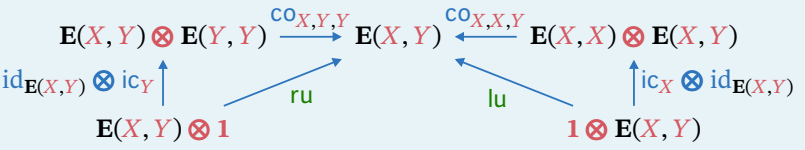
called *identity-choosing morphism*.

Conditions

1. Associativity: for any $X, Y, Z, U \in \mathrm{Ob}_{\mathbf{C}}$, the diagram in Fig. 103a commutes.
2. Neutrality: for any $X, Y, Z \in \mathrm{Ob}_{\mathbf{C}}$ the diagram in Fig. 103b commutes.



(a) Associativity



(b) Neutrality

Figure 103: Coherence diagrams for enriched categories

In this section, we will show that **DP** is locally posetal (i.e., enriched in the category of posets **Pos**).

First, we prove a preliminary result.

Lemma 6.41. The map

$$\begin{aligned} \mathrm{co}_{\mathbf{P},\mathbf{Q},\mathbf{R}} : \mathrm{Hom}_{\mathbf{DP}}(\mathbf{P};\mathbf{Q}) \times \mathrm{Hom}_{\mathbf{DP}}(\mathbf{Q};\mathbf{R}) &\to \mathrm{Hom}_{\mathbf{DP}}(\mathbf{P};\mathbf{R}), \\ \langle \mathbf{d}, \mathbf{e} \rangle &\mapsto \mathbf{d} \fatsemi \mathbf{e}. \end{aligned}$$

is monotone.

See proof on page 223.

Lemma 6.42. **DP** is enriched in **Pos**.

See proof on page 224.

Solving co-design problems 7

If I had an hour to solve a problem I'd spend 55 minutes thinking about the problem and 5 minutes thinking about solutions.
—Albert Einstein[15]

In this chapter, we present the methodology to compute the solution of co-design problems. Starting from the solution concept (Section 7.1), we will present a categorical interpretation, in which there is a category of problems, a category of solutions (Section 7.2), and a functor between them (Section 7.3). We will then present in more practical terms how to solve co-design problems (Section 7.4 to Section 7.6), reporting early results from [122], and finally present two examples (Section 7.7 and Section 7.8).

7.1 Solution concept

In this and the following sections we are going to build towards the solution of co-design problems.

Formulation as an optimization problem

We will consider an arbitrary multigraph of design problems, in which nodes are design problems and edges are arbitrary interconnections between functionality and resources, obtained through the operations of a traced monoidal category (series, parallel, feedback) plus the lattice structure (and, or) of design problems. On this structure we want to solve the query FixFunMinRes or, symmetrically, FixResMaxFun (Section 3.4). For the sake of narrative, we now consider FixFunMinRes, and organize the problem statement as in classic optimization.

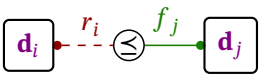
Data An arbitrary multigraph of design problems $\mathbf{d}_k : \mathbf{F}_k \nrightarrow \mathbf{R}_k$. *Variables* are functionalities $f_k \in \mathbf{F}_k$ and resources $r_k \in \mathbf{R}_k$. Functionalities are specified by the user (for FixResMaxFun, the user specifies the resources). Let's denote the product of functionalities and resources by $\mathbf{F}$ and $\mathbf{R}$, respectively.

Constraints We have two types of constraints.

[15] Einstein was a German-born theoretical physicist, who received the Nobel Prize in Physics in 1921. Notably, he was both a student and a faculty at ETH Zürich.

▹ For each *node*, we need *component feasibility*, i.e., $\mathbf{d}_k(f_k^*, r_k) = \mathsf{T}$:

▹ For each *edge* between $\mathbf{d}_i$ and $\mathbf{e}_j$, we need *co-design constraints*, i.e., $r_i \preceq f_j$:



Objective The objective is to minimize resources:

$$\operatorname*{Min}_{\preceq_{\mathbf{R}}} r$$

Remark 7.1 (Expressivity of design problems)**.** We are not assuming properties like convexity, or even weaker properties such as differentiability or continuity of the constraints. In fact, we are not even assuming that functionality and resources are continuous spaces; they could be arbitrary discrete posets.

Sketch of the solution procedure

We look at the solution procedure from a compositional point of view. We will assume that we know the solution to FixFunMinRes for each of the components. We think of the components as primitive blocks, because they are given in a catalogue format as a DPI, or they are special cases ($+$, $\otimes$, *etc.*) which we will solve as special cases. Given the solution for the primitive blocks, we want to know what is the solution for FixFunMinRes for the entire diagram.

What is the form of the solution that we expect? Given a DP $\mathbf{d} : \mathbf{F} \nrightarrow \mathbf{R}$ we expect the solution to FixFunMinRes to be a function that, given a fixed functionality $f \in \mathbf{F}$, returns the minimal resources, which form an upper set. We call this function $H_{\mathbf{d}}$.

Definition 7.2
Given a DP $\mathbf{d} : \mathbf{F} \nrightarrow \mathbf{R}$ we denote by $H_{\mathbf{d}} : \mathbf{F} \to_{\mathbf{Pos}} U\mathbf{R}$ the map that associates to each functionality f the set of minimal resources sufficient to realize f:

$$H_{\mathbf{d}} : \mathbf{F} \to_{\mathbf{Pos}} U\mathbf{R},$$
$$f \mapsto \{r \in \mathbf{R} : \mathbf{d}(f^*, r)\}.$$

If a certain functionality f is infeasible, then $H(f) = \emptyset$.

Symmetrically, the solution to FixResMaxFun is given by a function that we call $K_{\mathbf{d}}$.

Definition 7.3
Given a DP $\langle \mathbf{F}, \mathbf{R}, \mathbf{I}, \mathsf{prov}, \mathsf{req} \rangle$, define the map $K_{\mathbf{d}} : \mathbf{R} \to_{\mathbf{Pos}} L\mathbf{F}$ that associates to each resource r the set of functionalities which can be realized with r:

$$K_{\mathbf{d}} : \mathbf{R} \to_{\mathbf{Pos}} L\mathbf{F},$$
$$r \mapsto \{f \in \mathbf{F} : \mathbf{d}(f^*, r)\}.$$

If a certain resource r only leads to infeasible functionalities, then $K(r) = \emptyset$.

Both maps are indeed monotone.

Lemma 7.4. Both H and K are monotone maps.

See proof on page 225.

A question that arises naturally is whether the map $H_{\mathbf{d}}$ is sufficient to reconstruct the original DP. The answer is yes. We will prove that $H_{\mathbf{d}}$ defines a morphism in a category called $\mathbf{Pos}_U$, and that this category is equivalent to **DP**, therefore being traced monoidal, with a lattice structure. In fact, FixFunMinRes can be seen as a functor from **DP** to $\mathbf{Pos}_U$. Symmetrically, $K_{\mathbf{d}}$ is a morphism in a category $\mathbf{Pos}_L$ equivalent to **DP** and FixResMaxFun can be seen as the functor from **DP** to $\mathbf{Pos}_L$. This situation is represented in Fig. 104.

In the course of this chapter, by defining the two functors FixFunMinRes and FixResMaxFun, we effectively have solved the problem of optimization for DPs in the "mathematical" way. However, this is only the first step, because it does not say anything about whether the functor is actually computable. Further along this chapter we will look at finite approximations of DPs and the computational complexity of the solution.

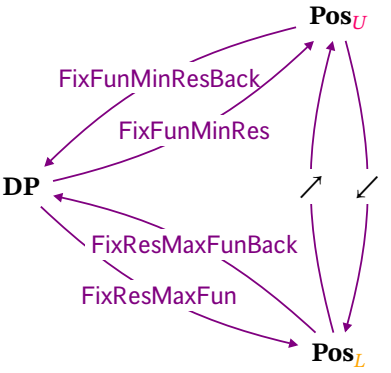


Figure 104: In this chapter, we show that the queries FixResMaxFun and FixFunMinRes can be seen as functors from **DP** to two new categories, $\mathbf{Pos}_U$ and $\mathbf{Pos}_L$. We show that **DP** is equivalent to these categories: a **DP** is univocally defined by the answers to the two queries.

7.2 Categories of solutions

One can see the previous sections as a definition of the category of design problems **DP**, together with its structure. We can define a category of solutions, and eventually show that the two categories are connected by a functor. The notion of solution, in the context of co-design problems, is the one of a map from functionalities to uppersets of resources, or, to antichains of resources.

Definition 7.5 (Category $\mathbf{Pos}_U$)
The category $\mathbf{Pos}_U$ consists of:

1. *Objects*: objects are posets;
2. *Morphisms*: given posets $\mathbf{P}, \mathbf{Q}$, morphisms from $f : \mathbf{P} \to \mathbf{Q}$ are monotone maps of the form $f^\star : \mathbf{P} \to_{\mathbf{Pos}} U\mathbf{Q}$.
3. *Composition of morphisms*: Given morphisms $f : \mathbf{P} \to \mathbf{Q}$, $g : \mathbf{Q} \to \mathbf{R}$,

their composition $f \fatsemi g : \mathbf{P} \to \mathbf{R}$ is given by

$$(f \fatsemi g)^\star : \mathbf{P} \to_{\mathbf{Pos}} U\mathbf{R}$$
$$p \mapsto \bigcup_{q \in f^\star(p)} g^\star(q);$$

4. *Identity morphism*: given an object $\mathbf{P} \in \mathrm{Ob}_{\mathbf{Pos}_U}$, the identity morphism $\mathrm{id}_{\mathbf{P}} : \mathbf{P} \to \mathbf{P}$ is given by the application of the upper closure operator:

$$\mathrm{id}^\star_{\mathbf{P}}(p) := \uparrow\{p\}.$$

Analogously, we can define the $\mathbf{Pos}_L$ category.

Definition 7.6 (Category $\mathbf{Pos}_L$)
The category $\mathbf{Pos}_L$ consists of:

1. *Objects*: objects are posets;
2. *Morphisms*: given posets $\mathbf{P}, \mathbf{Q}$, morphisms $f : \mathbf{P} \to \mathbf{Q}$ are monotone maps of the form $f^\star : \mathbf{P} \to_{\mathbf{Pos}} L\mathbf{Q}$.
3. *Composition of morphisms*: Given morphisms $f : \mathbf{P} \to \mathbf{Q}$, $g : \mathbf{Q} \to \mathbf{R}$, their composition $f \fatsemi g : \mathbf{P} \to \mathbf{R}$ is given by

$$(f \fatsemi g)^\star : \mathbf{P} \to_{\mathbf{Pos}} L\mathbf{R}$$
$$p \mapsto \bigcup_{q \in f^\star(p)} g^\star(q);$$

4. *Identity morphism*: given an object $\mathbf{P} \in \mathrm{Ob}_{\mathbf{Pos}_L}$, the identity morphism $\mathrm{id}_{\mathbf{P}} : \mathbf{P} \to \mathbf{P}$ is given by the application of the lower closure operator:

$$\mathrm{id}^\star_{\mathbf{P}}(p) := \downarrow\{p\}.$$

We now show that $\mathbf{Pos}_U$ and $\mathbf{Pos}_L$ are indeed categories.

Lemma 7.7. $\mathbf{Pos}_U$ and $\mathbf{Pos}_L$ are categories.

See proof on page 225.

We can show that $\mathbf{Pos}_U$ and $\mathbf{Pos}_L$ are equivalent categories.

Lemma 7.8. $\mathbf{Pos}_U$ and $\mathbf{Pos}_L$ are isomorphic: there exists a pair of functors

$$\swarrow : \mathbf{Pos}_U \to \mathbf{Pos}_L,$$
$$\nearrow : \mathbf{Pos}_L \to \mathbf{Pos}_U,$$

such that $\swarrow \fatsemi \nearrow = \mathrm{id}_{\mathbf{Pos}_U}$ and $\nearrow \fatsemi \swarrow = \mathrm{id}_{\mathbf{Pos}_L}$, where $\mathrm{id}_{\mathbf{Pos}_U}$ and $\mathrm{id}_{\mathbf{Pos}_L}$ are the identity functors on $\mathbf{Pos}_U$ and $\mathbf{Pos}_L$, respectively.

See proof on page 226.

$\mathbf{Pos}_U$ and $\mathbf{Pos}_L$ are symmetric monoidal categories

We can show that $\mathbf{Pos}_U$ and $\mathbf{Pos}_L$ are monoidal categories (and will eventually show that they are symmetric monoidal). We show the construction for $\mathbf{Pos}_U$, and the one for $\mathbf{Pos}_L$ is analogous. First, we want to identify the constituents for the monoidal structure on $\mathbf{Pos}_U$. On objects, the monoidal product corresponds to the product of posets. Given two morphisms $f : \mathbf{P} \to \mathbf{Q}$ and $g : \mathbf{R} \to \mathbf{S}$, we have $f \otimes g : \mathbf{P} \times \mathbf{R} \to \mathbf{Q} \times \mathbf{S}$, with

$$\begin{aligned}(f \otimes g)^\star : \mathbf{P} \times \mathbf{R} &\to_{\mathbf{Pos}} U(\mathbf{Q} \times \mathbf{S}) \\ \langle p, r \rangle &\mapsto f^\star(p) \times g^\star(r).\end{aligned}$$

Note that the product of upper sets is an upper set.

Lemma 7.9. The product of upper sets is an upper set. The product of lower sets is a lower set.

See proof on page 227.

The monoidal unit is the singleton poset $\mathbf{1}$. First of all, we check that the monoidal product is functorial.

Lemma 7.10. The monoidal product defined for $\mathbf{Pos}_U$ is functorial.

See proof on page 227.

We now identify left and right unitors. The left unitor $\mathsf{lu}_\mathbf{P} : \mathbf{1} \times \mathbf{P} \to \mathbf{P}$ is given by

$$\begin{aligned}\mathsf{lu}_\mathbf{P}^\star : \mathbf{1} \times \mathbf{P} &\to_{\mathbf{Pos}} U\mathbf{P}, \\ \langle \bullet, p \rangle &\mapsto \uparrow\{p\}.\end{aligned}$$

The right unitor $\mathsf{ru}_\mathbf{P} : \mathbf{P} \times \mathbf{1} \to \mathbf{P}$ is given by

$$\begin{aligned}\mathsf{ru}_\mathbf{P}^\star : \mathbf{P} \times \mathbf{1} &\to_{\mathbf{Pos}} U\mathbf{P}, \\ \langle p, \bullet \rangle &\mapsto \uparrow\{p\}.\end{aligned}$$

Lemma 7.11. The left and right unitors for $\mathbf{Pos}_U$ are valid isomorphisms.

See proof on page 228.

Furthermore, the associator $\mathsf{as}_{\mathbf{P},\mathbf{Q},\mathbf{R}} : (\mathbf{P} \times \mathbf{Q}) \times \mathbf{R} \to \mathbf{P} \times (\mathbf{Q} \times \mathbf{R})$ is given by:

$$\begin{aligned}\mathsf{as}_{\mathbf{P},\mathbf{Q},\mathbf{R}}{}^\star : (\mathbf{P} \times \mathbf{Q}) \times \mathbf{R} &\to_{\mathbf{Pos}} U\mathbf{P} \times (U\mathbf{Q} \times U\mathbf{R}), \\ \langle \langle p, q \rangle, r \rangle &\mapsto \uparrow\{p\} \times (\uparrow\{q\} \times \uparrow\{r\}).\end{aligned}$$

Lemma 7.12. The associator for $\mathbf{Pos}_U$ is a valid isomorphism.

See proof on page 228.

Lemma 7.13. $\langle \mathbf{Pos}_U, \otimes, \mathbf{1}, \mathsf{lu}, \mathsf{ru}, \mathsf{as} \rangle$ is a monoidal category.

See proof on page 229.

We now want to show that $\mathbf{Pos}_U$ can be equipped to become a symmetric monoidal category. The braiding isomorphism for $\mathbf{Pos}_U$ is given by:

$$\begin{aligned} \mathsf{br}^{\star}_{\mathbf{P},\mathbf{Q}} : \mathbf{P} \times \mathbf{Q} &\to_{\mathbf{Pos}} U(\mathbf{Q} \times \mathbf{P}), \\ \langle p, q \rangle &\mapsto \uparrow\{q\} \times \uparrow\{p\}. \end{aligned}$$

Lemma 7.14. The braiding for $\mathbf{Pos}_U$ is a valid natural isomorphism.

See proof on page 231.

Lemma 7.15. $\langle \mathbf{Pos}_U, \otimes, \mathbf{1}, \mathsf{lu}, \mathsf{ru}, \mathsf{as}, \mathsf{br} \rangle$ is a symmetric monoidal category.

See proof on page 232.

$\mathbf{Pos}_U$ and $\mathbf{Pos}_L$ are traced monoidal categories

Definition 7.16 (Trace in $\mathbf{Pos}_U$)
Given a morphism $f : \mathbf{P} \times \mathbf{R} \to \mathbf{Q} \times \mathbf{R}$ in $\mathbf{Pos}_U$, its trace in is defined as a morphism $\mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{Q}}(f) : \mathbf{P} \to \mathbf{Q}$, given by

$$\mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{Q}}(f)^{\star} : \mathbf{P} \to U\mathbf{Q}$$

$$p \mapsto \left\{ q \in \mathbf{Q} \mid \bigvee_{r \in \mathbf{R}} \langle q, r \rangle \in f^{\star}(p, r) \right\}.$$

Lemma 7.17. $\langle \mathbf{Pos}_U, \otimes, \mathbf{1}, \mathsf{lu}, \mathsf{ru}, \mathsf{as}, \mathsf{br} \rangle$ equipped with the trace operation defined in Def. 7.16 is a traced monoidal category.

See proof on page 232.

$\mathbf{Pos}_U$ and $\mathbf{Pos}_L$ also have bounded lattice structure

First of all, we can order morphisms in $\mathbf{Pos}_U$ and $\mathbf{Pos}_L$, carrying the same meaning as the order introduced for morphisms in **DP**.

Definition 7.18 (Order on morphisms in $\mathbf{Pos}_U$ and $\mathbf{Pos}_L$)
Given any two morphisms $f, g : \mathbf{P} \to \mathbf{Q}$ in $\mathbf{Pos}_U$, we define an order between them as

$$\frac{f \preceq_{\mathbf{Pos}_U} g}{f^{\star}(p) \preceq_{U\mathbf{Q}} g^{\star}(p), \quad \forall p \in \mathbf{P}}.$$

Given any two morphisms $f, g : \mathbf{P} \to \mathbf{Q}$ in $\mathbf{Pos}_L$, we define an order between them as

$$\frac{f \preceq_{\mathbf{Pos}_L} g}{f^\star(p) \preceq_{L\mathbf{Q}} g^\star(p), \quad \forall p \in \mathbf{P}}.$$

We can also define union and intersection of morphisms analogously to the **DP** case.

Definition 7.19 (Intersection of morphisms in $\mathbf{Pos}_U$ and $\mathbf{Pos}_L$)
Given two morphisms $f, g : \mathbf{P} \to \mathbf{Q}$ in $\mathbf{Pos}_U$, their *intersection* (meet) is a morphism $f \wedge g : \mathbf{P} \to \mathbf{Q}$, given by

$$\begin{aligned}(f \wedge g)^\star : \mathbf{P} &\to U\mathbf{Q} \\ p &\mapsto f^\star(p) \cap g^\star(p).\end{aligned}$$

Given two morphisms $f, g : \mathbf{P} \to \mathbf{Q}$ in $\mathbf{Pos}_L$, their *intersection* (meet) is a morphism $f \wedge g : \mathbf{P} \to \mathbf{Q}$, given by

$$\begin{aligned}(f \wedge g)^\star : \mathbf{P} &\to L\mathbf{Q} \\ p &\mapsto f^\star(p) \cap g^\star(p).\end{aligned}$$

Definition 7.20 (Union of morphisms in $\mathbf{Pos}_U$ and $\mathbf{Pos}_L$)
Given two morphisms $f, g : \mathbf{P} \to \mathbf{Q}$ in $\mathbf{Pos}_U$, their *union* (join) is a morphism $f \vee g : \mathbf{P} \to \mathbf{Q}$, given by

$$\begin{aligned}(f \vee g)^\star : \mathbf{P} &\to U\mathbf{Q} \\ p &\mapsto f^\star(p) \cup g^\star(p).\end{aligned}$$

Given two morphisms $f, g : \mathbf{P} \to \mathbf{Q}$ in $\mathbf{Pos}_L$, their *union* (join) is a morphism $f \vee g : \mathbf{P} \to \mathbf{Q}$, given by

$$\begin{aligned}(f \vee g)^\star : \mathbf{P} &\to L\mathbf{Q} \\ p &\mapsto f^\star(p) \cup g^\star(p).\end{aligned}$$

Lemma 7.21. Given any $\mathbf{P}, \mathbf{Q} \in \mathrm{Ob}_{\mathbf{Pos}_U}$, $\mathrm{Hom}_{\mathbf{Pos}_U}(\mathbf{P}; \mathbf{Q})$ is a bounded lattice with union $\vee$ of morphisms in $\mathbf{Pos}_U$ as join, intersection $\wedge$ of morphisms in $\mathbf{Pos}_U$ as meet, least upper bound $\top_{\mathrm{Hom}_{\mathbf{Pos}_U}(\mathbf{P};\mathbf{Q})} : \mathbf{P} \to \mathbf{Q}$ given by

$$\begin{aligned}\top_{\mathrm{Hom}_{\mathbf{Pos}_U}(\mathbf{P};\mathbf{Q})}{}^\star : \mathbf{P} &\to U\mathbf{Q} \\ p &\mapsto \emptyset,\end{aligned}$$

and greatest lower bound $\bot_{\mathrm{Hom}_{\mathbf{Pos}_U}(\mathbf{P};\mathbf{Q})} : \mathbf{P} \to \mathbf{Q}$ given by

$$\begin{aligned}\bot_{\mathrm{Hom}_{\mathbf{Pos}_U}(\mathbf{P};\mathbf{Q})}{}^{\star} : \mathbf{P} &\to U\mathbf{Q}\\ p &\mapsto \mathbf{Q}.\end{aligned}$$

See proof on page 237.

Lemma 7.22. Given any $\mathbf{P}, \mathbf{Q} \in \mathrm{Ob}_{\mathbf{Pos}_U}$, $\mathrm{Hom}_{\mathbf{Pos}_L}(\mathbf{P};\mathbf{Q})$ is a bounded lattice with intersection $\wedge$ of morphisms in $\mathbf{Pos}_L$ as meet, union $\vee$ of morphisms in $\mathbf{Pos}_L$ as join, least upper bound $\top_{\mathrm{Hom}_{\mathbf{Pos}_L}(\mathbf{P};\mathbf{Q})} : \mathbf{P} \to \mathbf{Q}$ given by

$$\begin{aligned}\top_{\mathrm{Hom}_{\mathbf{Pos}_L}(\mathbf{P};\mathbf{Q})}{}^{\star} : \mathbf{P} &\to L\mathbf{Q}\\ p &\mapsto \mathbf{Q},\end{aligned}$$

and greatest lower bound $\bot_{\mathrm{Hom}_{\mathbf{Pos}_U}(\mathbf{P};\mathbf{Q})} : \mathbf{P} \to \mathbf{Q}$ given by

$$\begin{aligned}\bot_{\mathrm{Hom}_{\mathbf{Pos}_L}(\mathbf{P};\mathbf{Q})}{}^{\star} : \mathbf{P} &\to L\mathbf{Q}\\ p &\mapsto \emptyset.\end{aligned}$$

Proof. The proof is analogous to the one of Lemma 7.21. Note that meets/joins and top/bottom are switched in meaning, because of the difference in order between $U\mathbf{P}$ and $L\mathbf{P}$. □

7.3 Queries as functors from statements to solutions

We are now ready to prove the functoriality from problem statements to solutions.

Lemma 7.23. There is a functor

$$\mathsf{FixFunMinRes} : \mathbf{DP} \to \mathbf{Pos}_U \tag{50}$$

that maps:

1. An object (poset) in **DP** to the same object (poset) in $\mathbf{Pos}_U$.
2. A morphism $\mathbf{e} \in \mathrm{Hom}_{\mathbf{DP}}(\mathbf{F};\mathbf{R})$ to the morphism $H_{\mathbf{e}} \in \mathrm{Hom}_{\mathbf{Pos}_U}(\mathbf{F};\mathbf{R})$, where:

$$\begin{aligned}H_{\mathbf{e}}^{\star} : \mathbf{F} &\to_{\mathbf{Pos}} U\mathbf{R}\\ f &\mapsto \{r \in \mathbf{R} \mid \mathbf{e}(f^*, r)\}.\end{aligned}$$

See proof on page 237.

Lemma 7.24. There is a functor

$$\mathsf{FixResMaxFun} : \mathbf{DP} \to \mathbf{Pos}_L$$

which maps:

1. An object (poset) of **DP** to the same object (poset) in $\mathbf{Pos}_L$.
2. A morphism $\mathbf{e} \in \mathrm{Hom}_{\mathbf{DP}}(\mathbf{F}; \mathbf{R})$ to the morphism $K_{\mathbf{e}} \in \mathrm{Hom}_{\mathbf{Pos}_L}(\mathbf{R}; \mathbf{F})$, where:

$$\begin{aligned} K^{\star}_{\mathbf{e}} : \mathbf{R} &\to_{\mathbf{Pos}} L\mathbf{F} \\ r &\mapsto \{f \in \mathbf{F} \mid \mathbf{e}(f^*, r)\}. \end{aligned}$$

Proof. The proof is analogous to the one of Lemma 7.23. □

Lemma 7.25. There is a functor $\mathsf{FixFunMinResBack} : \mathbf{Pos}_U \to \mathbf{DP}$ which maps:

1. An object (poset) in $\mathbf{Pos}_U$ to the same object (poset) in **DP**.
2. A morphism $g \in \mathrm{Hom}_{\mathbf{Pos}_U}(\mathbf{F}; \mathbf{R})$ to the morphism $\mathbf{d}_g \in \mathrm{Hom}_{\mathbf{DP}}(\mathbf{F}; \mathbf{R})$, where:

$$\begin{aligned} \mathbf{d}_g : \mathbf{F}^{\mathrm{op}} \times \mathbf{R} &\to_{\mathbf{Pos}} \mathbf{Bool} \\ \langle f^*, r\rangle &\mapsto r \in g^{\star}(f). \end{aligned}$$

See proof on page 238.

Lemma 7.26. There is a functor $\mathsf{FixResMaxFunBack} : \mathbf{Pos}_L \to \mathbf{DP}$ which maps:

1. An object (poset) in $\mathbf{Pos}_U$ to the same object (poset) in **DP**.
2. A morphism $g \in \mathrm{Hom}_{\mathbf{Pos}_L}(\mathbf{F}; \mathbf{R})$ to the morphism $\mathbf{d}_g \in \mathrm{Hom}_{\mathbf{DP}}(\mathbf{F}; \mathbf{R})$, where:

$$\begin{aligned} \mathbf{d}_g : \mathbf{F}^{\mathrm{op}} \times \mathbf{R} &\to_{\mathbf{Pos}} \mathbf{Bool} \\ \langle f^*, r\rangle &\mapsto f \in g^{\star}(r). \end{aligned}$$

Proof. The proof is analogous to the one of Lemma 7.25. □

Lemma 7.27. The pair of functors FixFunMinRes and FixFunMinResBack together with the natural isomorphisms

$$\mathsf{FixFunMinRes} \fatsemi \mathsf{FixFunMinResBack} \cong \mathrm{id}_{\mathbf{DP}},$$

and

$$\mathsf{FixFunMinResBack} \fatsemi \mathsf{FixFunMinRes} \cong \mathrm{id}_{\mathbf{Pos}_U},$$

form an equivalence for **DP** and $\mathbf{Pos}_U$.

See proof on page 239.

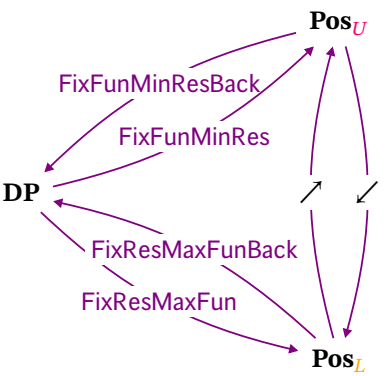


Figure 105: From **DP** to $\mathbf{Pos}_U$ and $\mathbf{Pos}_L$, and back.

Lemma 7.28. FixFunMinRes preserves the bounded lattice structure.

See proof on page 239.

Lemma 7.29. FixFunMinRes preserves traces.

See proof on page 240.

The introduced functors actually have more structure. Let's recall the notion of strong monoidal functor.

Definition 7.30 (Strong monoidal functor)
Let $\langle \mathbf{C}, \otimes_{\mathbf{C}}, 1_{\mathbf{C}} \rangle$ and $\langle \mathbf{D}, \otimes_{\mathbf{D}}, 1_{\mathbf{D}} \rangle$ be two monoidal categories.
A *strong monoidal functor* between $\mathbf{C}$ and $\mathbf{D}$ is given by:

1. A functor
$$F : \mathbf{C} \to \mathbf{D}; \tag{51}$$
2. An isomorphism
$$\mathrm{iso} : 1_{\mathbf{D}} \to F(1_{\mathbf{C}}); \tag{52}$$
3. A natural isomorphism μ
$$\mu_{X,Y} : F(X) \otimes_{\mathbf{D}} F(Y) \to F(X \otimes_{\mathbf{C}} Y), \quad \forall X, Y \in \mathbf{C}, \tag{53}$$

satisfying the following conditions:

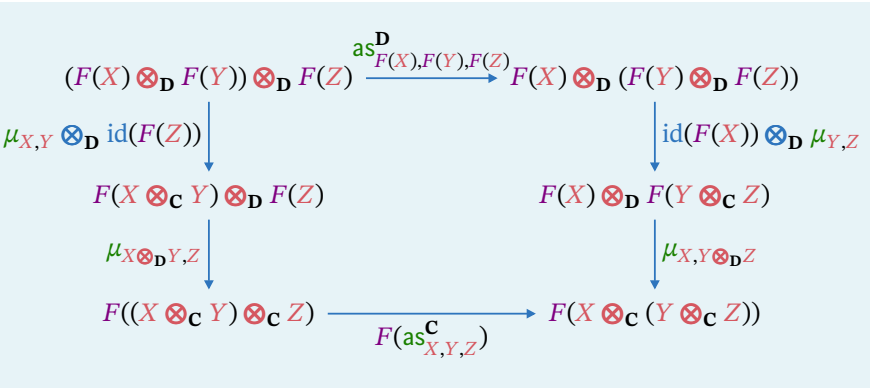
(a) *Associativity*: For all objects $X, Y, Z \in \mathbf{C}$, there are *associators* $\mathrm{as}^{\mathbf{C}}$ and $\mathrm{as}^{\mathbf{D}}$ such that the diagram in Fig. 106a commutes.

(b) *Unitality*: For all $X \in \mathbf{C}$, there exist left and right *unitors* $\mathrm{lu}^{\mathbf{C}}$ and $\mathrm{ru}^{\mathbf{C}}$, the diagram in Fig. 106b commutes.

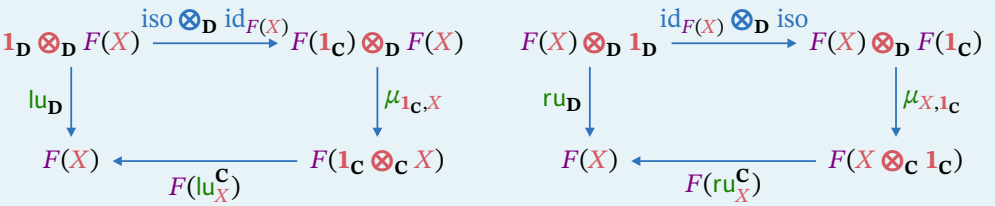
We can now prove the following result.

Lemma 7.31. FixFunMinRes and FixResMaxFun are strong monoidal functors.

See proof on page 241.



(a) Natural associativity



(b) Natural unitality

Figure 106: Commuting diagrams used in Def. 7.30

7.4 Finite co-design problems

If we want a computable algorithm for solving co-design queries, it is necessary that the solution can be finitely representable. One way to do this is to zero-in on those design problems that are guaranteed, by construction, to have a finite solution. This is what we do in this section. For other cases, we refer to [123].

In the FixFunMinRes queries, the solution lives in an upper set of resources. We now look at upper sets that can be represented as the upper closure of finite antichains.

Definition 7.32 (Finitely-supported upper sets)
Given a poset $\mathbf{P}$, we call an upper set $\mathbf{S} \in U\mathbf{P}$ *finitely supported* if it can be written as the upper closure of a finite antichain:

$$\mathbf{S} = (\uparrow \alpha), \text{ for } \alpha \in \mathsf{Anti}\,\mathbf{P}, \mathsf{card}(\alpha) < \infty.$$

We call $\overline{U}_{\mathrm{f}}\mathbf{P}$ the set of finitely-supported upper sets of a poset $\mathbf{P}$.
We call $\mathsf{Anti}_{\mathrm{f}}\ \mathbf{P}$ the set of finite antichains.

For brevity, in the following we denote $\mathsf{FixFunMinRes}(\mathbf{d})$ by $H_{\mathbf{d}}$.

Definition 7.33 (Finite design problems)
We call a design problem $\mathbf{d} : \mathbf{P} \nrightarrow \mathbf{Q}$ finite if, in its representation $H_{\mathbf{d}} : \mathbf{F} \to U\mathbf{R}$, $H_{\mathbf{d}}(f) \in \overline{U}_{\mathrm{f}}\mathbf{R}$ for all $f \in \mathbf{F}$.

We show that finite co-design problems form a subcategory of **DP** that is also monoidal and locally posetal. (Note that we are leaving out "traced" for now.) To show this, we just need to check that all the ways to compose finite DPs result in finite DPs. The formulas that we derive also describe an algorithm to compute the solution to the queries.

Definition 7.34 (Category of finite design problems $\mathbf{Pos}_{\overline{U}_{\mathrm{f}}}$)
The category of *finite design problems* $\mathbf{Pos}_{\overline{U}_{\mathrm{f}}}$ consists of the following constituents:

1. *Objects*: The objects are posets.
2. *Morphisms*: The morphisms are *finite* design problems (Def. 7.33).
3. *Identity morphism*: The identity morphism $\mathrm{id}_{\mathbf{P}} : \mathbf{P} \nrightarrow \mathbf{P}$ is as in **DP**.
4. *Composition operation*: Given morphisms $\mathbf{d} : \mathbf{P} \nrightarrow \mathbf{Q}$ and $\mathbf{e} : \mathbf{Q} \nrightarrow \mathbf{R}$, their composition $\mathbf{d} \fatsemi \mathbf{e} : \mathbf{P} \nrightarrow \mathbf{R}$ is as in **DP**.

Lemma 7.35. The series composition of finite design problems gives a finite design problem.

See proof on page 241.

Lemma 7.36. The parallel interconnection of finite design problem gives a finite design problem.

See proof on page 241.

Lemma 7.37. The intersection and union of finite design problems gives finite design problems.

Proof. The proof follows simply from the fact that union and intersection of finite upper sets returns finite upper sets. □

7.5 Domain theory and fixed points

In this section we recall some fundamentals of domain theory. It is used in computer science for defining denotational semantics (see *e.g.*, [149]). It is used in embedded systems for defining the semantics of models of computation (see, *e.g.*, [150]). What we need from domain theory is the least necessary to define *least fixed points* and to use Kleene's theorem.

Domain theory builds on order theory by defining "directed" and "complete" partial orders. These attributes play the same role as compactness in analysis: they will be used to make sure that certain sequences can converge to a fixed point.

Directed and complete partial orders

Definition 7.38 (Directed set)
In a poset $\mathbf{P} = \langle \mathbf{P}, \leq_{\mathbf{P}} \rangle$, we say that a set $\mathbf{S} \subseteq \mathbf{P}$ is *directed* if each pair of elements in $\mathbf{S}$ has an upper bound: for all $x, y \in \mathbf{S}$, there exists $z \in \mathbf{S}$ such that $x \leq z$ and $y \leq z$.

Definition 7.39 (Completeness)
A poset is a *directed complete partial order* (DCPO) if each of its directed subsets has a supremum (least of upper bounds). It is a *complete partial order* (CPO) if it also has a bottom.

Example 7.40 (Completion of $\mathbb{R}_{\geq 0}$ to $\overline{\mathbb{R}}_{\geq 0}$)**.** The poset $\langle \mathbb{R}, \leq \rangle$ is not a CPO, because it lacks a bottom.

The non-negative reals $\mathbb{R}_{\geq 0} = \{x \in \mathbb{R} : x \geq 0\}$ have a bottom $\bot = 0$, however, they are not a DCPO because some of their directed subsets do not have an upper bound. For example, take $\mathbb{R}_{\geq 0}$, which is a subset of $\mathbb{R}_{\geq 0}$. Then $\mathbb{R}_{\geq 0}$ is directed, because for each $a, b \in \mathbb{R}_{\geq 0}$, there exists $c = \max\{a, b\} \in \mathbb{R}_{\geq 0}$ for which $a \leq c$ and $b \leq c$.

One way to make $\langle \mathbb{R}_{\geq 0}, \leq \rangle$ a CPO is by adding an artificial top element $\top$ that we think as "a point at infinity". We can define then the completion

$$\overline{\mathbb{R}}_{\geq 0} := \mathbb{R}_{\geq 0} \cup \{\top\},$$

and extending the partial order $\leq$ so that $a \leq \top$ for all $a \in \mathbb{R}_{\geq 0}$.

Example 7.41. Any lattice is a DCPO.

Example 7.42. For any poset $\mathbf{P}$, $U\mathbf{P}$ is a CPO, because it is a bounded lattice.

Scott continuity

Scott continuity is a property of maps on DCPOs that is slightly stronger than monotonicity.

Definition 7.43 (Scott continuity)
A map $f : \mathbf{P} \to_{\mathbf{Pos}} \mathbf{Q}$ between DCPOs is *Scott continuous* iff for each directed subset $\mathbf{S} \subseteq \mathbf{P}$, the image $f(\mathbf{S})$ is directed, and

$$f(\mathsf{Sup}\,\mathbf{S}) = \mathsf{Sup}\, f(\mathbf{S}). \tag{54}$$

Lemma 7.44. Scott continuity implies monotonicity.

See proof on page 242.

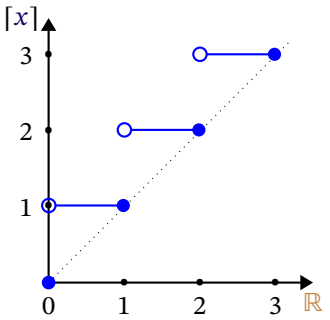


Figure 107: The ceiling function is Scott continuous.

Remark 7.45. Scott continuity is not the same as the notion of continuity as used in analysis you might be familiar with. A map from the CPO $\langle \overline{\mathbb{R}}_{\geq 0}, \leq \rangle$ to itself is Scott continuous iff it is nondecreasing and left-continuous. For example, the ceiling function $x \mapsto \lceil x \rceil$ is not continuous in the usual sense, but it is Scott continuous (Fig. 107).

However, the name "continuity" for this property is aptly chosen. In analysis, a function is continuous if it preserves limits, in the sense that

$$\lim_{n\to\infty} f(a_n) = f(\lim_{n\to\infty} a_n),$$

which is, in spirit, the same as (54).

Least fixed points

Definition 7.46 (Fixed points)
A *fixed point* of $f : \mathbf{P} \to_{\mathbf{Pos}} \mathbf{P}$ is a point x such that $f(x) = x$.

Definition 7.47 (Least fixed points)
A *least fixed point* of $f : \mathbf{P} \to_{\mathbf{Pos}} \mathbf{P}$ is the minimum (if it exists) of the set of fixed points of f:

$$\mathsf{lfp}(f) \;:=\; \min_{\leq} \{x \in \mathbf{P} : \; f(x) = x\}. \tag{55}$$

In general, a function need not have a fixed point. It also might have multiple fixed points; and also it that case there are might not be a *least* fixed point.

However, the conditions for a least fixed point to exist are quite weak. Monotonicity of the map f plus completeness is sufficient to ensure existence.

Lemma 7.48. If $\mathbf{P}$ is a CPO and $f : \mathbf{P} \to_{\mathbf{Pos}} \mathbf{P}$ is monotone, then $\mathsf{lfp}(f)$ exists and is unique.

This is given as CPO Fixpoint Theorem II, 8.22 in [125].

With the additional assumption of Scott continuity, Kleene's algorithm is a systematic procedure to find the least fixed point.

Lemma 7.49 (Kleene's fixed-point theorem). Assume $\mathbf{P}$ is a CPO, and $f : \mathbf{P} \to_{\mathbf{Pos}} \mathbf{P}$ is Scott continuous. Then the least fixed point of f is the supremum of the Kleene ascent chain

$$\bot \leq f(\bot) \leq f(f(\bot)) \leq \cdots \leq f^{(n)}(\bot) \leq \cdots . \tag{56}$$

This is given as CPO fixpoint theorem I, 8.15 in [125].

Example: party invite

Consider the case in which a subset $\mathbf{S} \subseteq \mathbf{A}$ of people decide to throw a party. They then proceed to call all their friends, who accept, and, if they were not invited already, enthusiastically call *their* friends to extend the invite. We want to find out what is the final group of people that will show up at the party. We call this map $\phi : \mathsf{Pow}\,\mathbf{A} \to \mathsf{Pow}\,\mathbf{A}$, so that if $\mathbf{S}$ is the initial group, $\phi(\mathbf{S})$ is the complete set of invites.

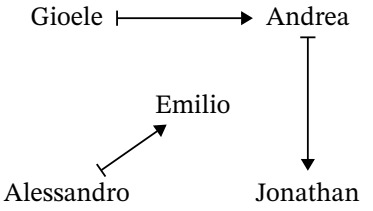


Figure 108: Party invite relation.

Note that this is related to the transitive closure operation, but we are only interested in the transitive closure from a certain initial set $\mathbf{S}$.

For example, consider the case in which the relation is as in Fig. 108. In this case, we would have

$$\phi(\emptyset) = \emptyset,$$

which means that, if nobody starts a party, no party takes place. Jonathan does not invite anybody, so we would have

$$\phi(\{\text{Jonathan}\}) = \{\text{Jonathan}\}$$

If Gioele and Alessandro start the party, everybody will get invited:

$$\phi(\{\text{Alessandro, Gioele}\}) = \text{everybody}.$$

We can show that

1. The function ϕ can be computed as a fixed point.
2. The recursive invite strategy corresponds to Kleene's iteration.

We summarize the properties that we want the function ϕ to have. Given an initial subset $\mathbf{S}$, we would like to find the set of people $\mathbf{T} = \phi(\mathbf{S})$ such that:

1. $\mathbf{T}$ contains the initial set $\mathbf{S}$:

$$\mathbf{S} \subseteq \mathbf{T}$$

2. $\mathbf{T}$ is closed with respect to a certain invite relation $R : \mathbf{A} \to \mathbf{A}$. If $x\,R\,y$, then x invites y to the party. Define the function

$$\begin{aligned} m : \mathsf{Pow}\,\mathbf{A} &\to \mathsf{Pow}\,\mathbf{A}, \\ \mathbf{T} &\mapsto \mathbf{T} \cup \bigcup\nolimits_{x \in \mathbf{T}} \{y \in \mathbf{A} : x\,R\,y\}. \end{aligned}$$

This represents one iteration of the invite process: given a set $\mathbf{T}$, we add to $\mathbf{T}$ all invitees of each of the elements of $\mathbf{T}$.

We are looking for a set $\mathbf{T}$ such that it is a fixed point of the invite function:

$$\mathbf{T} = m(\mathbf{T}).$$

3. $\phi(\mathbf{S})$ is the smallest among all such sets that satisfy the two conditions above.

Let $\mathbf{P}$ be the upper principal set of $\mathbf{S}$: given Item 1, we know that we want sets that contain at least $\mathbf{S}$:

$$\mathbf{P} = {\uparrow}\mathbf{S} = \{\mathbf{T} \in \mathsf{Pow}\,\mathbf{A} : \mathbf{S} \subseteq \mathbf{T}\}.$$

The poset $\mathbf{P}$ is a sublattice of $\mathsf{Pow}\,\mathbf{A}$. Note also that the bottom of $\mathbf{P}$ is $\mathbf{S}$.

In summary, we are looking for the smallest point of $\mathbf{P}$ that is closed to m:

$$\phi(\mathbf{S}) = \min_{\subseteq} \{\mathbf{T} \in \mathbf{P} : \mathbf{T} = m(\mathbf{T})\}$$

Comparing this with (55), we see that $\phi(\mathbf{S})$ is the least fixed point of m:

$$\phi(\mathbf{S}) = \mathsf{lfp}(m).$$

Take Kleene's iteration in (56):

$$\bot \preq f(\bot) \preq f(f(\bot)) \preq \cdots \preq f^{(n)}(\bot) \preq \cdots.$$

Because the bottom of $\mathbf{P} = {\uparrow}\mathbf{S}$ is $\mathbf{S}$, we can rewrite it as:

$$\mathbf{S} \subseteq m(\mathbf{S}) \subseteq m(m(\mathbf{S})) \subseteq m(m(m(\mathbf{S}))) \ldots.$$

Each element of the sequence corresponds to one iteration of the invite algorithm.

7.6 Handling loops

We are close to having a complete solution. The only part that is missing is dealing with loops (trace).

The feedback operator has signature

$$\mathsf{loop} : (\mathbf{A} \times \mathbf{B} \nrightarrow \mathbf{B}) \to (\mathbf{A} \nrightarrow \mathbf{B})$$

The following theorem establishes a closed form for $h_{\mathsf{loop}(\mathbf{d})}$ as a least fixed point (note that we use h the map which maps to antichains, and not upper sets). Here on we consider $\mathsf{Anti}_{\mathsf{f}}\ \mathbf{R}$ as a poset with the order given by

$$\frac{\alpha_1 \preceq_{\mathsf{Anti}_{\mathsf{f}}\ \mathbf{R}} \alpha_2}{\uparrow \alpha_1 \preceq_{\overline{U}_{\mathsf{f}}\mathbf{R}} \uparrow \alpha_2}.$$

Theorem 7.50 (From Thm. 2 in [122])**.** For any DP $\mathbf{d}$ of the right shape, we can compute $h_{\mathsf{loop}(\mathbf{d})}$ as follows:

$$h_{\mathsf{loop}(\mathbf{d})} : f_1 \mapsto \uparrow \mathsf{lfp}(\Phi_{f_1}),$$

that is, as the *least fixed point* of a map Φ_{f_1} defined as

$$\begin{aligned} \Phi_{f_1} : \mathsf{Anti}_{\mathsf{f}}\ \mathbf{R} &\to \mathsf{Anti}_{\mathsf{f}}\ \mathbf{R}, \\ \alpha &\mapsto \operatorname*{Min}_{\preceq_{\mathbf{R}}} \bigcup_{r \in \alpha} h_{\mathbf{d}}(f_1, r) \cap \uparrow r. \end{aligned} \tag{57}$$

See proof on page 242.

Lemma 7.51. Let $\mathbf{S}$ be an antichain in $\mathbf{P}$. Then

$$\frac{x \in \mathbf{S}}{\{x\} = \mathbf{S} \cap \uparrow x}.$$

Lemma 7.52. For $\mathbf{S}, \mathbf{T} \in \mathsf{Anti}_{\mathsf{f}}\ \mathbf{P}$, and $\mathbf{A} \subseteq \mathbf{B}$, $\mathbf{S} \preceq_{\mathsf{Anti}_{\mathsf{f}}\ \mathbf{R}} \mathbf{T}$ implies $\mathbf{S} \cap \mathbf{A} \preceq_{\mathsf{Anti}_{\mathsf{f}}\ \mathbf{R}} \mathbf{T} \cap \mathbf{A}$.

Lemma 7.53. For $\mathbf{S}, \mathbf{T}, \mathbf{U}, \mathbf{V} \in \mathsf{Anti}_{\mathsf{f}}\ \mathbf{P}$, $\mathbf{S} \preceq_{\mathsf{Anti}_{\mathsf{f}}\ \mathbf{R}} \mathbf{U}$ and $\mathbf{T} \preceq_{\mathsf{Anti}_{\mathsf{f}}\ \mathbf{R}} \mathbf{V}$ implies $\mathbf{S} \cup \mathbf{T} \preceq_{\mathsf{Anti}_{\mathsf{f}}\ \mathbf{R}} \mathbf{U} \cup \mathbf{V}$.

Lemma 7.54. Let $f : \mathbf{P} \times \mathbf{Q} \to_{\mathbf{Pos}} \mathbf{Q}$ be Scott continuous. For each $x \in \mathbf{P}$, define the map

$$f_x : y \mapsto f(x, y)$$

Then the map

$$f^\dagger : x \mapsto \mathsf{lfp}(f_x)$$

is Scott continuous.

Proof. Davey and Priestly [125] leave this as Exercise 8.26. A proof is found in Gierz *et al.* [151, Exercise II-2.29]. □

Guarantees of Kleene ascent

Solving an CDP with cycles reduces to computing a Kleene ascent sequence α_k. At each instant k we have some additional guarantees.

For any finite k, the resources "below" α_k (the set $\mathbf{R} \setminus \uparrow \alpha_k$,) are infeasible.

If the iteration converges to a non-empty antichain α_∞, the antichain α_∞ divides $\mathbf{R}$ in two. Below the antichain, all resources are infeasible. However, above the antichain, it is not necessarily true that all points are feasible, because there might be holes in the feasible set. Note that this method does not compute the entire feasible set, but rather only the *minimal elements* of the feasible set, which might be much easier to compute.

Finally, if the sequence converges to the empty set, it means that there are no solutions. The sequence α_k can be considered a certificate of infeasibility.

Complexity of the solution

Consider first the case of a DP that can be described as $\mathbf{d} = \mathsf{loop}(\mathbf{d}_0)$, where $\mathbf{d}_0$ is an DP that is described only using the series and par operators. Suppose that $\mathbf{d}_0$ has resource space $\mathbf{R}$. Then evaluating h for $\mathbf{d}$ is equivalent to computing a least fixed point iteration on the space of antichains $\mathsf{Anti}\,\mathbf{R}$. This allows to give worst-case bounds on the number of iterations.

Proposition 7.55 (Prop. 5 in [122])**.** Suppose that $\mathbf{d} = \mathsf{loop}(\mathbf{d}_0)$ and $\mathbf{d}_0$ has resource space $\mathbf{R}_0$ and evaluating h_0 takes at most c computation. Then we can obtain the following bounds for the algorithm's resources usage:

memory	$O(\mathsf{width}(\mathbf{R}_0))$
number of steps	$O(\mathsf{height}(\mathsf{Anti}\,\mathbf{R}_0))$
total computation	$O(\mathsf{width}(\mathbf{R}_0) \cdot \mathsf{height}(\mathsf{Anti}\,\mathbf{R}_0) \cdot c)$

See proof on page 243.

Remark 7.56 (Considering relations with infinite cardinality)**.** In [152], Censi presents a solution for the case in which $\mathsf{width}(\mathbf{R}_0)$ is infinite, so that one needs to represent a continuum of solutions. For instance, suppose that the platform to be designed must travel a distance d [m], and we need to choose the endurance T [s] and the velocity v [m/s]. The relation among the quantities is $d \leq T\,v$. This is a design problem described by the map

$$H : \overline{\mathbb{R}}_{\geq 0} \quad \to \quad \mathsf{Anti}\,\overline{\mathbb{R}}_{\geq 0} \times \overline{\mathbb{R}}_{\geq 0},$$

$$d \quad \mapsto \quad \{\langle T, v\rangle \in \overline{\mathbb{R}}_{\geq 0} \times \overline{\mathbb{R}}_{\geq 0} \;:\; d = T\, v\}.$$

For each value of d, there is a continuum of solutions. One approach to solving this problem would be to discretize the functionality **F** and the resources **R** by sampling and/or coarsening. However, sampling and coarsening makes it hard to maintain completeness and consistency. One effective approach (described in the paper) is to *approximate the design problem itself*, rather than the spaces **F**, **R**, which are left as possibly infinite. The basic idea is that an infinite antichain can be bounded from above and above by two antichains that have a finite number of points. This idea leads to an algorithm that, given a prescribed computation budget, can compute an inner and outer approximation to the solution antichain.

7.7 Example: Optimizing over the natural numbers

This is a simple example, adapted from [122], that can show two interesting properties of CDPIs:

1. the ability to work with discrete posets; and
2. the ability to treat multi-objective optimization problems.

Consider the family of optimization problems indexed by $c \in \mathbb{N}$:

$$\begin{cases} \mathrm{Min}_{\preceq_{\overline{\mathbb{N}}\times\overline{\mathbb{N}}}} & \langle x, y\rangle, \\ \text{s.t.} & x + y \geq \lceil\sqrt{x}\,\rceil + \lceil\sqrt{y}\,\rceil + c. \end{cases} \tag{58}$$

One can show that this optimization problem is a CDP by producing a co-design diagram with an equivalent semantics, such as the one in Fig. 109.

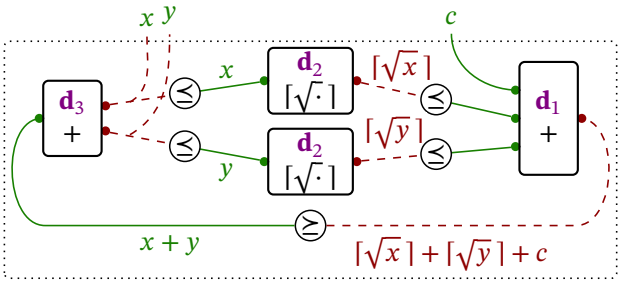


Figure 109: Co-design diagram equivalent to (58)

The diagram contains three primitive DPIs: $\mathbf{d}_1$, $\mathbf{d}_2$ (used twice), and $\mathbf{d}_3$. Their h maps are:

$$\begin{aligned} h_1 : \overline{\mathbb{N}} \times \overline{\mathbb{N}} \times \overline{\mathbb{N}} \quad &\to \quad \mathsf{Anti}_{\mathsf{f}}\, \overline{\mathbb{N}}, \\ \langle f_1, f_2, f_3\rangle \quad &\mapsto \quad \{f_1 + f_2 + f_3\}, \end{aligned}$$

$$
\begin{aligned}
h_2 : \overline{\mathbb{N}} &\to \mathrm{Anti}_{\mathrm{f}}\, \overline{\mathbb{N}},\\
f &\mapsto \{\lceil \sqrt{f}\, \rceil\},\\
h_3 : \overline{\mathbb{N}} &\to \mathrm{Anti}_{\mathrm{f}}\, (\overline{\mathbb{N}} \times \overline{\mathbb{N}}),\\
f &\mapsto \{\langle a, b\rangle \in \overline{\mathbb{N}} \times \overline{\mathbb{N}} : a + b = f\}.
\end{aligned}
$$

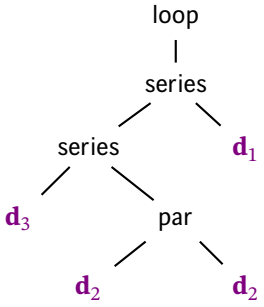


Figure 110: Tree decomposition of the problem.

The tree decomposition (Fig. 110) corresponds to the expression

$$\mathbf{d} = \mathsf{loop}(\mathsf{series}(\mathsf{par}(\mathbf{d}_2, \mathbf{d}_2), \mathsf{series}(\mathbf{d}_1, \mathbf{d}_3))). \tag{59}$$

From (59) we obtain an expression for h:

$$h = ((h_2 \otimes h_2) \circledcirc h_1 \circledcirc h_3)^{\dagger}. \tag{60}$$

This problem is small enough that we can write down an explicit expression for h. By substituting in (60) the definitions for $\otimes, \dagger, \circledcirc$, we obtain that evaluating $h(c)$ means finding the least fixed point of a map Ψ_c:

$$h : c \mapsto \mathsf{lfp}(\Psi_c).$$

The map $\Psi_c : \mathrm{Anti}_{\mathrm{f}}\, (\overline{\mathbb{N}} \times \overline{\mathbb{N}}) \to \mathrm{Anti}_{\mathrm{f}}\, (\overline{\mathbb{N}} \times \overline{\mathbb{N}})$ can be obtained from Theorem 7.50 as follows:

$$\Psi_c : \alpha \mapsto \mathrm{Min} \bigcup_{\langle x, y\rangle \in \alpha} \uparrow\langle x, y\rangle \cap$$

$$\cap \left\{\langle a, b\rangle \in \mathbb{N}^2 : (a + b \geq \lceil \sqrt{x}\, \rceil + \lceil \sqrt{y}\, \rceil + c)\right\}.$$

Kleene's algorithm is the iteration $\alpha_{k+1} = \Psi_c(\alpha_k)$ starting from

$$\alpha_0 = \bot_{\mathrm{Anti}_{\mathrm{f}}\, (\overline{\mathbb{N}} \times \overline{\mathbb{N}})} = \{\langle 0, 0\rangle\}.$$

For $c = 0$, the sequence converges immediately:

$$\alpha_0 = \{\langle \mathbf{0}, \mathbf{0}\rangle\} = h(0).$$

For $c = 1$, the sequence converges at the sixth step; however, some solutions (in

bold) converge sooner:

$$\begin{aligned}
\alpha_0 &= \{\langle 0, 0\rangle\},\\
\alpha_1 &= \{\langle 0, 1\rangle, \langle 1, 0\rangle\},\\
\alpha_2 &= \{\langle 0, 2\rangle, \langle 1, 1\rangle, \langle 2, 0\rangle\},\\
\alpha_3 &= \{\langle \mathbf{0}, \mathbf{3}\rangle, \langle 1, 2\rangle, \langle 2, 1\rangle, \langle \mathbf{3}, \mathbf{0}\rangle\},\\
\alpha_4 &= \{\langle \mathbf{0}, \mathbf{3}\rangle, \langle 2, 2\rangle, \langle \mathbf{3}, \mathbf{0}\rangle\},\\
\alpha_5 &= \{\langle \mathbf{0}, \mathbf{3}\rangle, \langle \mathbf{3}, \mathbf{0}\rangle\} = h(1).
\end{aligned}$$

For $c = 2$, the sequence converges at the fifth step; however, some solutions (in bold) converge sooner:

$$\begin{aligned}
\alpha_0 &= \{\langle 0, 0\rangle\},\\
\alpha_1 &= \{\langle 0, 2\rangle, \langle 1, 1\rangle, \langle 2, 0\rangle\},\\
\alpha_2 &= \{\langle \mathbf{0}, \mathbf{4}\rangle, \langle 1, 3\rangle, \langle 2, 2\rangle, \langle 3, 1\rangle, \langle \mathbf{4}, \mathbf{0}\rangle\},\\
\alpha_3 &= \{\langle \mathbf{0}, \mathbf{4}\rangle, \langle 3, 2\rangle, \langle 2, 3\rangle, \langle \mathbf{4}, \mathbf{0}\rangle\}\\
\alpha_4 &= \{\langle \mathbf{0}, \mathbf{4}\rangle, \langle \mathbf{3}, \mathbf{3}\rangle, \langle \mathbf{4}, \mathbf{0}\rangle\} = h(2).
\end{aligned}$$

The next values in the sequence are:

$$\begin{aligned}
h(3) &= \{\langle \mathbf{0}, \mathbf{6}\rangle, \langle \mathbf{3}, \mathbf{4}\rangle, \langle \mathbf{4}, \mathbf{3}\rangle, \langle \mathbf{6}, \mathbf{0}\rangle\},\\
h(4) &= \{\langle \mathbf{0}, \mathbf{7}\rangle, \langle \mathbf{3}, \mathbf{6}\rangle, \langle \mathbf{4}, \mathbf{4}\rangle, \langle \mathbf{6}, \mathbf{3}\rangle, \langle \mathbf{7}, \mathbf{0}\rangle\}.
\end{aligned}$$

7.8 Example: co-designing an autonomous drone

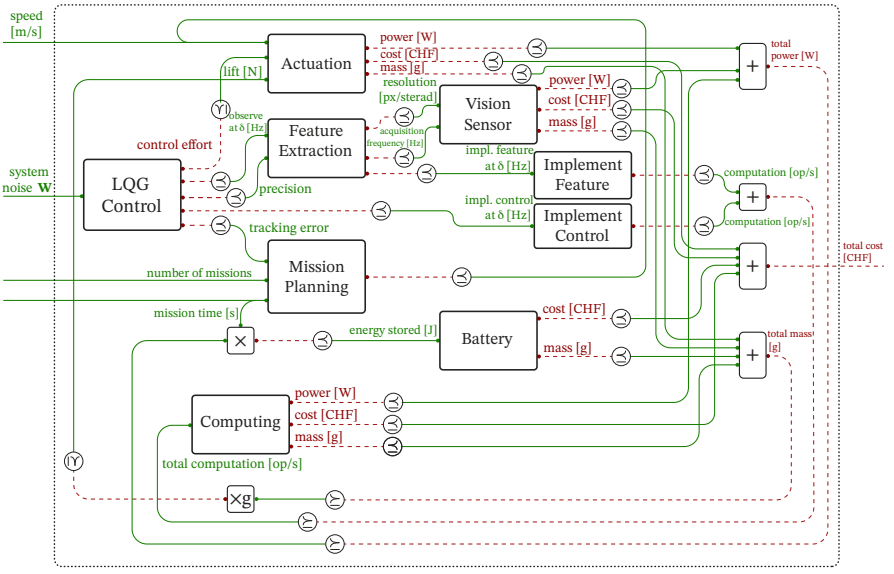


Figure 111: Co-design diagram for the design of an autonomous drone that needs to execute an idealized search-and-rescue task. The functionalities are task characteristics and the environment. We choose costs as the resources to minimize.

Recall the co-design problem of an autonomous drone introduced in Section 5.5. We now want to show how the proposed framework is able to solve the co-design problem of an autonomous drone, co-designing the controller synthesis together with the rest of the platform. We consider the design problem reported in Fig. 111 and provide design insights in terms of cost, power consumption, tracking error, missions duration and number of missions. We list the design variables in Table 4. Specifically, they include the selection of sensors, control parameters, battery technologies, computing units and actuators.

Table 4: Variables, options and sources for the drone co-design problem.

Variable	Options	Source
Actuators	Type 1, 2, 3	[122]
Computer	RaspPi 4B, Jetson Nano/TX1/TX2	[153]
Control	0.2-50.0 Hz, $\alpha \in [10^{-4}, 10^{4}]$	-
Sensor	Basler Ace251gm/222gm/13gm/7gm/5gm/15um	[154]
	Flir Pointgrey/Blackfly/BlackflyBoard	[155]
Battery	LCO, LFP, LiPo, LMO, NiCad, NiH2, NiMH, SLA	[122]

Cost and performance trade-off

We consider a variation of the co-design problem in Fig. 111 in which the resources are the total cost, and the tracking error (i.e., an indicator of performance). In practice, we are dealing with a design problem $\mathbf{d} : \mathsf{PSDM}(2) \times \overline{\mathbb{R}}_{\geq 0} \times \overline{\mathbb{R}}_{\geq 0} \nrightarrow \overline{\mathbb{R}}_{\geq 0} \times \overline{\mathbb{R}}_{\geq 0}$, where the functionalities are the system noise, the number of missions, and the mission duration, and the resources are the total cost and the tracking error. In the following, we bound the system noise and fix $q_0 = r_0 = 1$, to propose a sample of design insights that the framework can produce.

We query (using FixFunMinRes) the optimal design solutions which enable the drone to perform 5,000 missions lasting 40 minutes (Fig. 112). The red dots represent the elements belonging to the antichain of optimal design solutions, expressed in terms of the platform cost and the tracking error. The solutions are not comparable, since no instance leads simultaneously to lower cost and tracking error. The upper set of resources (not necessarily optimal, but making the design problem feasible) is given in solid red. Furthermore, we report implementations corresponding to specific optimal solutions. For instance, the solution with the lowest cost (and the highest tracking error), consists in using a Nano computer, a control frequency of 6.25 Hz, $\alpha = 3.73$, an LCO battery, the first kind of actuators, and a Pointgrey camera. As can be gathered from the plot, a budget increase for the drone reduces the tracking error. For instance, an investment from 900 CHF to 1,100 CHF reduces the tracking error by 90 %. This kind of plots helps the stakeholders involved in the design process making decisions. For instance, a 27 % investment from 1,100 CHF to 1,400 CHF only reduces the tracking error by 5 %, and one might think that such investment is not a good

idea for most applications.

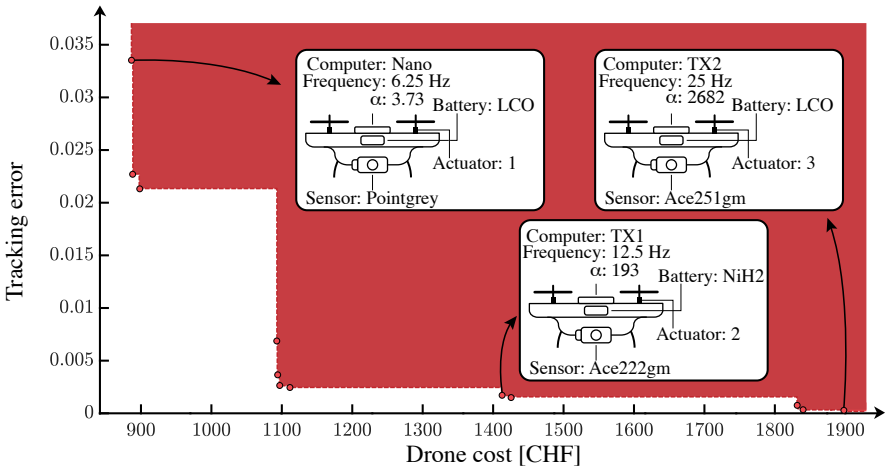


Figure 112: Pareto front of cost and tracking error (performance) in the design of a drone, able to complete 5,000 missions of 40 minutes. The figure shows the antichain of optimal solutions for the given scenario. Red dots characterize optimal design solutions and the colored area describes upper sets of resources for which functionalities are feasible. Selected implementations corresponding to specific points in the antichain are reported.

Power and performance trade-off

We consider a variation of the co-design problem in Fig. 111 in which the resources are the total power needed, and the tracking error (i.e., an indicator of performance). In practice, we are dealing with a design problem $\mathbf{d}$: PSDM(2) $\times$ $\overline{\mathbb{R}}_{\geq 0} \times \overline{\mathbb{R}}_{\geq 0} \nrightarrow \overline{\mathbb{R}}_{\geq 0} \times \overline{\mathbb{R}}_{\geq 0}$, where the functionalities are the system noise, the number of mission, and the mission duration, and the resources are the total power needed and the tracking error. We query the co-design problem in a similar way, now looking at the trade-offs between power consumption and tracking error for the case in which the drone must complete 1,000 missions lasting 10 minutes (Fig. 113). When more power is available, better sensors and more performing computers, batteries and actuators can be used, reducing the tracking error. Interestingly, solutions needing beyong 16 W do not seem to substantially reduce the tracking error.

Monotonocity in the co-design problem of the drone

We consider increasing mission time and number of missions and assess the evolution in trade-offs in platform power consumption and tracking error. Fig. 114 shows multiple co-design queries. In particular, for each functionality (left plot), we compute the map $h_{\mathbf{d}}$, which maps a functionality to the minimum antichain of resources which provide it. Monotonicity can be seen in the dominance of subsequent Pareto fronts (right plot), illustrated in increasing red tonality.

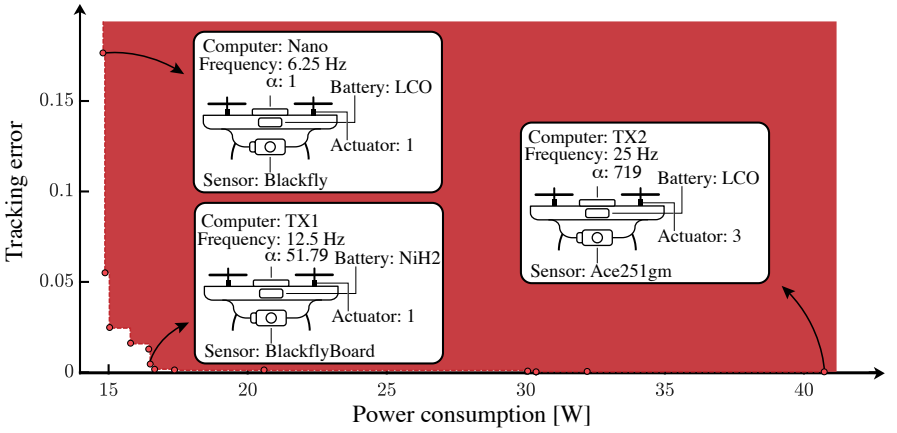


Figure 113: Pareto front of power consumption and tracking error (performance) in the design of a drone, able to complete 1,000 missions of 10 minutes.

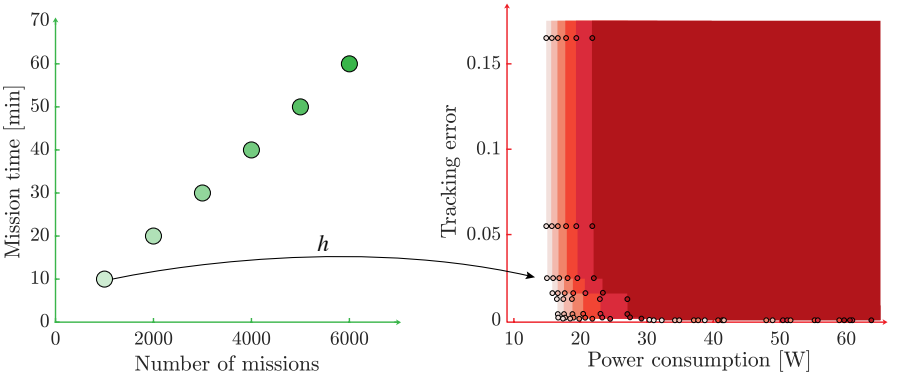
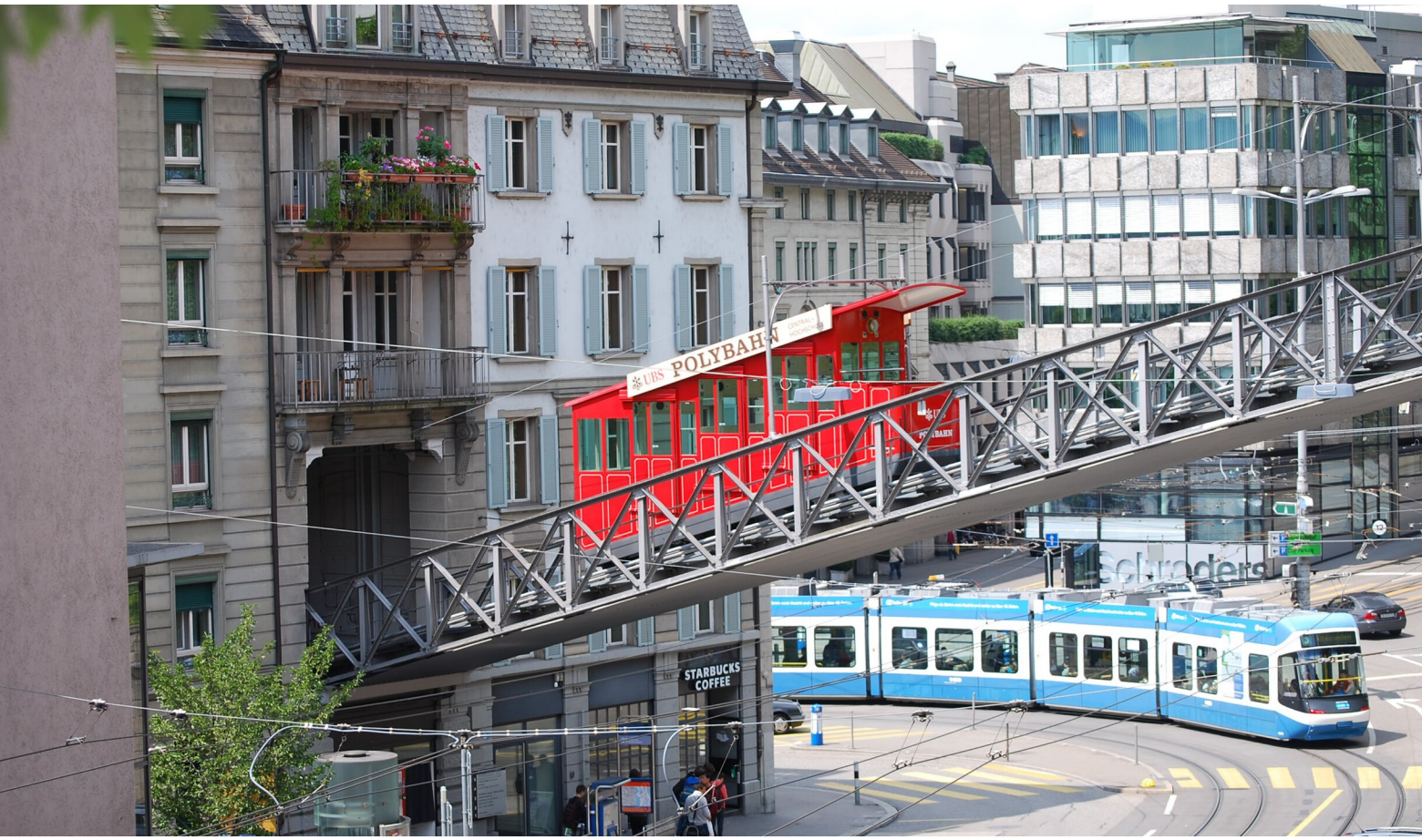


Figure 114: Monotonicity of the drone design problem with implementation (DPI). Higher mission time and number of missions requires higher power and tracking error.

From autonomy to future mobility | Part B



The *Polybahn* is an autonomous funicular railway that connects the Central square with the terrace by the main building of ETH Zürich. It was opened in 1889 and it is owned by the banking group UBS AG.

Systematic process for the co-design of complex systems

8

Divide each difficulty into as many parts as is feasible and necessary to resolve it.

—René Descartes[16]

In the previous part, we have presented a theory of co-design and have shown some examples motivating its potential in dealing with complex design problems. However, all the examples we provided so far, while compelling, were defined deductively, as opposed to inductively. In this chapter, we will present a systematic process to employ the presented theory to come up with task-driven co-design models for complex systems. We will start by showing how to define a function/-task (Section 8.1), and how to decompose it into simpler pieces (Section 8.2). Given the functional decomposition, we will then explain the procedure to turn it into a co-design diagram (Section 8.3), and to identify circular constraints (Section 8.4).

8.1 Defining the task

When designing a complex system, the first thing one needs to do is to identify the reason for the system to exist in the first place: its *function*. In the remainder of this manuscript, we will refer to "functions" (typical in the systems engineering jargon) as "tasks", since the term "function" clashes with the co-design term "functionalities".

Why are we designing the system? In his popular paper "The structure of invention", Arthur says that humans develop technology *as a means to fulfill a purpose* [156]. For instance, we design a mobility system (e.g., a fleet of AVs providing robotaxi services) to satisfy a certain demand (i.e., people willing to move from certain origins to certain destinations), or an autonomous drone to monitor a certain area (e.g., in the context of gas leak detections). But how do we go about identifying these crucial functions or tasks? At first sight, it might seem that the universe of things we care about when designing a complex system is too large to be characterized. However, there are structured ways in which one can identify key tasks, which transcend domains.

Example 8.1 (Complex systems classification)**.** For instance, De Weck and collaborators identify a list of basic tasks, characterized by the "process" at play, and whatever the process is acting on (denoted as "operands") [18], [157]. The three basic operands are "matter" (e.g., vehicles, water, oil, packages), "energy" (in various forms), and "information" (e.g., a signal, an e-mail). The three basic pro-

[16] Descartes was a French philosopher, scientist, and mathematician.

cesses acting on the operands are “transforming”, “transporting”, and “storing”. Table 5 exemplifies these concepts with practical cases. De Weck also explains how one can extend operands and processes (e.g., adding money as operand, and exchange as process).

Table 5: Processes and operands examples [18, Table 2.2], [158].

Process	Matter	Energy	Information
Transforming	Furnace	Engine, electric motor	Calculator
Transporting	Train,car, airplane	Power grid	Cables, internet
Storing	Warehouse	Battery	Book, disk

Example 8.2 (Functional specifications for cyber-physical systems)**.** In formal robotics, one often refers to tasks as “functional specifications”, and typically specified using some sort of language (e.g., via temporal logic formulas, assume-guarantee contracts, etc.). Given the specifications, one then designs algorithms that lead to behaviors which comply to the specifications (e.g., generating trajectories which satisfy certain constraints on the dynamics, and promote certain performance metrics). Examples works in this area include [37], [92], [109], [159]–[162] and references therein.

In this chapter, we take robots as a proxy for complex systems involving components arising from heterogeneous domains. In the context of co-designing such a system, we care about identifying a task for each component (physical or virtual) to be designed. In particular, each task is a design problem, in which functionalities are the environment to which one has to be robust to, and task-specific performance metrics. Usually, providing these functionalities comes at several costs (resources), which vary depending on the context.

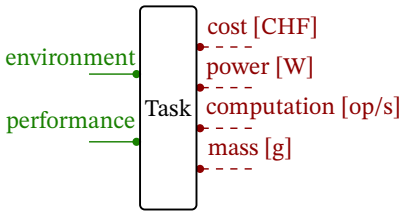


Figure 115: General design problem for a robotic task.

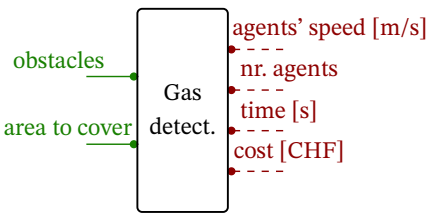


Figure 116: Design problem for a swarm of drones which need to detect gas leakages.

Example 8.3 (Robotics)**.** The characterization of environments and function-specific performances varies depending on the particular application. For instance, in the context of designing a swarm of drones which need to cover a particular area to detect gas leakages, the environment might be characterized by the topology of the area to be covered (e.g., outdoor vs. indoor, presence of static/-dynamic obstacles, weather conditions). In this case, one could characterize the task-specific performance metric via the area to be covered (the higher, the more complex), or via the probability of detecting a leakage, given that there is one. For what concerns the resources, typical ones in robotics include the shape (of the robot), weight (which adds to the payload), power needs, computation needs, and monetary costs. On top of these, one might have task-specific ones. For instance, when designing a swarm of drones, one might care about the speed limitations of each agent, the time needed to cover a particular area, or the number of agents needed to cover it. Fig. 115 and Fig. 116 represent a general and specific design problems for the aforementioned robotic tasks.

Examples treated in this chapter go beyond robotics.

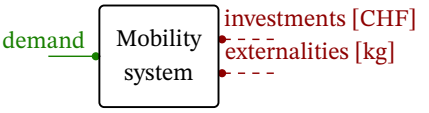


Figure 117: Design problem for a mobility system.

Example 8.4 (Mobility)**.** When designing a mobility option (e.g., a fleet of robotaxis), the environment could be characterized by the mobility network and its status (e.g., roads), and the task could be specified in form of demand (i.e., people willing to move from certain origins to certain destinations). The task-specific performance by the average travel time in the system. Typical resources in this context are externalities (e.g., emissions), and investment costs (e.g., to buy and maintain a fleet) Fig. 117.

Example 8.5 (Powertrain design)**.** When designing the powertrain of an electric vehicle, the environment could be characterized by the actual inclination profile of the roads to be navigated (e.g., a flat highway in southern Italy, vs. the steep and curvy Lombard Street in San Francisco). The task is typically specified by a drive cycle (usually provided in terms of required acceleration/velocity), and the performance is measured in terms of power consumption, motor efficiency, comfort. In racing cases (e.g., Formula 1), one cares about driving time.

Once the task is clear, it is time to decompose it unto specific sub-tasks.

8.2 Functional decomposition

Once the main task has been identified, one needs to perform a *functional decomposition* exercise. In other words, one needs to decompose the main task in sub-tasks, each of which we can model as design problems.

In the literature, this is often referred to as "function structure" or "functional requirements". For instance, Pahl explains how to decompose the function of a system in subsystems which exchange signals, materials, and energy [163], and Shankar et al. presents an overview of functional requirements in engineering design [164].

Here, we are interested in a hierarchical decomposition of the main task [18]. Following the "V model" for system design, originally developed by the German department of defense [116], this section deals with the "development of a functional architecture".

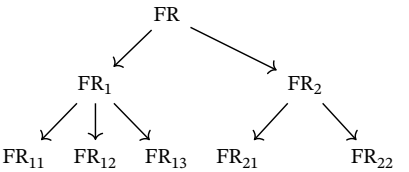


Figure 118: Functional decomposition for the refrigerator example.

Example 8.6 (Refrigerator design [29])**.** In [29], Suh presents a principle to decompose the design of a complex systems, following functional requirements (FR). For instance, a refrigerator needs to freeze food for long-term preservation (FR_1), and keep some food at cold temperature without freezing for short-term preservation (FR_2). For FR_1, we can further decompose the task into controlling the temperature of the freezer in a particular range (FR_{11}), mantain a uniform temperature (FR_{12}), control humidity of the freezer at a certain level (FR_{13}). Similarly, the second requirement can be further decomposed into controlling the

temperature of the chiller in a particular range (FR_{21}), and maintaining a uniform temperature in the chiller (FR_{22}). This decomposition is depicted in Fig. 118. Note that Suh states the "independence axiom" by saying that different functional requirements should not influence each other, creating couplings, calls designs violating this principle "flawed". This is not a problem for our formulation. Rather, violating this axiom is what makes a system interesting. We discuss this concept more in Section 8.4.

The resulting decomposition should not have a notion of the actual implementation needed to solve a particular task, but rather only focus on the sub-tasks involved. This idea, referred to as *orthogonalization of concerns* (see [110]) is centeral to this theory, and separates what the system is supposed to do (function), from how the system does it (implementation). More discussions related to this concept are provided in Section 9.1.

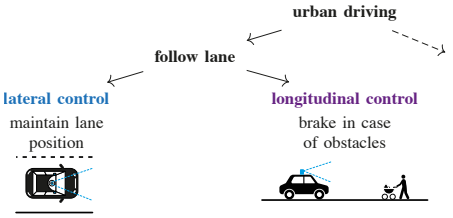


Figure 119: Functional decomposition for the task of urban driving.

As a guiding example, we consider the need for designing an AV (Fig. 119). In this context, the main task is the one of "urban driving". It can be decomposed into various sub-tasks, for instance handling intersections, following a lane, picking up passengers, etc. For the sake of the narrative, we focus on "lane following". This sub-task, can be further decomposed into two more specifc tasks: "lateral control" (i.e., maintaing the lane position), and "longitudinal control" (i.e., accelerate and brake in presence of obstacles).

8.3 From functional decompositions to co-design diagrams

A big drawback of function decompositions as the ones presented in [163] is that while intuitive, they do not lead to formal representations. We have presented a theory of co-design which is at the same time intuitive and practical, and formal, and we now want to map a functional decomposition produced as described in Section 8.2 to a formal co-design diagram. To do so, starting from a functional decomposition, we need to find the components, express them as design problems, and interconnect them via functionalities and resources constraints, by identifying common patterns.

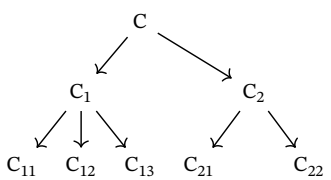


Figure 120: Decomposition in components for the refrigerator example.

Example 8.7 (Refrigerator design)**.** In the language of Suh, this step is equivalent to finding components in the "physical domain" (although we also allow non-physical components) [29]. Continuing Example 8.6, one can identify the components "freezer section" (C_1) and "chiller section" (C_2). Given the task of controlling the temperature of the freezer/chiller, we need a sensor-compressor system which turns the compressor on (off) when the air temperature is higher (lower) than the set temperature in the freezer/chiller section (C_{11}/C_{21}). Similarly, maintaining a uniform temperature in the freezer/chiller happens via a air

circulation system which blows air into the freezer/chiller section (C_{12}/C_{22}), and a condenser condenses the moisture to maintain the desired humidity (C_{13}). For a graphical representation, see Fig. 120.

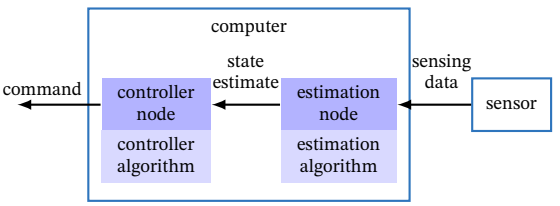
Finding components

We first have to implement a paradigm shift, from data flow, to logical dependencies. For instance, when thinking about decision making for autonomous robots, we usually adopt a data flow reasoning: some sensor produces sensing data, which is elaborated by an estimation algorithm, which produces a state estimate, with is fed into a control algorithm, which in turn produces a command for the actuators (Fig. 121a). Co-design diagrams are not data flow diagrams, but rather highlight logical dependencies. In this context, decision making *requires* state estimation, which *requires* sensing data, algorithms, and computation, which in turn *require* a sensor, programmers, and a computer, respectively (Fig. 121b).

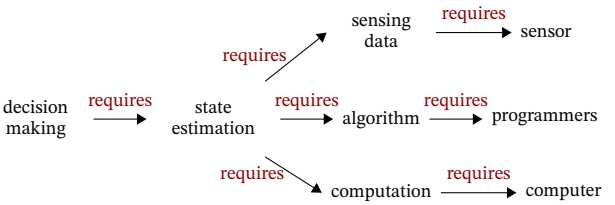
Now, given a functional decomposition, we want to identify the components at play. Slightly modifying Haiken's famous quote, our slogan is

> *A system is composed of components;*
> *a component is something you understand* ***how to design****.*

In other words, we want to identify the design problems for which we know how to express the implementation space (via the different modeling techniques presented before).



(a) Data flow between sensing data, state estimation, and decision making.



(b) Logical dependency between decision making, state estimation, and sensing data.

Figure 121: Data flow vs. logical dependencies.

When this process is over, it is time to interconnect the components.

Interconnecting the components

Given the decomposition of a task in specific sub-tasks, we feed the specific scenarios/environments into each sub-task, and a general task performance is assigned to the problem (Fig. 123). In particular, the functional decomposition design problem has knowledge of the task decomposition logic, and knows for each

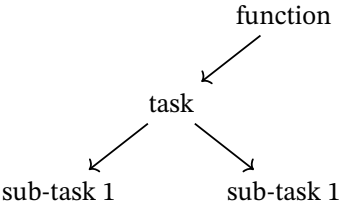


Figure 122: Functional decomposition schematics.

task performance level which combinations of performance levels are required. The resources required by different sub-tasks are generated independently and then merged via an associative operation (e.g., max or +). In robotics, these would be the resources presented in Example 8.3. Note the compositionality property of this formalization: the resulting diagram has the same interface as the one of a task, meaning that a composite task is a task.

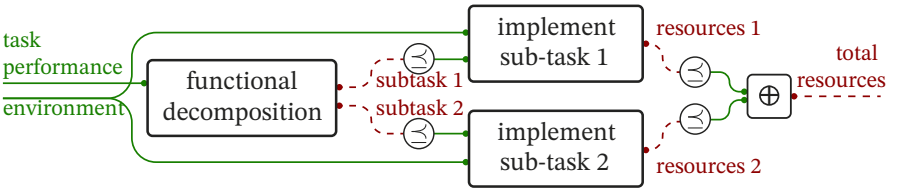


Figure 123: Functional decomposition provides us with sub-tasks, each of which can be modeled as a design problem.

Example: lateral control

Going back to the functional decomposition for urban driving (Fig. 119), we now show the aforementioned steps in practice for the sub-task of lateral control. We first want to identify the components involved. Subsequently, we want to interconnect them in a coherent co-design diagram.

To perform lateral control, one needs several ingredients. Essentially, one needs to:

- Employ lane cameras to observe the lane;
- Feed such information to a control algorithm to maintain the lane position;

For each of these processes, we need to ask ourselves the question: "do we know how to design it?". In practice, simple sentences like the above, hide multiple processes, which might feature separate design exerices. For instance, a new iteration would result in:

- Employ lane cameras to produce raw observations of the lanes;
- Extract information from the data produced by the lane cameras (feature extraction) via an algorithm;
- Feed such information to a control algorithm, designed to maintain the lane position;
- Actually implement the algorithms (control, and feature extraction).

At this point, we are ready to identify the components.

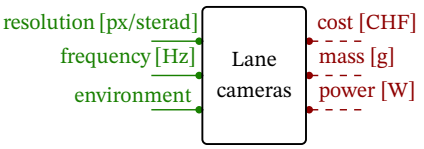


Figure 124: Design problem for lane cameras.

Lane cameras Lane cameras exist to provide measurements at a specific frequency, with a specific resolution (usually measured in px/sterad), in a specific environment (e.g., rainy vs. sunny, day vs. night). The burden they bring

is typically expressed in terms of their monetary cost, their mass, and power consumption. This gives rise a design problem as the one in Fig. 124.

Feature extraction Usually, a feature extraction algorithm processes the measurements provided by the cameras, producing observations which are fed to the control algorithm at a certain frequency and with a certain precision (e.g., as expressed as covariance matrix). These quantities depend on the frequency at which the feature extraction algorithm actually is operating, the actual resolution of the raw images, and the acquisition frequency of the cameras. This gives rise to a design problem as the one in Fig. 125

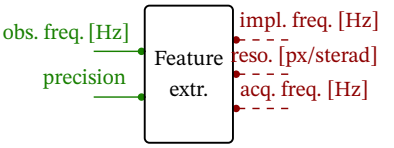


Figure 125: Design problem for feature extraction.

Lane control Given the task of the AV to align itself with the lane, typically one will consider a system with some noise (to which the controller should be robust to), and defining the control law will require observations at a certain frequency and carrying certain information. Furthermore, one usually tries to minimize tracking error and control effort, and is limited by the frequency at which the control law can be generated. This gives rise to a design problem as the one in Fig. 126.

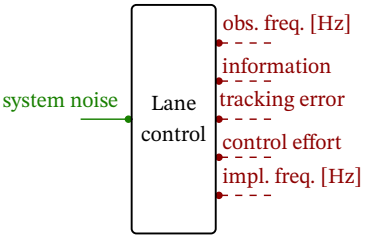


Figure 126: Design problem for lane control.

Algorithms implementation Both feature detection and lane control algorithms need to be actually implemented. Typically, implementing an algorithm can be done at a certain frequency, and costs computation. This gives rise to two analogous design problems, as the one in Fig. 127.

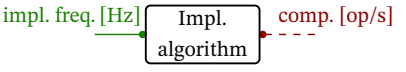


Figure 127: Design problem for the implementation of algorithms.

Interconnecting the diagram Once we have identified all the components, and defined the structure of the respective design problems, it is inevitable to interconnect them. The lane control design problem requires the observations provided by the feature extraction algorithm (at a certain frequency and with a certain precision), which in turn require the raw sensing data provided by the lane cameras (at a certain frequency, and resolution). Both the lane control and feature extraction algorithms are implemented at a certain frequency, requiring a certain computation, which can be summed. All the other unconnected functionalities and resources are the ones of the resulting interconnected design problem: lateral control is robust to a specific system noise and environment, and requires monetary cost, mass, power and computation, together with tracking error and control effort. If the reader agrees that each block is a design problem, then their composition is a design problem.

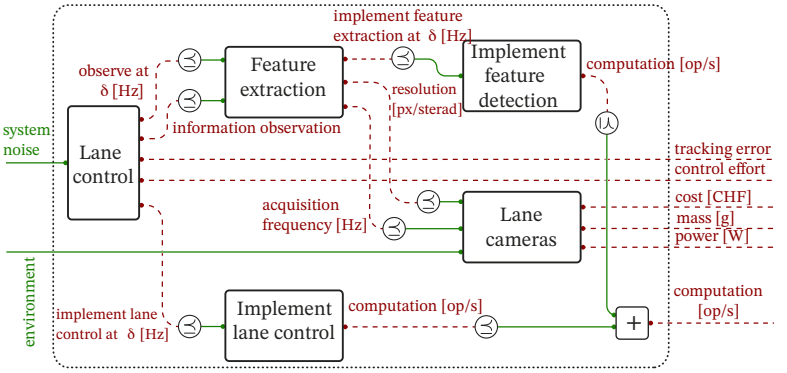


Figure 128: We consider lateral control as design problem, involving the design of control strategies and feature detection algorithms, together with sensor selection. Resources are cost, mass, power, computation, control effort and tracking error.

8.4 Finding feedback loops

When designing complex systems, the computational complexity arises mainly from *recursive* constraints. Historically, some design methodologies explicitly avoided handling such constraints. For instance, in Suh's theory of axiomatic design, the first axiom dictates to keep the design functional requirements orthogonal (i.e., do not introduce cycles) [28], [29]. In particular, when looking at Example 8.6 and Example 8.7, this means there cannot interdependencies between different requirements/components. For instance, in case of limited power budget, the choice of design for the temperature control of the freezer might influence the other processes.

The theory of co-design not only embraces the complexity arising from recursive constraints, but also deals with them in a formal way. Below, some examples of physical and abstract feedback loops are reported.

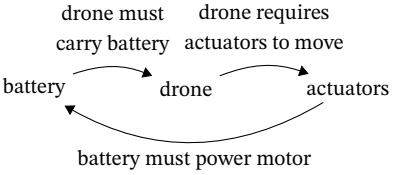


Figure 129: Recursive constraints when designing a drone.

Physical interpretation

Physical recursive constraints are easy to identify. For instance, in the context of designing an autonomous drone, the drone requires actuators to move, which are powered by a battery, which must be carried by the drone. Here, a larger battery provides more power, but it also increases the payload, increasing power needs (Fig. 129).

When looking at the design of the controllers for the drone, typically a better state estimate requires better sensors, which cost more and increase the payload, or require more computation, increasing the power needs.

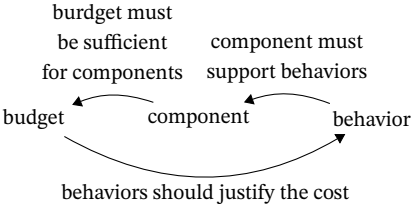


Figure 130: Recursive constraints between engineering problems and business cases.

Abstract interpretation

At a more abstract level, we can find recursive constraints between an engineering problem and its business case. In practice, any designed component must support certain behaviors, which should be implemented in a way that justifies the cost

(i.e., profitability for the system). At the same time, however, the budget must be sufficient to produce the component (Fig. 130).

An example of this concept can be found in designing mobility systems, where the infrastructural investments should be justified (i.e., covered) by an improvement in ridership and system performance.

Implementation 9

In theory, theory and practice are the same. In practice, they are not.
—Albert Einstein

So far, we have presented a formal framework in which one can formulate and solve co-design problems. While intuitive, the formalism might seem heavy to understand, featuring relationships between posets, interconnections of several problems, and interpretations in category theory. In this chapter, we will show that the "scary" part is the *developer* viewpoint, and that there exists a *user* one, which is very friendly. In particular, we will first guide the reader through the creation of co-design problems (via co-design skeletons and their populations), show some examples, and present a user-friendly interface.

9.1 Writing a skeleton

When implementing the lessons learned so far to co-design a system, the first step is to write a skeleton of a co-design diagram. The skeleton acknowledges the *logical* dependencies between different components, and ideally transcends modeling details which populate the feasibility relations. For instance, consider the skeleton expressing the logical dependencies of the design of a chassis, requiring torque to move, provided by an electric motor (Fig. 131).

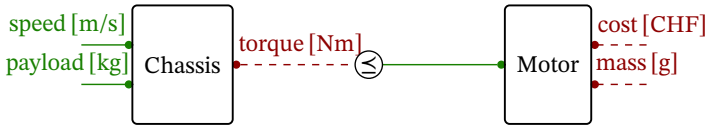


Figure 131: Co-design skeletons transcend model implementations.

This diagram represents the relationships between the two design problems, but does not specify how one should populate them. For the population, one might simply use catalogues of existing chassis and motors. At a more granular level, instead, one might consider complex FEM simulations for the design of a chassis, and more complex designs for electric motors.

Following this principle allows the designer to separate dependencies reasoning and actual model implementations, and hence avoid to fall into the trap of "starting from the solution and not from the problem". Notably, this principle was already introduced by Ferrari and Sangiovanni-Vincentelli in 1999 [110].

The trap of starting from the solution instead of the problem

When delving into the co-design of a complex system, our initial focus should be on formulating the problem itself, setting aside preconceived notions of solu-

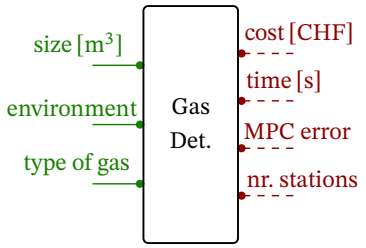


Figure 132: Designing a system to detect gas leakages in a factory-like environment.

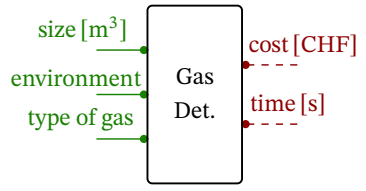


Figure 133: Designing a system to detect gas leakages in a factory-like environment.

tions.

Consider the scnario of designing a system tasked with detecting gas leaks in a factory-like environment. Our primary concerns are to minimize both the overall system cost and the average time required to detect a gas leak. In your laboratory, you have developed a setup utilizing drones that efficient carry out exploration tasks. These drones make use of visual sensors and employ model predictive control to navigate their sorroundings effectively. Given this capability, it is natural to conceptualize the co-design problem as in Fig. 132. We identify key functionalities, such as the size of the gas plume to be detected, the specific environment in which the system will operate, and the type of gas to be detected. Additionally, we consider various resources, including system cost, time required to detect the leaks, the accuracy of the controller, and the number of charging stations required for the drones. You populate the design problems with the components of the usual autonomy stack you use, and consider as design parameters the number of drones needed, as well as some specific control parameters.

While these choices may seem valid at first glance, the inadvertently steer us towards starting with a predetermined solution in mind. We assume that the optimal system must necessarily involve a swarm of drones utilizing vision-based model predictive control, with an inherent need for charging stations. This limited perspective may inadvertently narrow the solution space, preventing us from exploring potentially superior alternatives.

In reality, the solution space for this problem is far more expansive than our initial assumptions suggest. We should consider alternative approaches, such as ground-based robots equipped with onboard sensors, different types of flying robots (like the concept of "roboflies" charging via Wi-Fi stations [165], [166]), or even hybrid fleets combining multiple robot types. Another option might involve stationary sensors, strategically placed within the environment, without the need for any mobile robots. Additionally, exploring different control schemes, such as learning-based approaches, could yield innovative solutions.

The key takeaway here is that by prematurely prescribing specific components and strategies, we risk overspecifying our problem model. To address this issue effectively, our aim should be to identify the common ground shared by all potential solutions. In this context, this common ground would lead us to formulate a more encompassing design problem, as reported in Fig. 133. In this revised problem formulation, we focus primarily on initial performance metrics that are applicable to all heterogeneous solutions. Parameters like the number of charging stations (if necessary) and the controller error are now integrated into the total cost calculation, and they play a role in determininig the average time required to detect the gas leaks. By taking this broader approach, we open the door to a more flexible and comprehensive exploration of potential solutions, ultimately enabling us to arrive at a more robust and effective system for detecting gas leaks

in a factory-like environment.

Context informs the level of detail

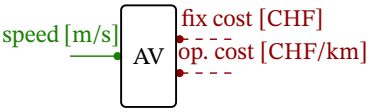


Figure 134: Designing an AV in the context of co-designing a mobility system.

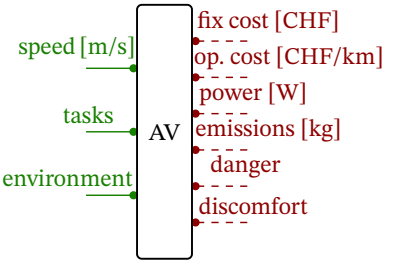


Figure 135: Designing an AV in the context of co-designing the autonomy stack.

The importance of context becomes clear when we contemplate the intricacies of co-design. Some might inquire: "Ok, your theory appears sound, but isn't the real magic in the art of selecting these wires and their interconnections? How does one make these critical choices?". As we progress through the upcoming chapters, we will introduce a diverse array of co-design models, each representing a spectrum of systems, ranging from individual AVs to entire mobility ecosystems. Within these design scenarios, the level of detail we employ for the interfaces varies, and this variation is informed by the context in which the design problem arises.

Let's delve into the example of designing an AV to illustrate the significance of context in co-design. When tackling the challenge of co-designing a mobility system which utilizes a fleet of such vehicles, our primary focus might be on a few crucial aspects, such as the attainable speed of a single AV and its associated fixed and operational costs (Fig. 134). In this context, these are the wires that matter most, the essential components of the design that directly impact the system's overall performance and cost-effectiveness, from the point of view of a mobility solutions designer.

Now, let's pivot to a different context: co-designing the autonomy stack of the AV itself. In this scenario, our considerations become more intricate and multifaceted. Beyond merely considering speed and cost, we now need to go into the nitty-gritty details. We are concerned with precisely defining the tasks, such as the AV's ability to follow a specific trajectory at a particular speed, navigating through curves with a particular curvature. We are also attentive to specifics of the environment, such as the distribution of dynamic obstacles within the scene. Additionally, we expand our resource palette to include parameters like power consumption, emissions produced, and metrics related to safety and passenger comfort.

The overarching lesson here is that the level of detail and the selection of functionalities/resources and connections in a co-design problem are influenced by the context within which the problem is framed.

Example: lane control

Going back to the example of urban driving, we can show explicitly the production of the skeleton for the case of lateral control.

Recall that we had identified a co-design diagram as in Fig. 136, including lane control, feature extraction, their implementation, and the selection of lane cameras. As you have noticed, we were able to identify the logical dependencies between these components without actually specifying the details of their actual

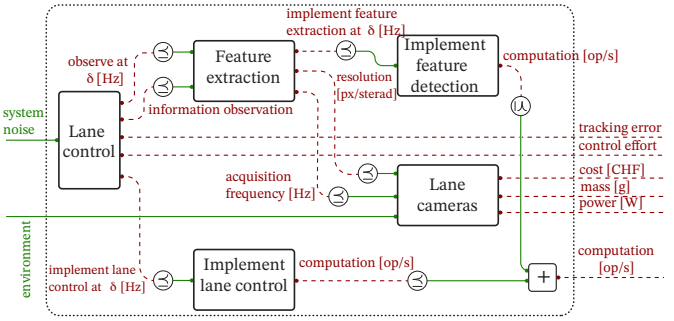


Figure 136: We consider lateral control as design problem, involving the design of control strategies and feature detection algorithms, together with sensor selection. R Resources are cost, mass, power, computation, control effort and tracking error.

models. Once we have this skeleton, we can start thinking about populating the design problems (feasibility relations).

9.2 Populating the models

Once the skeleton has been created, one can populate the models. To do so, one can use catalogues, analytical relationships, and data-driven methods. Note that the process of populating the models can be dynamic. More on that in Part C.

We now provide an exemplary population scheme for the models of lateral control, starting from the diagram in Fig. 136

Example: lateral control

Lane control Given the task of the AV to align itself with the lane, we define the vehicle configuration as $\boldsymbol{x}_t = \langle \theta_t, y_t \rangle$, where θ_t is the heading of the AV and y_t is its relative lateral position with respect to the center of the lane. The desired lane-aligned configuration at time t is denoted by $\boldsymbol{x}_t^{\mathrm{g}} = \langle \theta_t^{\mathrm{g}}, y_t^{\mathrm{g}} \rangle$ and the control input is the steering torque τ_t. We assume that from the sensor observations we can have Gaussian observations of the state $\boldsymbol{o}_t = \boldsymbol{x}_t - \boldsymbol{x}_t^{\mathrm{g}} + \boldsymbol{v}_t$, where $\boldsymbol{v}_t$ is white Gaussian noise with covariance $\boldsymbol{\Sigma}_{\boldsymbol{v}}$. As we show in [134], this problem can be solved with LQG control, by choosing to minimize the objective

$$\lim_{T\to\infty} \frac{1}{T} \mathbb{E}\{ \int_0^T \left((\boldsymbol{x}_t^{\mathsf{T}} \alpha \mathbf{Q}_0 \boldsymbol{x}_t) + (r_0 \tau_t^2 / \alpha) \right) \mathrm{d}t \}.$$

We formalized this as the *lane control* design problem, for which the functionality is the ability of the control to handle a given system noise and the resources are the needs to obtain the observations at a certain frequency with given precision and to implement the optimal control law at a specific frequency, requiring control effort $J_{\text{eff}} = \lim_{t\to\infty} \mathbb{E}\{r_0 \tau_t^2\}$ and tracking error $J_{\text{track}} = \lim_{t\to\infty} \mathbb{E}\{\boldsymbol{x}_t^{\mathsf{T}} \mathbf{Q}_0 \boldsymbol{x}_t\}$. In this case, the implementations the designer can choose are the different cost weights, parametrized by α. We show in [134] how the nature of the problem

allows one to obtain the optimal solutions for the design problem by solving specific Riccati equations. Hence, the model can be populated in a data-driven fashion.

Lane cameras To populate the model for lane cameras, we need to define a relation between the accuracy of the sensing and the physical sensors. Measurements are provided at a given frequency and with a specific resolution (in px/sterad) by *lane cameras*, which have a cost, mass and power consumption. Such data is obtained straight from camera catalogues.

Feature detection A feature detection algorithm processes the measurements providing the lane control design problem with observations at a certain frequency and with a certain precision. Obtaining the model for this is the realm of photogrammetry. In practice, we need to answer questions such as "what resolution (in px/sterad) is needed to achieve a certain feature detection accuracy?".

Algorithms implementation Finally, it is necessary to choose the implementation of the actual feature detection and lane control algorithms. For each of these we have a design problem, characterized by a catalogue of algorithms, each requiring different computation. To obtain the model, one needs to perform a benchmarking exercise. An excellent example of how to create benchmarking catalogues for algorithms, going as deep as to also search over the compiler flags, is given by *SLAMBench* [147] and the successive papers by Nardi and collaborators. For perception problems which cannot be adequately modeled by analytical photogrammetry relations, it also makes sense to not only vary implementation details (e.g., compiler flags) but also algorithm parameters. In that case, benchmarking would include the scope of the last two blocks together.

9.3 Expressivity and properties of the framework

When populating the models, it is important to consider some properties of the presented framework, listed below.

Uncertainty In co-design, there are instances where we grapple with uncertainties sorrounding the feasibility of specific components. Take, for instance, the design of a battery. A particular technology may promise a certain level of power performance while guaranteeing a range of life cycles, say, between 500 and 700 cycles. The question then arises: how do we characterize and handle this kind of uncertainty effectively?

To address this challenge, Censi developed an extension to the co-design theory which takes into account the introduction of tolerances and upper and lower

bounds for design problems, ordering them [152]. Interestingly, one can then interconnect the "uncertain versions" of the design problems, and the uncertainty propagates at the level of the interconnected co-design problem.

When querying an uncertain co-design problem, instead of receiving a single Pareto front of solutions, we obtain two distinct fronts: a lower bound and an upper bound. For instance, when designing an autonomous drone performing search-and-rescue missions, one solution might indicate that the drone, meeting all task specifications, will cost between 1,000 and 2,000 CHF. Another solution could reveal that, in the best-case scenario, the cost will be 1,000 CHF, but in the worst-case scenario, the design might not be feasible at all.

Dealing with parametric uncertainty is just one of the features of this extension. Other interesting ones include relaxing certain relations (via lower and upper bounds) for a reduced number of solution iterations, and relaxing relations with infinite cardinality (which need to be relaxed in this framework). From the categorical point of view, this structure defines a new category of uncertain design problems, leveraging the twisted arrow construction [123].

Temporality Co-design is often perceived as a static exercise, but in reality, many complex systems are dynamic and undergo changes over time. The timeframes involved in such changes can vary significantly, spanning from hours and days to months and even years, depending on the particular system. This aspect of systems engineering, known as *temporality* [18], plays a crucial role in our framework, and it can manifest in various forms.

In the simplest case, temporality entails considering how different components within the system might evolve in the future. It involves making predictions or assumptions about their behaviors and examining the resulting system-level impacts. For instance, a company focused on developing sensors of AVs might pose the question "If we can manufacture a sensor with specific sensing capabilities at a particular cost and power consumption level, how will it affect the state of the art in sensor technology? In which applications will it excel?"

Taking a more advanced perspective, one could try to define dynamic design problems. Although this area is a work in progress, we can provide an intuitive understanding. In dynamic design problems, the relationships governing feasibility become dynamic, and the availability of resources and functionalities evolves with time. Queries in this context revolve around the temporal aspects of functionalities and resources. Efficiently addressing such queries requires an analysis of the dynamic systems at play. More on that in Chapter 14.

Decentralization So far, we have presented co-design problems for which we have specified all the models personally. Nevertheless, as soon as the system

interfaces are established, the process of populating the feasibility relations can become a *decentralized* effort. This introduces various levels of collaboration, aligning with one of the key desiderata for our framework. Furthermore, it opens the door to the active involvement of humans in the design process.

To illustrate one of the collaborative modes, consider the co-design of the autonomy stack for a robotic system. In this case, different teams, such as those responsible for perception, planning and control, mapping, and liability considerations, can each contribute their expertise to address specific co-design problems tailored to their domain. This collaborative approach enables cross-functional teams to collectively shape the design of the system, ensuring that various aspects harmoniously integrate with one another.

This particular facet of co-design gives rise to several intriguing challenges and opportunities for further investigation. We will explore some of these ideas in Chapter 13.

9.4 Developer vs. user viewpoints

At this point, it is clear that *developer* and *user* viewpoints are different. From the point of view of the user, one only has to:

- ▹ Identify the design problems (i.e., the "blocks"), which means:
  - Identify the posets at play;
  - Assign them to functionalities or resources;
- ▹ Identify the co-design diagram (i.e., the interconnection of various design problems);
- ▹ Actually populate the feasibility relations of every design problem, while making sure that they respect the monotonicity assumption;
- ▹ Finally, solve specific queries (i.e., FixFunMinRes or FixResMaxFun).

Thanks to the formal framework we have presented, interconnecting different problems in the various ways introduced can be thought of as a mere "graphical" exercise (all the formal things hold in the background).

Starting from the steps outlined above, it is clear that we need two kind of tools to use the framework in practice: a *modeling language* to express the specifications for the first three points, and a *solver/interpreter* to solve the problem, once specified.

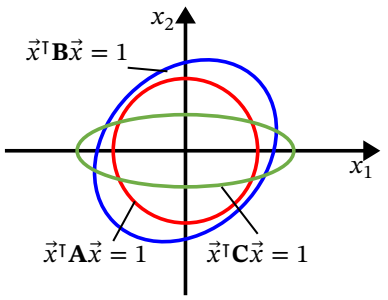
Language Censi developed a *modeling language* called MCDPL[17], which can be used to specify co-design problems, describing posets and systems of relations between them. Whenever co-design assumptions are violated, syntax errors are

[17] See more at co-design.science/software/

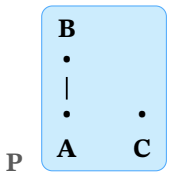
triggered.

For instance, one can create a co-design problem focusing on the actuation of a drone:

```
mcdp  {
    provides endurance [s]
    provides payload [kg]
    battery = instance template
         mcdp  {
            provides capacity [J]
            requires mass [kg]
        }
    actuation = instance template
         mcdp  {
            provides lift [N]
            requires power [W]
        }
    capacity provided by battery >= endurance * (power required by actuation)
    g = 9.81 m/s^2
    lift provided by actuation >= (mass required by battery + payload) * g
}
```



(a) Ellipses representing positive definite matrices.



(b) Order between positive definite matrices.

Figure 137: Poset of positive (semi-) definite matrices.

The language automatically produced the co-design problem reported in Fig. 138.

Note two aspects. First, all the posets employed in this examples are standard, and pre-built in MCDPL (i.e., $\overline{\mathbb{R}}_{\geq 0}$ associated to particular units). One can very easily define custom posets. For instance, designing a poset for covariance matrices (e.g., borrowing a previous example in Fig. 137), one can create a file `cov.mcdp_poset` as follows.

```
poset {
   A B C
   A <= B
}
```

Second, note that in the drone example all the feasibility relations were explicitly specified in analytic form. One can also borrow catalogues. For instance, for lateral control one could specify the design problem as follows:

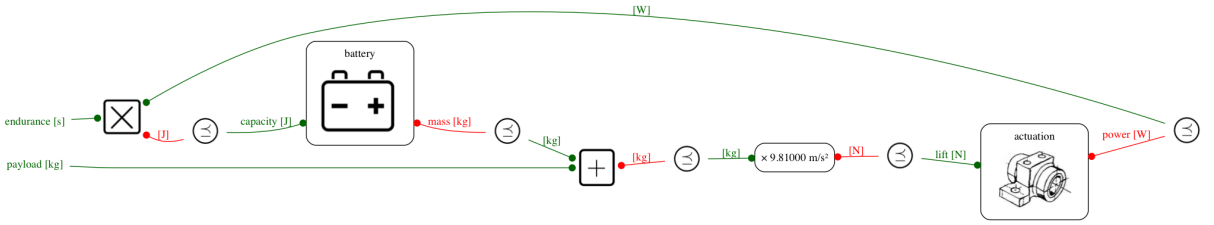


Figure 138: Design problem of the actuation of a drone, produced by MCDPL.

```
dp {
    provides system_noise [`cov]
    requires tracking_error [dimensionless]
    requires observation_noise [`cov]
    requires implement_lane_keeping_at_delta [Hz]
    requires control_effort [dimensionless]
    requires observe_at_delta [Hz]
    implemented-by code
     ↪ mcdp_importers.read_yaml(fn="catalogue_lateral_control.dpc.yaml")
}
```

Here, the last line is a reference to a piece of code which imports specific implementations from a YAML file, which could be generated out of simulations, or any other process.

Solution Once defined, co-design problems can be queried, via a dedicated solver. The solver can interpret the co-design models, and, at will, report antichain of solutions and lists of related implementations.

Back to the desiderata Going back to the desiderata, the presented interface makes the framework user-friendly, intellectually tractable, and easy to manipulate. Furthermore, it allows multiple designers to work together in a collaborative environment, each specifying different design problems (e.g., different files), asynchronously, promoting temporality. In a sense, this contributes to the *democratization* of design processes. For instance, think of a challenge in which different teams have to work on models (e.g. physical components, algorithms) to optimally design a robot for a particular set of tasks. Here, each would be contributing to a number of models for design problems, and the solver would be able to constantly tell which components correspond to optimal solutions for the entire system design.

Co-design of autonomy 10

When I look out in the future, I can't imagine a world, 500 years from now, where we don't have robots everywhere.
—Rodney Brooks[18]

In this chapter we continue the exposition of the co-design problem for AVs in a urban driving context. In particular, we first propose a basic model, completing the ones presented in previous chapters (Section 10.1). Further, we present a modeling with more advanced autonomy models (Section 10.2). For both problems, we provide compelling case studies, showing the properties of the co-design framework.

10.1 Co-design of an autonomous vehicle

We recall the urban driving problem and its functional decomposition, introduced in Section 8.2, and now provide the entire co-design formulation of the problem. Then, we show interesting solutions by means of trade-offs of quantities of interest. This section is based on authored works [128], [134]. The two main functions involved in this co-design problem were *lateral* and *longitudinal* control, and we have provided a co-design formalization for lateral control in Section 8.3, as well as its actual implementation. We can therefore now proceed with the rest of the models.

Modeling longitudinal control

As highlighted above, lateral control can be modeled as a co-design problem using analytical solutions of optimal control problems. For the longitudinal control sub-task, however, we want to showcase the ability of the framework to handle cases in which co-design relations are not directly available in analytical form, to the point at which one has to rely on numerical simulation (Fig. 141).

The AV is required to brake in time in the presence of obstacles and to guarantee a desired cruise speed. The AV is characterized by a dynamic performance $\langle v_{\max}, a_{\max}, a_{\min} \rangle$, where $v_{\max}$ is the maximum vehicle's achievable speed and $a_{\max}, a_{\min}$ are its maximum acceleration and deceleration. These parameters depend on the chosen vehicle type (e.g., on the propulsion system). The vehicle's longitudinal dynamics are $\mathrm{d}x_t = v_t \mathrm{d}t$, $\mathrm{d}v_t = a_t \mathrm{d}t$, where a_t and v_t represent the vehicle's acceleration and velocity.

[18] Brooks is an Australian roboticist, former MIT faculty. Among others, he is a founder and former CTO of iRobot.

In practice, to perform longitudinal control, one needs to:

- ▹ Employ sensors to measure the environment (i.e., presence of obstacles);
- ▹ Feed the information about obstacles to a reactive brake controller;
- ▹ Actually implement the brake controller.

This allows one to identify the following key components.

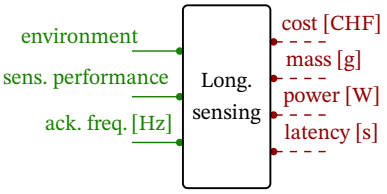


Figure 139: Design problem for longitudinal sensing.

Longitudinal sensing The AV is equipped with sensors, which provide obstacle detections along the road. It has already been observed that sensors can be ordered by their ability to discriminate states [127]. In our work, each sensor is characterized by its sensing performance, expressed as a tuple $\langle \mathsf{FP}(d), \mathsf{FN}(d), \mathrm{ACC}(d) \rangle$, where $\mathsf{FP}, \mathsf{FN} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{[0,1]}$ represent false positives (i.e., given an environment without obstacles, the probability of detecting one) and false negatives (i.e., assuming the presence of an obstacle, the probability of not detecting it) curves as a function of distance from the obstacle and with $\mathrm{ACC} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ denoting the sensing accuracy (range) as a function of distance from the obstacle. These curves can be obtained in different ways, as explained in Section 2.3. Note that to consider the curves as functionalities in the *longitudinal sensing* design problem, we compare them in posets using the point-wise order, and combine the posets by taking their product, resulting in the sensor performance poset. The detections also depend on the environment in which the task needs to be solved. This could include the time of the day as well as the density of obstacles on the road. Furthermore, a sensor provides measurements at a certain acquisition frequency and has specific latency (in s), cost, mass and power.

This gives rise to a design problem as the one in Fig. 139.

Obtaining the model: One obtains sensor specifications from catalogues and detection properties from benchmarking routines.

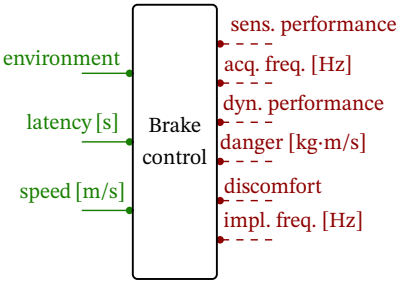


Figure 140: Design problem for longitudinal sensing.

Brake control Based on the generative model of the measurements, we produce a Bayesian estimate of the probability of having a pedestrian at each distance d and denote it by $f(d)$. Because it is not possible to derive a closed form solution for this POMDP, we choose the following parametrized control law. Starting from the current speed of the vehicle v_t and the Bayesian estimate $f(x)$, we want to compute the control input $a_t \in \{a_{\min}, a_{\max}, 0\}$. The critical braking distance is given by $d_{\mathrm{crit},t} = v_t^2/(2 \cdot a_{\min})$. By fixing a certain threshold Θ, if $\int_0^{d_{\mathrm{crit},t}} f(x)\mathrm{d}x \geq \Theta$, then $a_t = a_{\min}$. Else, in case we are slower than the desired speed (i.e., $v_t < v_{cruise}$), $a_t = a_{\max}$. If none of these cases applies, then one chooses $a_t = 0$.

Doing so, we can write the *brake control* problem as a design problem in which functionalities are the provided cruise speed (i.e., the performance, in km/h) and the handled sensing latency and environment. The resources are the sensing performance,

the sensing acquisition frequency, the computation power needed to execute the control law and the dynamic performance of the vehicle. Furthermore, we measure the performance of the longitudinal control action by means of discomfort and danger, defined as follows. Given a time horizon T, discomfort is expressed as $\mathsf{dis} = \int_0^T |a_t| \mathrm{d}t$ and penalizes changes in acceleration [167]. Danger captures both the *probability* and the *impact* of failures and is expressed as the product of the probability of hitting an obstacle and the momentum of the collision (in kg·m/s). This gives rise to a design problem as the one in Fig. 140 (the monotonicity can be assessed empirically).

Obtaining the model: Given the sensing model and the controller parameters, the *brake control* design problem can be modeled by running numerical simulations. This approach has two fundamental advantages. First, the simulations can be run in parallel, as they do not depend on each other. Second, the general co-design optimization can be run with incomplete simulation results, obtaining reduced levels of accuracy for the co-design solutions. It can actually be shown that the accuracy of the solutions of the co-design problem is monotone with the number of simulations one has [152].

Implementation brake control Brake control needs to be actually *implemented* at a certain frequency, requiring computation.

Obtaining the model: A catalogue of different algorithms (which may differ in nature, or just in the choice of some parameters).

Entire longitudinal control diagram By interconnecting the aforementioned blocks, one obtains the co-design diagram for the longitudinal control of an AV (Fig. 141). In particular, the brake control design problem requires sensing information from the longitudinal sensing design problem, and also requires to be implemented. The general *longitudinal control* co-design diagram is subject to the environment and desired cruise speed and requires cost, mass, power, dynamic performance, danger, discomfort and computation.

Composing the full diagram

In the previous sections we detailed the modeling of the *lateral control* and *longitudinal control* design problems. Following the principle of functional decomposition, we can now interconnect these two components with the rest of the system, obtaining the general co-design diagram reported in Fig. 144. In particular, in addition to lateral and longitudinal control, we actually need:

- A vehicle to be automated, providing power to the whole system;
- A computing unit providing computation to the system;

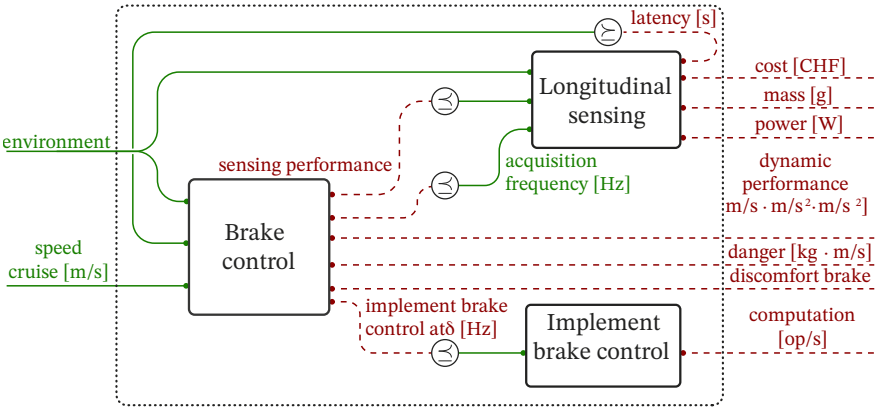


Figure 141: The longitudinal control design problem consists of a brake control, a longitudinal sensing and an implementation block. It provides the AV with the ability of reaching a cruise speed in a given environment, requiring cost, mass, power, dynamic performance, danger, discomfort and computation.

▹ A model for the discomfort of passengers;

This observation allows one to identify three other components, which we detail in the following.

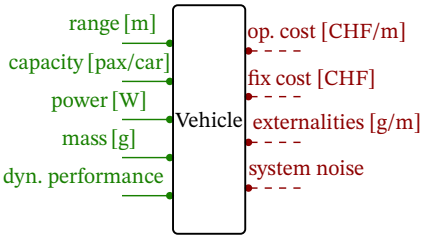


Figure 142: Design problem for the vehicle.

Vehicle We start from a vehicle to be automated, which represents the mechanical part of the system. This can be formulated as a design problem which provides certain dynamic performance and range (in m) and has a certain capacity (in pax/car), requiring both operational and fix costs and energy externalities (e.g., emissions, in g/km). Specifically, the vehicle has the ability to provide power which allows *computing*, *longitudinal control* and *lateral control* to happen, and to carry extra mass, arising from the sum of the mass of the sensors and computing unit. Both increased mass and power reduce the vehicle range. Each vehicle is characterized by a system noise, which is fed into the *lateral control* block. This gives rise to a design problem as in Fig. 142.

Obtaining the model: This model can be extracted from catalogues (e.g., by switching propulsion systems and chassis).

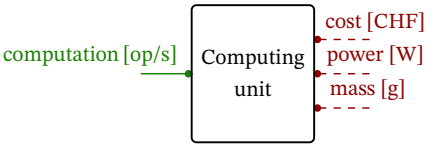


Figure 143: Design problem for computing unit.

Computing unit The computing unit needs to provide computation required by all the processes we have presented, and requires power and has a mass and a monetary cost. This gives rise to a design problem as the one in Fig. 143.

Obtaining the model The *computing unit* can be modeled through computer catalogues.

Discomfort Finally, a *discomfort* design problem joins the discomfort metrics arising from the *longitudinal control* and from the *lateral control* (in terms of control effort) to produce the total discomfort performance metric.

Obtaining the model: One can combine discomfort metrics arising from *lateral* and *longitudinal* controls using different assumptions, generating particular dis-

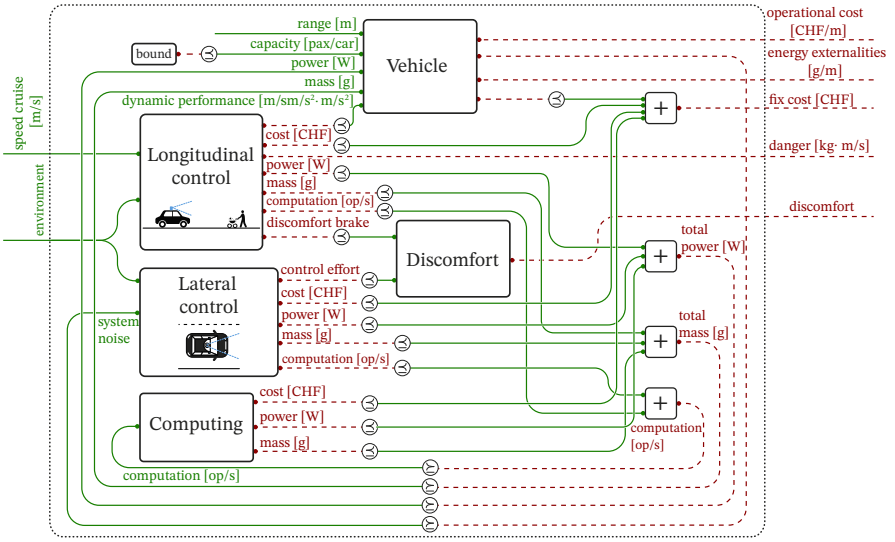


Figure 144: Co-design diagram for the design of an AV which needs to drive safely in a given environment, at a given cruise speed, and following a lane, without hitting obstacles. We choose costs, externalities, discomfort and danger as resources to minimize.

comfort models.

Full diagram The interconnection in a full diagram is reported in Fig. 144 and is quite intuitive. The dynamic properties of the vehicle required by the longitudinal control design problem are provided by the vehicle one, the computing required by all the processes at play are provided by the computing unit, and (total) power and mass are provided by the vehicle. Furthermore, the discomfort design problem merges the discomfort metrics for longitudinal and lateral control.

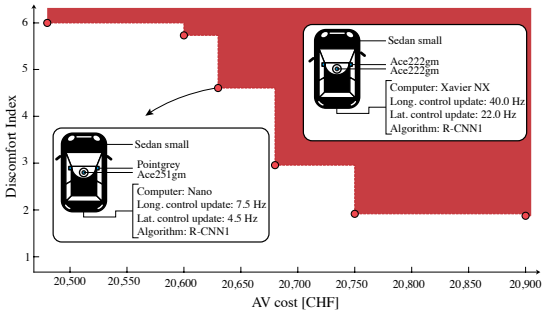
Co-design results

We now showcase the abilities of the proposed framework to solve the co-design problem of an AV. By considering the design problem proposed in Fig. 144, we optimize the design of an AV by means of cost, danger, discomfort, emissions, robustness to environment and cruise speed. In particular, the design options, listed in Table 7, include the selection of longitudinal and lateral sensors, control

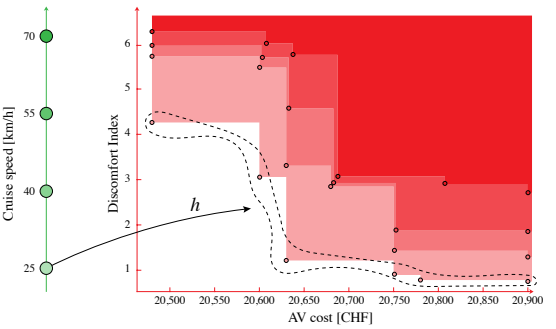
Variable	**Options**	**Source**
Vehicle	Sedan Small, Large, BEV, SUV, Minivan	[168]
Computers	Xavier, AGX, Nano, XavierNX	[153]
Control update	Brake: 1.0-50.0 Hz, Lane: 0.1-100.0 Hz	-
Sensors		
Lidars	Puck, HDL32E/64E	[169]
	OS032/64/128, OS232/64/128	[170]
Cameras	Ace251gm/222gm/13gm/5gm/15um	[154]
	Flir Pointgrey	[155]
Algorithms	Cameras: R-CNN1, R-CNN3	[171], [172]
	Lidars: KDE, STM-RBNN, STM-KDE	[173]

Table 6: Variables, options and sources for the AV co-design problem.

parameters, vehicles, computers and detection algorithms.



(a) Trade-off (antichain) of cost and discomfort in the design of an AV able to drive during the day at 55.0 km/h, with corresponding design choices.



(b) Monotonicity of the design problem: Higher cruise speed requires higher cost and discomfort.

Figure 145: Trade-offs of cruise speed, cost and discomfort in the design of an AV. (a) The figure shows the antichain of optimal design solutions. The red dots represent the optimal design solutions and the colored area represents the upper sets of resources for which the cruise speed of 55.0 km/h is feasible. One can see selected highlighted implementations corresponding to specific points in the antichain. (b) Pareto fronts of resources (expressed in terms of cost and discomfort) as a function of the provided cruise speed. Monotonicity is expressed via inclusion of the drawn upper sets.

We assume obstacles to be distributed following a spatial Poisson process and generate sensor performance curves for the 16 sensors listed in Table 7 using the results from [171], [173]. The authors of [171] compare the pedestrian camera detection performance during day- and night-time at two different operating points of a Faster R-CNN object detection algorithm. The performance in terms of recall and precision is investigated against object distances and object heights in px. We leveraged these results to interpolate the performance of the object detection with different sensors, using sensor resolution and angle of view [174]. For the camera accuracy we used the stereo depth error for a fixed baseline, assuming monocular camera depth estimation [175], [176]. Lidar sensor performance curves are generated using the results of [173], who show that the 3D pedestrian detection efficiency decreases as the distance between lidar and objects increases. This effect can be explained by the decrease in number of reflected points available per obstacle as the distance increases. Recall and precision performances are presented for different distances for three different algorithms (KDE-based, STM-RBNN and STM-KDE). The authors trained and validated their models on the KITTI dataset [177], originating from measurements of a Velodyne HDL-64 lidar. We estimated the measurement points per object at different distances and used them to interpolate for other lidars, generating curves similar to the ones presented in [172]. Finally, we assumed a constant accuracy for the lidars, extracting it from sensor catalogues. Note that the framework can accommodate arbitrary sensor curves, enabling us to test non-existent sensor performances as well. Given this setting, we simulated both the longitudinal and lateral control of the AV, creating numerical catalogues for the design problem. As previously explained, these simulations can be run in parallel, and build a database on which the solver

can operate.

Solutions which guarantee a certain speed (Fig. 145a) We assume an obstacle density of 5.0 obstacles/km during the day and query the optimal design solutions which enable the AV to reach 55.0 km/h (Fig. 145a). The dashed line represents the antichain of optimal solutions for the co-design problem, consisting of cost and discomfort. These solutions are not comparable, meaning that there is no instance which yields simultaneously lower discomfort and cost. In solid red we represent the upper set of resources. Attached, one can find selected design implementations, corresponding to specific optimal solutions. In general, as the budget for the AV increases, one is able to reduce the discomfort. For instance, with a cost of of 20,630 CHF one can obtain a discomfort index of 4.49, buying a small Sedan, equipping it with Pointgrey lateral cameras and an Ace251gm longitudinal camera paired with the R-CNN1 detection algorithm, together with a Nano computer. The vehicle is controlled at 7.5 Hz longitudinally and at 4.5 Hz laterally. Notably, investing only 150 CHF more per AV improves the discomfort by 250 %, requiring a Xavier NX computing unit, which controls the AV at 40.0 Hz and 22 Hz longitudinally and laterally, respectively, using measurements of two Ace222gm cameras.

Monotonicity of the AV design problem (Fig. 145b) We consider increasing cruise speeds and analyze the evolution in trade-offs in cost and discomfort. As can be gathered from Fig. 145b, we query the co-design solver for multiple performances, given arbitrary environments (here 5.0 obstacles/km during the day). Specifically, for each functionality, we compute the map h which maps to the minimum antichain of resources which provide it. Note that by increasing the desired cruise speed, one increases the required resources, as can be observed from the dominating upper sets in increasing red tonality. Given such trade-offs and the discrete nature of our tool, one can reason about the design problem by distinguishing the desired objectives.

10.2 Co-design of an autonomous vehicle 2.0

In this section, based on the authored paper [178] and work-in-progress extensions, we report another case study on the co-design of autonomous vehicle, employing more granular autonomy models. In particular, we focus on more detailed control models, extending the functional decomposition of the problem, by including speed tracking (Fig. 146). Furthermore, we extend existing models to consider state-of-the-art dynamic models and control techniques. We start by describing the updated model of the vehicle.

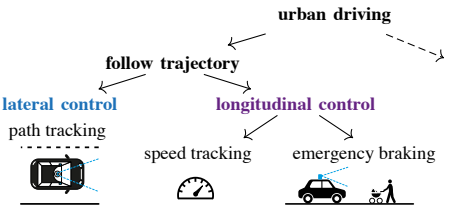


Figure 146: New functional decomposition for the task of urban driving.

Vehicle model

We consider the kinematic single-track model from [179], [180] and extend it by considering model uncertainty. Consider $p_r = [x_\mathrm{r}, y_\mathrm{r}]$ and $p_\mathrm{f} = [x_\mathrm{f}, y_\mathrm{f}]$ as the positions of the rear and front wheel with respect to the inertial coordinate frame. The heading θ describes the vehicle's orientation (the angle between $p_\mathrm{f} - p_r$ and the inertial frame). The steering angle δ describes the front wheel orientation with respect to the vehicle's one. Finally l is the distance between front and rear axles. We can write the no-slip condition for the wheels with the following non-holonomic constraints:

$$\begin{aligned} -\dot{x}_r \sin(\theta) + \dot{y}_r \cos(\theta) = 0, \\ -\dot{x}_f \sin(\theta + \delta) + \dot{y}_f \cos(\theta + \delta) = 0. \end{aligned}$$

We compute the rear velocity as

$$v_\mathrm{r} = \dot{p}_r (p_\mathrm{f} - p_r)/\|p_\mathrm{f} - p_r\|.$$

By considering the state space $\boldsymbol{s} = [x_\mathrm{r}, y_\mathrm{r}, \theta, \delta, v_\mathrm{r}]$, and control inputs $\boldsymbol{u} = [v_\mathrm{s}, a_\mathrm{r}]$, the dynamics read

$$\dot{\boldsymbol{s}} = f(\boldsymbol{s}, \boldsymbol{u}) + \boldsymbol{w} = \begin{bmatrix} v_\mathrm{r} \cos(\theta) \\ v_\mathrm{r} \sin(\theta) \\ v_\mathrm{r} \tan(\delta)/l \\ v_\mathrm{s} \\ a_\mathrm{r} \end{bmatrix} + \boldsymbol{w}, \tag{61}$$

where $v_\mathrm{s} \in [\dot{\delta}_\mathrm{min}, \dot{\delta}_\mathrm{max}]$ and $a_\mathrm{r} \in [\dot{v}_\mathrm{r,min}, \dot{v}_\mathrm{r,max}]$ are control inputs, $\delta \in [\delta_\mathrm{min}, \delta_\mathrm{max}] \subset [-\pi/2, \pi/2]$, $v_\mathrm{r} \in [v_\mathrm{r,min}, v_\mathrm{r,max}]$, and $\boldsymbol{w}$ is a standard Brownian process with effective noise covariance $\mathbf{W}$. The motion of the front wheel is then:

$$\begin{aligned} \dot{x}_\mathrm{f} &= v_\mathrm{f} \cos(\theta + \delta), \\ \dot{y}_\mathrm{f} &= v_\mathrm{f} \sin(\theta + \delta), \\ v_\mathrm{f} &= v_\mathrm{r} / \cos(\delta). \end{aligned}$$

Furthermore, the angular velocity is $\omega = v_\mathrm{f} \sin(\delta)/l$.

Measurement model We consider the discrete-time measurement model

$$\boldsymbol{y}_k = \gamma_k(\boldsymbol{s}_k + \boldsymbol{v}_k), \tag{62}$$

where $\gamma_k \in \{0, 1\}$ represents an intermittent observations process (e.g., linear Gaussian Bernoulli, linear Gaussian Markov, and linear Gaussian semi-markov [140]), and $\boldsymbol{v}_k$ is a standard Brownian process with effective noise covariance $\mathbf{V}$ (parametriz-

ing the fidelity of the measurement, i.e., the quality of the sensor).

State estimation

Following the literature, we consider an extended Kalman filter (EKF) with the discrete-time measurement model in (62). We summarize the estimation procedure.
Initialization:

$$\hat{\boldsymbol{s}}(t_0) = \mathbb{E}\left[\boldsymbol{s}(t_0)\right], \ \mathbf{P}_{0|0} = \mathbb{E}[(\boldsymbol{s}(t_0) - \hat{\boldsymbol{s}}(t_0))(\boldsymbol{s}(t_0) - \hat{\boldsymbol{s}}(t_0))^\intercal].$$

Prediction update: Solve

$$\begin{aligned}
\dot{\hat{\boldsymbol{s}}}(t) &= f(\hat{\boldsymbol{s}}(t), \boldsymbol{u}(t)), \\
\dot{\mathbf{P}}(t) &= \mathbf{F}(t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}(t)^\intercal + \mathbf{W}(t), \ \mathbf{F}(t) = \frac{\partial f}{\partial \boldsymbol{s}}\Big|_{\hat{\boldsymbol{s}}(t), \boldsymbol{u}(t)},
\end{aligned}$$

with $\hat{\boldsymbol{s}}(t_{k-1}) = \hat{\boldsymbol{s}}_{k-1|k-1}$, and $\mathbf{P}(t_{k-1}) = \mathbf{P}_{k-1|k-1}$, to obtain $\hat{\boldsymbol{s}}_{k|k-1} = \hat{\boldsymbol{s}}(t_k)$ and $\mathbf{P}_{k|k-1} = \mathbf{P}(t_k)$.

Measurement update: Compute

$$\begin{aligned}
\mathbf{K}_k &= \mathbf{P}_{k|k-1}(\mathbf{P}_{k|k-1} + \mathbf{V})^{-1}, \\
\hat{\boldsymbol{s}}_{k|k} &= \hat{\boldsymbol{s}}_{k|k-1} + \mathbf{K}_k(\boldsymbol{y}_k - \hat{\boldsymbol{s}}_{k|k-1}), \\
\mathbf{P}_{k|k} &= (\mathbf{I} - \mathbf{K}_k)\mathbf{P}_{k|k-1},
\end{aligned}$$

where $\mathbf{P}$ represents the covariance estimate.

Given this model, one can prove the monotonicity of the covariance estimate with respect to the noise and measurement covariances, and to the probability of losing observations. These results will be important when building the control design problems.

Lemma 10.1. The sequence of covariance estimates $\mathbf{P}$ produced by the EKF is monotone in $\mathbf{V}$ and $\mathbf{W}$. In other words:

$$\langle \mathbf{V}, \mathbf{W} \rangle \preceq \langle \mathbf{V}', \mathbf{W}' \rangle \implies \mathbf{P}(\mathbf{V}, \mathbf{W}) \preceq \mathbf{P}(\mathbf{V}', \mathbf{W}').$$

See proof on page 244.

Lemma 10.2. The sequence of covariance estimates $\mathbf{P}$ is monotone in the probability of dropping observations.

See proof on page 244.

Control

We instantiate the functional decomposition approach for the self-driving task of an autonomous vehicle. In particular, one can decompose this task into lateral and longitudinal control (Fig. 146). Longitudinal control can be then split into speed tracking and emergency braking.

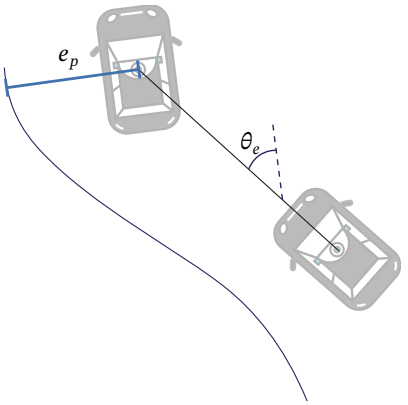


Figure 147: Stanley control.

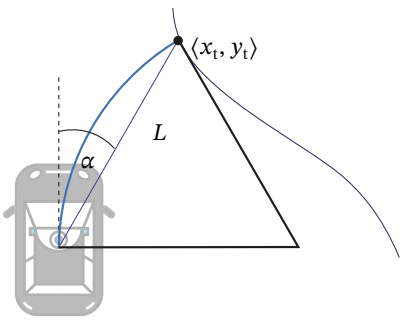


Figure 148: Pure pursuit control.

Lateral control The lateral control action can be formulated as choosing the steering velocity to track a given reference path. The vehicle's speed is typically considered constant at each time step (decoupling the longitudinal and lateral control problems). Controllers consider the first four states/equations of (61) (denoted by $\boldsymbol{z}$) and receive a state estimate at discrete times $\hat{\boldsymbol{z}}_k = \boldsymbol{z}_k + \boldsymbol{\mu}_k$, where $\boldsymbol{\mu}_k$ is a standard Brownian process with effective noise covariance $\mathbf{P}_k$ (as per state estimation procedure and measurement model). The control error $\boldsymbol{e}(t) = \begin{bmatrix} e_p(t) & \theta_e(t) \end{bmatrix}^\mathsf{T}$ is generally expressed as the distance between the vehicle's front axle and the reference point on the path, and the angle between the vehicle's heading and the tangent to the path at the reference point. In the following, we list the standard lateral control techniques we considered, and their properties [179], [181].

Stanley control: This is a geometric type of vehicle control based on the single-track bicycle model (i.e., the orientation and position of the front wheel with respect to the reference path are considered for generating control actions). Given the error, one can write the desired steering angle at any time as $\delta(t) = \theta_e(t) + \arctan(g e_p(t)/v_\mathrm{f})$, where g is the Stanley gain (Fig. 147). We denote by $e_{p,\mathrm{tot}}$ the total control positional error and by δ_tot the total control effort along a path.

Lemma 10.3. The total Stanley control lateral tracking error $e_{p,\mathrm{tot}}$ is monotonic in $\mathbf{W}$ and the sequence of estimate covariances $\mathbf{P}$.

See proof on page 244.

Lemma 10.4. The total Stanley control effort δ_tot is monotonic in $\mathbf{W}$ and the sequence of estimate covariances $\mathbf{P}$.

See proof on page 244.

Pure Pursuit: Given a reference path, the control law fits a semi-circle through the vehicle's current configuration to a point on the reference path, which has a distance (called "lookahead") L from the car (Fig. 148). We consider the algorithm presented in [182], and extend it by requiring the vehicle's heading to be tangent to the circle. The curvature of the semi-circle is $\kappa = 2\sin(\alpha)/L$. Given a constant rear velocity v_r, the angular velocity of a vehicle following the semi-circle is $\dot{\theta} = 2v_\mathrm{r}\sin(\alpha)/L$. From (61), one has $\delta = \arctan(2l\sin(\alpha)/L)$, where α is the angle between the vehicle's orientation and the vector from the current configuration $\langle x, y \rangle$ and the target one $\langle x_t, y_t \rangle$. Again, the control error $\boldsymbol{e}(t) = [e_p(t), \theta_e(t)]$

is expressed as the distance between the vehicle's front axle and the reference point on the path, and the angle between the vehicle's heading and the tangent to the path at the reference point. The control procedure is then: a) Find current location of the vehicle, b) find the path-point closest to the vehicle, c) find the goal point, d) transform it to the vehicle coordinates, e) calculate the curvature and set the steering angle accordingly, f) update vehicle's position [181].

Lemma 10.5. The total pure pursuit control lateral tracking error $e_{p,\text{tot}}$ is monotonic in $\mathbf{W}$ and in the sequence of estimate covariances $\mathbf{P}$.

See proof on page 244.

Lemma 10.6. The total pure pursuit control effort δ_{tot} is monotonic in $\mathbf{W}$ and in the sequence of estimate covariances $\mathbf{P}$.

See proof on page 244.

LQR with adaptive state space control: The error is given by $\boldsymbol{e}(t) = [e_p, \theta_e]$, where e_p is the positional error perpendicular to the path tangent, and θ_e is the difference between the path tangent and the vehicle orientation. The method linearizes the error dynamics around $[0, 0]$ at every time instant, and solves an infinite horizon optimization problem for the linearized system. The error dynamics for small errors can be formulated as

$$\dot{\boldsymbol{e}}(t) = \begin{bmatrix} 0 & v_\text{r} \\ 0 & 0 \end{bmatrix} \boldsymbol{e}(t) + \begin{bmatrix} 0 \\ v_\text{r}/l \end{bmatrix} \delta(t),$$

and the quadratic cost function to minimize takes the form

$$J(T) = \int_0^T \boldsymbol{e}(t)^\intercal \mathbf{Q} \boldsymbol{e}(t) + \delta^2(t) \mathbf{R} \mathrm{d}t.$$

Lemma 10.7. The total LQR control lateral tracking error $e_{p,\text{tot}}$ is monotonic in $\mathbf{W}$ and in the sequence of estimate covariances $\mathbf{P}$.

Lemma 10.8. The total LQR control effort δ_{tot} is monotonic in $\mathbf{W}$ and in the sequence of estimate covariances $\mathbf{P}$.

See proof on page 244.

Nonlinear Model Predictive Control (NMPC): The NMPC method with receding horizon strategy aims at minimizing the positional error $\boldsymbol{e}_k$ (expressed with respect to the point on the path which is closest to the vehicle's center of mass at instant k) and control effort δ over $n_\text{h} \in \mathbb{N}$ steps. The formulation of the

optimization problem is as follows:

$$\begin{aligned} u_k^\star &= \underset{U_k}{\operatorname{argmin}} \sum_{i=0}^{n_\mathrm{h}+1} e_{k+i}^\mathrm{T} \mathbf{Q} e_{k+i} + u_{k+i}^\mathrm{T} \mathbf{R} u_{k+i}, \\ U_k &= \{u_k, \dots, u_{k+n_\mathrm{h}-1}\}, \\ e_k &= \hat{e}(t_k), \\ e_{k+i} &= \int_{k+i-1}^{k+i} v_\mathrm{f} \sin(\theta_e(\tau) - u(\tau)) \mathrm{d}\tau, \end{aligned}$$

where $\dot{\theta}_e$ follows Section 16.2, and where one only applies $u_0^\star$ each time. This technique is characterized by different path approximation techniques (e.g., linear, quadratic, and cubic), integration techniques, and by different lateral error reference points on the vehicle (e.g., rear or center of gravity).

Lemma 10.9. The total NMPC lateral control tracking error is monotonic in $\mathbf{W}$ and the sequence of estimate covariances $\mathbf{P}$.

Lemma 10.10. The total NMPC lateral control effort is monotonic in $\mathbf{W}$ and the sequence of estimate covariances $\mathbf{P}$.

See proof on page 245.

Speed control The control goal is to track a certain target velocity v_t. From (61), the velocity dynamics are $\dot{v}_\mathrm{r} = a_\mathrm{r} + \boldsymbol{w}_{v_\mathrm{r}}$. The system receives an estimation of the current velocity through the measurement model at each time instant k: $\hat{v}_{\mathrm{r},k} = v_{\mathrm{r},k} + \rho_k$, where ρ_k is a standard Brownian process with effective noise covariance q_k (as per state estimation procedure and measurement model). The control input is typically formulated via a PID control scheme (i.e., just by choosing specific tuning parameters $k_\mathrm{p}, k_\mathrm{i}, k_\mathrm{d}$):

$$u(t) = k_\mathrm{p}(v_\mathrm{t} - \hat{v}(t)) + k_\mathrm{i} \int_0^t (v_\mathrm{t} - \hat{v}(\tau)) \mathrm{d}\tau - k_\mathrm{d} \frac{\partial \hat{v}}{\partial t}.$$

Lemma 10.11. The total PID control tracking error is monotonic in $\mathbf{W}$ and the sequence of estimate covariances q.

See proof on page 245.

Lemma 10.12. The total PID control effort is monotonic in $\mathbf{W}$ and the sequence of estimate covariances q.

See proof on page 245.

Brake control The topic of (emergency) braking has been treated in detail in [128], where longitudinal sensors (ordered by their performance, expressed via false positives, false negatives, and accuracy curves) were used to detect potential obstacles. Clearly, the more uncertain the obstacle detection, the more potentially dangerous will the braking maneuver be. We delay the treatment of this particular topic to future works, and refer the interested reader to [128].

Co-design diagrams for lateral control

To perform lateral control, one has to:

- Employ sensors to estimate the lateral positions (e.g., cameras);
- Design a lateral controller;
- Implement the control action.

Therefore, we identify the following design problems.

Lateral control Lateral control provides the fulfillment of a task (e.g., performing a maneuver at a certain speed, or with a certain curvature), in a specifc environment (e.g., characterized by the time of the day, or the density of obstacles on the road), and with robustness to particular uncertainty in the vehicle model. The vehicle is characterized by its dynamic performance (e.g., parametrized by the reachable speed, acceleration, and steering angle). For the control law to be implemented, observations (with particular precision) from sensors are required (to estimate position and heading), which are received at particular frequency. Control techniques will need to be implemented at certain frequencies and will cause specific control efforts and errors (expressed in terms of lateral deviation), as well as discomfort (e.g., intensity of the steering) and dangerous situations. This gives rise to a design problem as in Fig. 149.

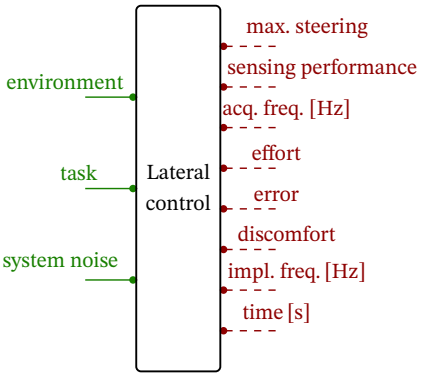


Figure 149: Design problem for lateral control.

Obtaining the model: Depending on the control technique, models can be obtained analytically and via numerical simulations.

Lateral sensing Lateral sensing provides observation at a certain acquisition frequency and with a certain sensing performance (e.g., via the poset of sensor-algorithms pairs). This comes at a monetary cost, and consumes power and computation (Fig. 150).

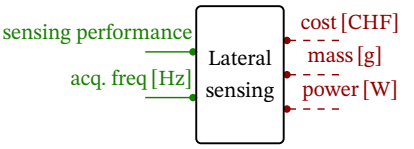


Figure 150: Design problem for lateral sensing.

Obtaining the model: Models are obtained by sensor catalogues, photogrammetry, and simulations (as explained for previous examples). Particular observation schemes can also be artificially perturbed, and observations dropping schemes can be applied [140].

Implementing algorithms Implementing algorithms can be formulated as a design problem, which have seen often in previous examples.

Co-design diagrams for longitudinal control

To perform longitudinal control, one has to:

- Employ sensors to estimate the longitudinal speed and the presence of obstacles;
- Design a longitudinal controller;
- Implement the control action.

Therefore, we identify the following design problems.

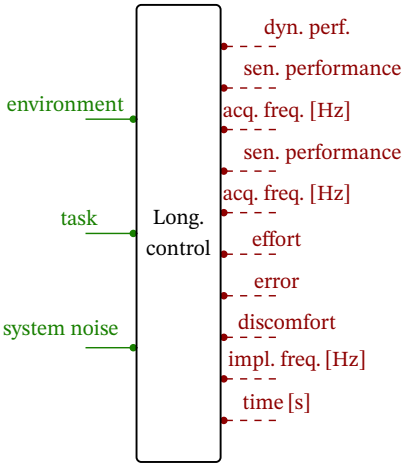


Figure 151: Design problem for longitudinal control.

Longitudinal control Longitudinal control provides the fulfillment of a task (e.g., performing a maneuver at a certain speed, or with a certain curvature), in a specifc environment (e.g., characterized by the time of the day, or the density of obstacles on the road), and with robustness to particular uncertainty in the vehicle model. The vehicle is again characterized by its dynamic performance. For the control law to be implemented, observations (with particular precision) from sensors are required (to estimate speed and presence of obstacles), which are received at particular frequency. Control techniques will need to be implemented at certain frequencies and will cause specific control efforts and errors (expressed in terms of velocity deviation), as well as discomfort (e.g., gravity of the accelerations) and dangerous situations (Fig. 151).

Obtaining the model: Depending on the control technique, models can be obtained analytically and via numerical simulations.

Longitudinal and speed sensing, as well as algorithm implementations, can be formulated as design problems, analogously to what we have shown for lateral control.

Interconnecting the full diagram

We now have defined both lateral and longitudinal control co-design problems, and we are ready to build the full diagram for the co-design of an autonomous vehicle (Fig. 152). As we did in Section 10.1, we introduce a computer, and a design problem to account for discomfort. Additionally, we now include design problems which account for control error and control effort, merging the contributions of the different control techniques. Furthermore, we introduce a design problem for the vehicle to be automated, which provides mass, dynamic performance, and power, and requires monetary cost, as well as system noise.

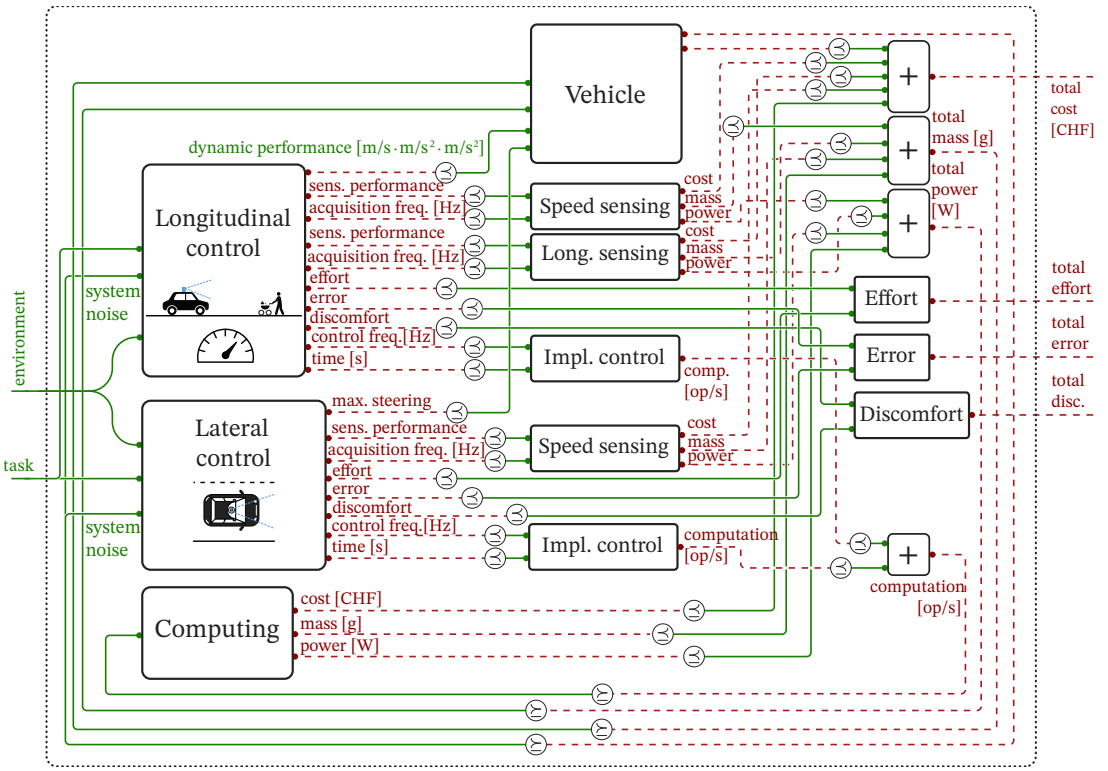


Figure 152: Co-design problem of an AV.

Once all of these design problems have been introduced, the interconnection is logical.

Co-Design results

We consider urban scenarios, extending and customizing the ones proposed in the CommonRoads framework [183]. We implemented the mentioned autonomy pipelines in our own simulator. While the proposed approach has been tested on several scenarios (e.g., racing, pursue-evasion, exploration), for exposition purposes we focus on two examples. We first look at the case in which an AV needs to perform a 90° degrees, and then look at a lane change example (Fig. 153, Fig. 154). The task of the AV consists in following a trajectory (with customized curvature severity) at a desired speed. By fixing a particular task we want to find the autonomy pipeline for the AV to minimize selected resource usages.

We now solve the co-design problem presented in the previous section focusing on a selection of queries. By fixing a task (i.e., a desired scenario, speed, and average curvature), we can characterize optimal design solutions in terms of monetary cost, control effort, control error, danger, and discomfort. The design space is characterized by the controllers presented above and their parameters,

Table 7: Variables and options for the AV co-design problem.

Variable	Options
PP	$L \in \{0.01, 0.05, 0.5, 1.0, 2.0\}$
Stanley	$g \in \{0.05, 0.1, 0.5, 1.0, 1.5, 2.0\}$
LQR	$\mathbf{R} \in \{0.001, 0.05, 0.5, 1.0, 10.0\}$, $\mathbf{Q} \in \{0.1 \cdot \mathbb{1}, 1.0 \cdot \mathbb{1}, 10.0 \cdot \mathbb{1}, 0.2 \cdot (\mathbb{1} - 0.9e_3), 1.0 \cdot (\mathbb{1} - 0.9e_3), 5.0 \cdot (\mathbb{1} - 0.9e_3)\}$
NMPC	$n_h \in \{10.0, 15.0, 20.0, 25.0\}$, $\mathbf{R} \in \{0.05, 0.5, 1.0, 5.0\}$, $\mathbf{Q} \in \{0.01 \cdot \mathbb{1}, 0.1 \cdot \mathbb{1}, 1.0 \cdot \mathbb{1}, 10.0 \cdot \mathbb{1}\}$
PID	$k_p \in \{0.1, 0.5, 1.0, 2.0\}$, $k_I \in \{0.01, 0.1, 0.5, 1.0\}$, $k_d \in \{0.01, 0.05, 0.1, 1, 0\}$
Computers	RPi 4B, Jetson Nano/TX1,2/AGX Xavier, Xavier NX
Sensors	Basler Ace251gm/222gm/13gm/7gm/5gm/15um, Flir Pointgrey, KistlerSMotion, OS032/128, OS232/128, HDL 32/64

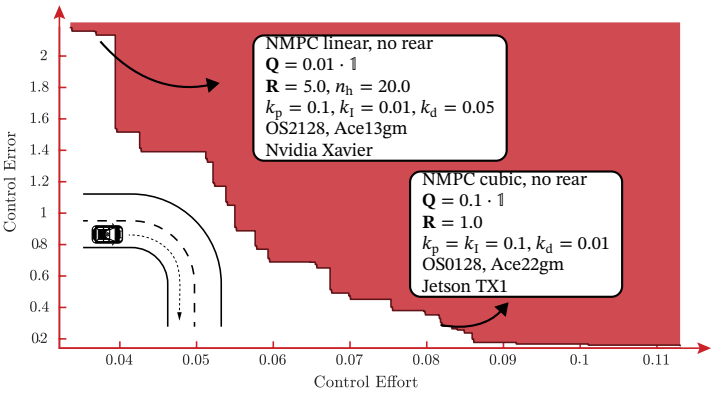


Figure 153: Trade-off (antichain) of total control error and effort for a 90° turn, with low curvature at 8 m/s, with corresponding design choices.

various sensors for the different perception blocks, and computer models, all listed for convenience in Table 7. For simplicity of exposition, we do not consider obstacle detection modeling (already treated in depth in Section 10.1), and focus on path tracking and speed control. Note that this represents just a sample of the designs we can look at. (For instance, we neglect control frequencies).

Control effort and control error trade-offs We first consider a case in which we want the vehicle to perform a 90° turn with a low curvature, at 8 m/s, with a standard battery electric vehicle. By solving the co-design problem, we obtain a Pareto front of optimal designs, which we can interpret by looking at its 2D-projections. We first look at the trade-offs between total control effort and total control error (Fig. 153). In red, the Pareto front of optimal solutions, which are not comparable since no instance leads simultaneously to lower control error and control effort. The upper set of feasible resources is given in solid red. Furthermore, for each point lying on the Pareto front, we are able to report details about the optimal designs, including the chosen control technique and its parameters, as well as considered sensors and computer. As one can see in Fig. 153, low control effort (discomfort) can be achieved with a specific combination of controllers and parameters, at the cost of an important control error. Similarly, low control error can be achieved by another design, with increased control effort.

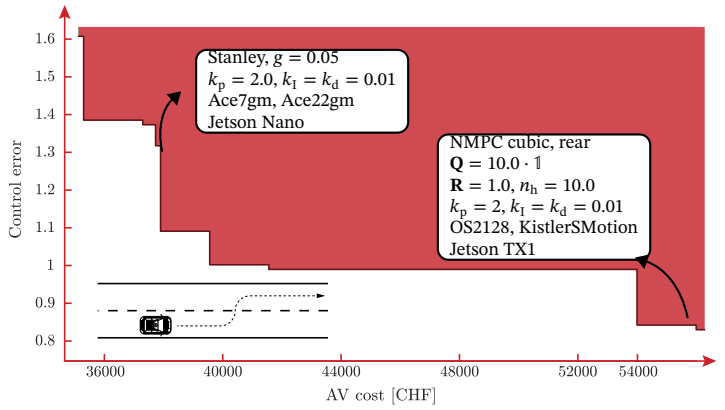


Figure 154: Trade-off (antichain) of cost and control error for lane change with high curvature at 15 m/s, with corresponding design choices.

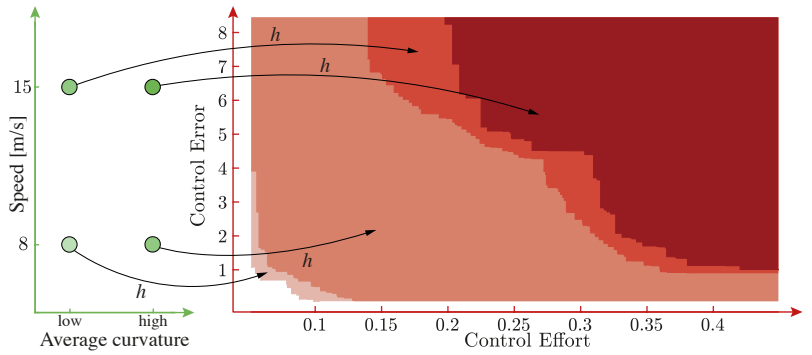


Figure 155: Monotonicity of the co-design problem: higher cruise speed or curvature will require higher control error and effort.

Monetary cost and control error trade-offs We look at the task of lane changing, choosing a high curvature and a speed of 15 m/s. We can now solve the co-design problem with the updated task. To showcase the richness of the insights we can produce, we now report the trade-offs between monetary costs and control error for the AV (Fig. 154). Clearly, to achieve lower control error one needs to pay more. Interestingly, investing 39,000 CHF instead of 54,000 CHF only deteriorates the error by 10%.

Monotonicity We consider the task of lane changing, now showcasing the monotonicity properties of the developed framework. We look at varying tasks, starting from a low speed of 8 m/s and low curvature and increasing speed to 15 m/s and high curvature. Fig. 155 shows multiple co-design queries. In particular, for each functionality (left plot), we report the Pareto front and the upper set of optimal resources (right plot). Monotonicity can be seen in the dominance of subsequent Pareto fronts (right plot), illustrated in increasing red tonality via inclusion of the upper sets.

Co-design of mobility systems | 11

We can't direct the wind but we can adjust the sails.
—Aristotle[19]

In this chapter, we climb the abstractions, and and model the co-design problem of a mobility system, where the design problem of an AV is only one of the many interconnected components at play. The chapter refers to authored papers [184]–[186].

First, we will motivate the example (Section 11.1). Second, we will present an intermodal mobility framework (Section 11.2), as well as the connected co-design framework for future mobility systems (Section 11.3). Finally, we will present extensive results, based on the real world case study of Washington D.C., USA (Section 11.4).

11.1 Motivation

While the motivation for co-design in the context of autonomy is clear, and has been covered in Chapter 1, the same ideas in the context of mobility systems need contextualization.

The present landscape of transportation systems is experiencing profound transformations driven by the implementation of innovative mobility solutions, such as AVs and micromobility (μM) systems. These emerging mobility paradigms hold the potential to substantially mitigate the adverse impacts traditionally associated with transportion systems, including emissions, travel time, parking space allocation, and, crucially, road fatalities (for a review on the subject, refer to [187]).

Nevertheless, practical experience in the industrial sector reveals a recurring challenge in the development of these new mobility solutions: the absence of well-defined, precise requirements in terms of the services they are intended to offer [188]. Yet, knowledge about their intended service (e.g., last-mile versus point-to-point travel) might dramatically impact how vehicles are designed and significantly ease their development process. For instance, if it were known that, for a particular city, an effective on-demand mobility system with AVs only necessitates operations at speeds of up to 30 mph on relatively simple roads, this knowledge would considerably simplify AV design and hasten their deployment. Furthermore, from a holistic transportation management perspective, insights into the trajectory of technological advancements in new mobility solutions would certainly affect decisions regarding future infrastructure investments and service

[19] Aristotle was an Ancient Greek philospher and polymath.

provisions. In other words, the design of future mobility solutions and the design of a mobility system leveraging them are intimately *intertwined*.

This calls for methods to reason about such a coupling, and in particular to *co-design* the invidual mobility solutions and the associated mobility systems. An essential requirement in this context is the ability to accomodate a spectrum of diverse objectives that are often challenging to directly compare (consider, for instance, travel time, public expenditure, and externalities). It also involves formulating hierarchical design challenges spanning various disciplines, and devising computationally tractable solutions.

Accordingly, the goal of this line of research is to lay the foundations for a framework through which one can systematically co-design future mobility systems. The framework is instantiated in the setting of co-designing intermodal mobility systems [189], whereby fleets of self-driving vehicles provide on-demand mobility jointly with fleets of micromobility vehicles (μMVs) such as e-scooters (ESs), shared bikes (SBs), mopeds (Ms) and fuel-cell mopeds (FCMs), and public transit. Aspects that are subject to co-design include fleet sizes, vehicle-specific characteristics for AVs and μMVs, and service features, such as public transit service frequency, prices, and serviced networks.

Related Literature This research stream lies at the intersection of two critical domains: the design of public transportation services and the creation of innovative mobility solutions.

The first research stream, reviewed in [190]–[192], primarily focuses on *strategic*, long-term infrastructure modifications and *operational* short-term scheduling within the realm of public transportation. Notably, joint design efforts targeting traffic network topology and control infrastructure have been explored in [193], [194]. Research into public transportation scheduling has yielded valuable insights, including optimization of passenger and operator costs in [195], satisfaction of demand in [196], and energy consumption management of the system in [197]. However, these investigations often concentrate on individual infrastructures, such as the road network or public transportation, and lack consideration for their integrated design alongside emerging mobility solutions.

The research on novel mobility solutions mainly pertains to AVs, AMoD systems, and μM. The research on design of AMoD systems is thoroughly reviewed in [187] and references therein, and mainly concerns their fleet sizing. In this regard, existing studies range from simulation-based approaches [198]–[203] to analytical methods [204]. For instane, in [205], the fleet size and the charging infrastructure of an AMoD system are jointly designed, and the arising design problem is formulated as a mixed integer linear program. In [206], the fleet sizing problem is solved together with the vehicle allocation problem. Furthermore, [207] presents a framework for the integrated design of AMoD fleet size and composition. More

recently, the joint design of multi-modal transit networks and AMoD systems was formulated in [208] as a bilevel optimization problem and solved with heuristics, and coupled with infrastructure design in [209], using multi-objective linear optimization. Overall, the problem-specific structure of existing design methods for AMoD systems is often not amenable to a modular and compositional problem formulation. Furthermore, key AV characteristics, such as the achievable speed, are not considered. Research on the design and impact of μM solutions has been reviewed in [210], with a particular focus on the urban deployment of SBs and ESs. In particular, [211] presents a comprehensive design framework for a multi-modal public transportation system, including various μM solutions and buses, optimizing user preferences and social costs. Fleet deployment models are analyzed in [212]–[215]. The optimal allocation of SBs in a city is studied through mathematical programming models in [212], and solved through stochastic optimization in [213], [214]. Finally, [216] explores the impact of μM on urban planning and identifies strategies to increase μMVs utilization.

In summary, existing design frameworks for mobility systems often possess rigid, problem-specific structures that hinder modular and compositional co-design of mobility infrastructure. Moreover, previous works frequently fail to capture crucial aspects of future mobility systems, such as the interplay among various transportation modes, and the specific design parameters of emerging mobility solutions, such as the autonomy levels and serviced networks of AVs, in a computationally tractable and compositional manner.

11.2 Intermodal Mobility Framework

Multi-Commodity Flow Model

The transportation system and its different modes are modeled using the edge-labeled digraph $\mathcal{G} = \langle \mathbf{V}, \mathbf{A}, c \rangle$, sketched in Figure 156. It is described through a set of nodes $\mathbf{V}$ and a set of arcs $\mathbf{A} \subseteq \mathbf{V} \times \mathbf{V}$, labeled with metrics $c : \mathbf{A} \to \mathbf{S}$. Specifically, $c_{ij} := c(\langle i, j \rangle) \in \mathbf{S}$ are the metrics associated to arc $\langle i, j \rangle \in \mathbf{A}$. Metrics of interest include edge length, travel time, energy consumption properties, and congestion models. $\mathcal{G}$ is composed of four layers: The road network layer $\mathcal{G}_{\mathrm{R}} = \langle \mathbf{V}_{\mathrm{R}}, \mathbf{A}_{\mathrm{R}}, c_{\mathrm{R}} \rangle$, consisting of an AVs layer $\mathcal{G}_{\mathrm{R,V}} = \langle \mathbf{V}_{\mathrm{R,V}}, \mathbf{A}_{\mathrm{R,V}}, c_{\mathrm{R,V}} \rangle$ and a μMVs layer $\mathcal{G}_{\mathrm{R,M}} = \langle \mathbf{V}_{\mathrm{R,M}}, \mathbf{A}_{\mathrm{R,M}}, c_{\mathrm{R,M}} \rangle$, the public transportation layer $\mathcal{G}_{\mathrm{P}} = \langle \mathbf{V}_{\mathrm{P}}, \mathbf{A}_{\mathrm{P}}, c_{\mathrm{P}} \rangle$, and a walking layer $\mathcal{G}_{\mathrm{W}} = \langle \mathbf{V}_{\mathrm{W}}, \mathbf{A}_{\mathrm{W}}, c_{\mathrm{W}} \rangle$. The AVs and the μMVs networks are characterized by intersections $i \in \mathbf{V}_{\mathrm{R,V}}$, $i \in \mathbf{V}_{\mathrm{R,M}}$ and road segments $\langle i, j \rangle \in \mathbf{A}_{\mathrm{R,V}}$, $\langle i, j \rangle \in \mathbf{A}_{\mathrm{R,M}}$, respectively. Similarly, public transportation lines are modeled through station nodes $i \in \mathbf{V}_{\mathrm{P}}$ and line segments $\langle i, j \rangle \in \mathbf{A}_{\mathrm{P}}$. The walking network describes walkable streets $\langle i, j \rangle \in \mathbf{A}_{\mathrm{W}}$, connecting

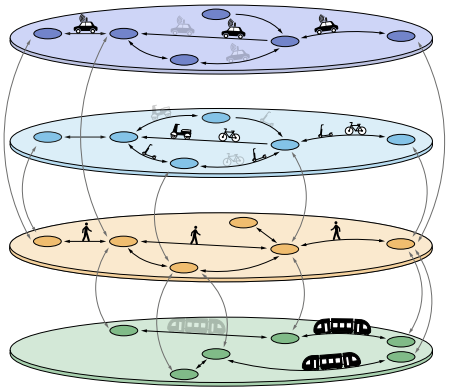


Figure 156: The intermodal AMoD network consists of road (AVs and μMVs), public transportation, and walking digraphs. The labeled circles represent stops or intersections and the black arrows denote road links, public transit arcs, or pedestrian pathways. The grey arrows represent the mode-switching arcs connecting them.

intersections $i \in \mathbf{V}_{\mathrm{W}}$. Mode-switching arcs are modeled as

$$\mathbf{A}_{\mathrm{C}} \subseteq \mathbf{V}_{\mathrm{R,V}} \times \mathbf{V}_{\mathrm{W}} \cup \mathbf{V}_{\mathrm{W}} \times \mathbf{V}_{\mathrm{R,V}} \cup \mathbf{V}_{\mathrm{R,M}} \times \mathbf{V}_{\mathrm{W}} \cup \mathbf{V}_{\mathrm{W}} \times \mathbf{V}_{\mathrm{R,M}} \cup \mathbf{V}_{\mathrm{P}} \times \mathbf{V}_{\mathrm{W}} \cup \mathbf{V}_{\mathrm{W}} \times \mathbf{V}_{\mathrm{P}},$$

connecting the AVs, the μMVs, and the public transportation layers to the walking layer. To these arcs, we associate metrics c_{C}.

Consequently, $\mathbf{V} = \mathbf{V}_{\mathrm{W}} \cup \mathbf{V}_{\mathrm{R,V}} \cup \mathbf{V}_{\mathrm{R,M}} \cup \mathbf{V}_{\mathrm{P}}$ and $\mathbf{A} = \mathbf{A}_{\mathrm{W}} \cup \mathbf{A}_{\mathrm{R,V}} \cup \mathbf{A}_{\mathrm{R,M}} \cup \mathbf{A}_{\mathrm{P}} \cup \mathbf{A}_{\mathrm{C}}$. Consistently with structural properties of transportation networks in urban environments, we assume $\mathcal{G}$ to be strongly connected.

We now characterize the partial order of edge labeled graphs, which will be instrumental when modeling the monotone design problem with implementation (MDPI) of the mobility system.

Definition 11.1 (Poset of edge-labeled multigraphs)
Consider the set of edge-labeled multigraphs, denoted by $\mathbf{G}$. Given $\mathcal{G}_1, \mathcal{G}_2 \in \mathbf{G}$, with $\mathcal{G}_1 = \langle \mathbf{V}_1, \mathbf{A}_1, c_1 \rangle$, $\mathcal{G}_2 = \langle \mathbf{V}_2, \mathbf{A}_2, c_2 \rangle$, $c_1 : \mathbf{A}_1 \to \mathbf{R}$, $c_2 : \mathbf{A}_2 \to \mathbf{R}$ we define the order:

$$\mathcal{G}_1 \preceq_{\mathbf{G}} \mathcal{G}_2 \Leftrightarrow (\mathbf{V}_1 \subseteq \mathbf{V}_2) \wedge (\mathbf{A}_1 \subseteq \mathbf{A}_2) \wedge (c_1 \succeq_{\mathbf{R}^{\mathbf{A}_1}} c_2|_{\mathbf{A}_1}),$$

where $c_2|_{\mathbf{A}_1}$ is the restriction of c_2 onto the domain of c_1.

Intuitively, a labeled multigraph dominates another if it includes its nodes and edges, and if the labels are dominating in their respective spaces.

Lemma 11.2. Definition 11.1 defines a poset.

See proof on page 245.

We represent customer movements by means of travel requests. A travel request refers to a customer flow starting its trip at a node $o \in \mathbf{V}$ and ending it at a node $d \in \mathbf{V}$.

Definition 11.3 (Travel demand)
A *travel demand* q is a triple $\langle o, d, \alpha \rangle \in \mathbf{V} \times \mathbf{V} \times \mathbb{R}_{\geq 0}$, described by an origin node $o \in \mathbf{V}$, a destination node $d \in \mathbf{V}$, and the request rate $\alpha > 0$ (i.e., the number of customers who want to travel from o to d per unit time).

Without loss of generality, we can assume that in a set of requests origin-destination pairs are not repeated, and denote the set of all possible set of requests $\mathbf{Q} \subset \mathsf{Pow}(\mathbf{V} \times \mathbf{V} \times \overline{\mathbb{R}}_{\geq 0})$. This set can be ordered as follows.

Definition 11.4 (Poset of travel demand)
Consider the set of sets of travel requests $\mathbf{Q}$. Given any $q_1, q_2 \in \mathbf{Q}$, one has:

$$q_1 := \{\langle o_i^1, d_i^1, \alpha_i^1\rangle\}_{i=1}^{M_1} \preceq_{\mathrm{Q}} \{\langle o_i^2, d_i^2, \alpha_i^2\rangle\}_{i=1}^{M_2} =: q_2$$

iff for all $\langle o^1, d^1, \alpha^1\rangle \in q_1$ there is some $\langle o^2, d^2, \alpha^2\rangle \in q_2$ with $o^1 = o^2$, $d^1 = d^2$, and $\alpha_i^2 \geq \alpha_i^1$. In other words, $q_1 \preceq_{\mathrm{Q}} q_2$ if every travel request in q_1 is in q_2 as well.

Lemma 11.5. Definition 11.4 defines a poset.

See proof on page 245.

To ensure that a customer is not biased to use a given transportation mode, we assume all requests to appear on the walking digraph, i.e., $o_m, d_m \in \mathbf{V}_{\mathrm{W}}$ for all $m \in \mathbf{M} := \{1, \dots, M\}$. The flow $f_m(i,j) \geq 0$ describes the number of customers per unit time traveling on arc $\langle i, j\rangle \in \mathbf{A}$ and satisfying a travel request m. Furthermore, $f_{0,\mathrm{V}}(i,j) \geq 0$ and $f_{0,\mathrm{M}}(i,j) \geq 0$ denote the flow of empty AVs and μMVs on AVs arcs $\langle i, j\rangle \in \mathbf{A}_{\mathrm{R,V}}$ and μMVs arcs $\langle i, j\rangle \in \mathbf{A}_{\mathrm{R,M}}$, respectively. This accounts for rebalancing flows of AVs and μMVs between a customer's drop-off and the next customer's pick-up. Assuming AVs and μMVs to carry one customer at a time, the flows satisfy

$$\sum_{i:\langle i,j\rangle\in\mathbf{A}} f_m(i,j) + \mathbb{1}_{j=o_m}\cdot\alpha_m = \sum_{k:\langle j,k\rangle\in\mathbf{A}} f_m(j,k) + \mathbb{1}_{j=d_m}\cdot\alpha_m \qquad \forall m \in \mathbf{M},\ j \in \mathcal{V} \tag{63a}$$

$$\sum_{i:\langle i,j\rangle\in\mathbf{A}_{\mathrm{R,V}}} f_{\mathrm{tot,V}}(i,j) = \sum_{k:\langle j,k\rangle\in\mathbf{A}_{\mathrm{R,V}}} f_{\mathrm{tot,V}}(j,k) \quad \forall j \in \mathbf{A}_{\mathrm{R,V}} \tag{63b}$$

$$\sum_{i:\langle i,j\rangle\in\mathbf{A}_{\mathrm{R,M}}} f_{\mathrm{tot,M}}(i,j) = \sum_{k:\langle j,k\rangle\in\mathbf{A}_{\mathrm{R,M}}} f_{\mathrm{tot,M}}(j,k) \quad \forall j \in \mathbf{A}_{\mathrm{R,M}}, \tag{63c}$$

where $\mathbb{1}_{j=x}$ denotes the boolean indicator function, $f_{\mathrm{tot,V}}(i,j) := f_{0,\mathrm{V}}(i,j) + \sum_{m\in\mathcal{M}} f_m(i,j)$, and $f_{\mathrm{tot,M}}(i,j) := f_{0,\mathrm{M}}(i,j) + \sum_{m\in\mathcal{M}} f_m(i,j)$. Specifically, (63a) guarantees flows conservation for every transportation demand, (63b) preserves flow conservation for AVs on every road node, and (63c) preserves flow conservation for μMVs on every road node. Combining conservation of customers (63a) with the conservation of AVs (63b) and μMVs (63c) guarantees rebalancing AVs and μMVs to match the demand.

Remark 11.6. The demand is assumed to be time-invariant and flows are allowed to have fractional values. This assumption is in line with the mesoscopic and system-level planning perspective of the proposed study. We allow AVs and μMVs to transport one customer at a time [217].

Labeling Graphs with Relevant Attributes

In the following, we specify how to label the graphs composing the full mobility network. Specifically, edge-labeling maps will be of the form $c : \mathbf{A} \to \mathbf{S}$, where $\mathbf{S} = \left\langle \mathbb{R}^5_{\geq 0}, \preceq_{\mathbf{S}} \right\rangle$ represents link length, time needed to traverse it, its speed limit, related emissions, and capacity, and the order the product order on $\mathbb{R}^5_{\geq 0}$.

Walking arcs We infer arc lengths s_{ij} from geographical data and, assuming constant walking speed v_{W}, travel time results from $t_{ij} = s_{ij}/v_{\mathrm{W}}$.

As speeds limits, congestion, and energy consumption do not apply to walking graphs, we set $v_{\mathrm{L},ij} = \infty$, $k_{ij} = \infty$, $e_{ij} = 0$. Accordingly,

$$\begin{aligned} c_{\mathrm{W}} : \mathbf{A}_{\mathrm{W}} &\to \mathbf{S} \\ \langle i, j \rangle &\mapsto \langle s_{ij}, t_{ij}, v_{\mathrm{L},ij}, e_{ij}, k_{ij} \rangle. \end{aligned}$$

Public transit arcs We infer arc lengths s_{ij} from public transit network data. Furthermore, assuming that the public transportation system at node j operates with the frequency φ_j, travel time results from $t_{ij} = t_{ij}^{\mathrm{nom}} + t_{\mathrm{WS}} + 1/(2\varphi_j)$, where t_{ij}^{nom} is the in-vehicle travel time (inferred from public transit schedules) and t_{WS} is a constant sidewalk-to-station travel time. We ignore capacity and speed limits, so that $k_{ij} = v_{\mathrm{L},ij} = \infty$. For the public transportation system we assume a constant energy consumption per unit time. This approximation is reasonable in urban environments, where the operation of the public transportation system is independent from the number of customer serviced, and its energy consumption is therefore invariant. Therefore, we write $e_{ij} = \kappa t_{ij}, \kappa > 0$. Accordingly,

$$\begin{aligned} c_{\mathrm{P}} : \mathbf{A}_{\mathrm{P}} &\to \mathbf{S} \\ \langle i, j \rangle &\mapsto \langle s_{ij}, t_{ij}, v_{\mathrm{L},ij}, e_{ij}, k_{ij} \rangle. \end{aligned}$$

Road arcs for AVs Each road arc is characterized by a length s_{ij}, a speed limit $v_{\mathrm{L},ij}$, and a capacity k_{ij}, all derived from road network data. We consider AVs driving at speed v_{V}, so that travel time reads

$$t_{ij} = \frac{s_{ij}}{\min\{v_{\mathrm{V}}, v_{\mathrm{L},ij}\}}.$$

We compute the energy consumption of AVs via an urban driving cycle. In particular, the cycle is scaled so that its average speed $v_{\mathrm{avg,cycle}}$ matches the free-flow speed on the link. The energy consumption of road link a is scaled as

$$e_{ij} = e_{\mathrm{cycle}} \cdot \frac{s_{ij}}{s_{\mathrm{cycle}}}.$$

Collectively,

$$c_{\mathrm{R,V}} : \mathbf{A}_{\mathrm{R,V}} \to \mathbf{S}$$
$$\langle i, j \rangle \mapsto \langle s_{ij}, t_{ij}, v_{\mathrm{L},ij}, e_{ij}, k_{ij} \rangle.$$

Road arcs for μMVs Each road arc is characterized by a length s_{ij} and a speed limit $v_{\mathrm{L},ij}$, derived from road network data, while we neglect arc capacity (i.e., $k_{ij} = \infty$). Assuming μMVs driving at speed v_{M}, travel time reads

$$t_{ij} = \frac{s_{ij}}{\min\{v_{\mathrm{M}}, v_{\mathrm{L},ij}\}}.$$

For μMVs we consider a distance-based energy consumption, i.e. $e_{ij} = \iota s_{ij}$, with $\iota > 0$. Overall,

$$c_{\mathrm{R,M}} : \mathbf{A}_{\mathrm{R,M}} \to \mathbf{S}$$
$$\langle i, j \rangle \mapsto \langle s_{ij}, t_{ij}, v_{\mathrm{L},ij}, e_{ij}, k_{ij} \rangle.$$

Transfer arcs We define travel time t_{ij} as follows: we assume that the average waiting time for AVs is t_{WV}, the average time needed to reach a μMV is t_{WM}, and switching from the AVs graph, the μMVs graph, and the public transit graph to the pedestrian graph takes the transfer times t_{VW}, t_{MW}, and t_{SW}, respectively. For each arc, we set length and energy consumption to zero (i.e., $s_{ij} = e_{ij} = 0$) and ignore capacity and speed limit (i.e., $k_{ij} = v_{\mathrm{L},ij} = \infty$). Overall,

$$c_{\mathrm{C}} : \mathbf{A}_{\mathrm{C}} \to \mathbf{S}$$
$$\langle i, j \rangle \mapsto \langle s_{ij}, t_{ij}, v_{\mathrm{L},ij}, e_{ij}, k_{ij} \rangle.$$

Road congestion

We assume that road arcs are subject to a normalized capacity k_{ij}, which could arise from the difference of the nominal road capacity and the exogenous road usage:

$$f_{\mathrm{tot,V}}(i, j) \leq k_{ij}. \tag{64}$$

We assume that the central authority operates the AMoD fleet such that vehicles travel at free-flow speed throughout the road network of the city, meaning that the total flow on each road link must be below the link's capacity. Therefore, we capture congestion effects with the threshold model. Finally, we assume μM to not significantly contribute to congestion [218].

Remark 11.7 (Threshold model for congestion)**.** We model congestion effects using a threshold model. This approach can be interpreted as a municipality preventing mobility solutions to exceed the critical flow density on road arcs.

AVs and μMVs can therefore be assumed to travel at free flow speed [219]. This assumption is realistic for an initial low penetration of new mobility systems in the transportation market, especially when the AV and μMV fleets are limited in size.

11.3 Co-Design Framework

We integrate the intermodal framework presented in the previous sections in the co-design formalism, allowing the decoupling of the entire co-design problem of a complex system in the design problems of its individual components. To achieve this, we decouple the co-design problem in the design of the individual AV, the AVs fleet, the individual μMV, the μMVs fleet, and the public transportation system. We then look at their interconnection, where we propose multiple model versions, showcasing the flexibility of the developed framework. We aim at computing the antichain of resources, quantified in terms of costs, average travel time per trip, and emissions required to provide the mobility service to a set of customers. For each model, we provide descriptions and formal proofs of integration in the co-design framework.

The AV design problem

The AV design problem selects the labeled graph on which the AV provider wants to operate. The selection happens via the choice of the achievable speed of the AVs as follows. AVs safety protocols impose a maximum achievable velocity v_{V}. Furthermore, in order to prevent too slow and therefore dangerous driving behaviors [220], we only consider AVs arcs through which the AVs can drive at least at a fraction β of the speed limit. Specifically, AVs can drive on arc $\langle i, j\rangle \in \mathbf{A}_{\mathrm{R,V}}$ if and only if

$$v_{\mathrm{V}} \geq \beta \cdot \pi_{v_{\mathrm{L}}} c_{\mathrm{R,V}}(i, j), \tag{65}$$

where $\beta \in (0, 1]$, and $\pi_{v_{\mathrm{L}}}$ projects the part of $c_{\mathrm{R,V}}(i, j)$ related to v_{L}. The elimination of forbidden arcs given an achievable speed can be achieved through the following map (mnemonics for reduction):

$$\begin{aligned} \mathrm{red}_{\mathrm{R,V}} : \overline{\mathbb{R}}_{\geq 0} &\to \langle \mathbf{G}, \preceq_{\mathbf{G}} \rangle \\ v_{\mathrm{V}} &\mapsto \left\langle \mathbf{V}_{\mathrm{R,V}}, \mathbf{A}, c \right\rangle, \end{aligned}$$

where

$$\begin{aligned} \mathbf{A} &= \{\langle i, j\rangle \in \mathbf{A}_{\mathrm{R,V}} : (65) \text{ holds}\}, \\ c &= \left\langle \pi_s c_{\mathrm{R,V}}, \frac{\pi_s c_{\mathrm{R,V}}}{\min\{v_{\mathrm{V}}, \pi_{v_{\mathrm{L}}} c_{\mathrm{R,V}}\}}, \pi_{v_{\mathrm{L}}} c_{\mathrm{R,V}}, \pi_e c_{\mathrm{R,V}}, \pi_k c_{\mathrm{R,V}} \right\rangle. \end{aligned} \tag{66}$$

We can now state the following fact.

Lemma 11.8. The map $\mathrm{red}_{\mathrm{R,V}}$ is monotone.

See proof on page 245.

In other words, the graph on which the AVs can operate is limited by the AV achievable speed.

Under the rationale that driving safely at higher speed requires more advanced sensing and algorithmic capabilities [128], we model the achievable speed of the AVs υ_{V} as a *monotone* function of the vehicle fixed costs $C_{\mathrm{V,f}}$ (resulting from the cost of the vehicle $C_{\mathrm{V,v}}$ and the cost of its automation $C_{\mathrm{V,a}}$) and the mileage-dependent operational costs $C_{\mathrm{V,o}}$ (accounting for maintenance, cleaning, energy consumption, depreciation, and opportunity costs [221]).

Co-design formulation The AV design problem, denoted d_{AV}, provides the functionality $\mathcal{G}_{\mathrm{AV}} \in \mathbf{G}$ (i.e., the functionality of servicing a specific network with a specific performance) and requires the resources $C_{\mathrm{V,f}}, C_{\mathrm{V,o}} \in \overline{\mathbb{R}}_{\geq 0}$ (Fig. 157). The implementations space $\mathbf{I}_{\mathrm{AV}}$ consists of models of the AVs.

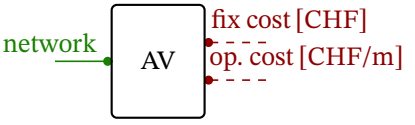


Figure 157: Design problem for an AV.

Lemma 11.9. d_{AV} is a well-defined design problem.

Proof. $C_{\mathrm{V,f}}, C_{\mathrm{V,o}}$ are monotone functions of the AV's achievable speed. Leveraging Lemma 11.8, we know that the serviced network is a monotone function of the speed. □

Remark 11.10. At this point you might be impatient to substitute this simple model of an AV with more complex ones developed in previous sections. Indeed, here we lump the AV's autonomy in its achievable velocity. Hold on! We will do this in Chapter 12.

The μMV design problem

The μM design problem comprises the selection of the labeled graph on which to operate, again resumed in the maximal speed achievable by μMVs. Given an achievable speed υ_{M}, one obtains the resulting graph as follows:

$$\begin{aligned} \mathrm{red}_{\mathrm{R,M}} : \overline{\mathbb{R}}_{\geq 0} &\to \langle \mathbf{G}, \preceq_{\mathrm{G}} \rangle \\ \upsilon_{\mathrm{M}} &\mapsto \langle \mathbf{V}_{\mathrm{R,M}}, \mathbf{A}_{\mathrm{R,M}}, c \rangle, \end{aligned}$$

where

$$c = \left\langle \pi_s c_{\mathrm{R,M}}, \frac{\pi_s c_{\mathrm{R,M}}}{\min\{\upsilon_{\mathrm{M}}, \pi_{\upsilon_{\mathrm{L}}} c_{\mathrm{R,M}}\}}, \pi_{\upsilon_{\mathrm{L}}} c_{\mathrm{R,M}}, \pi_e c_{\mathrm{R,M}}, \pi_k c_{\mathrm{R,M}} \right\rangle.$$

Lemma 11.11. The map $\mathrm{red}_{\mathrm{R,M}}$ is monotone.

See proof on page 245.

Following the rationale that different μMVs can reach different speeds and have different prices, we model the achievable speed of the μMV υ_{M} as a *monotone* function of the μMV fixed costs $C_{\mathrm{M,f}}$ and the mileage-dependent operational costs $C_{\mathrm{M,o}}$.

Co-design formalization Therefore, the μM design problem, denoted d_{MM}, provides the functionality $\mathcal{G}_{\mathrm{MM}} \in \mathbf{G}$ (i.e., the functionality of servicing a specific network with a specific performance) and requires the resources $C_{\mathrm{M,f}}, C_{\mathrm{M,o}} \in \overline{\mathbb{R}}_{\geq 0}$ (Fig. 158). The implementations space $\mathbf{I}_{\mathrm{M}}$ consists of instances of the μMVs.

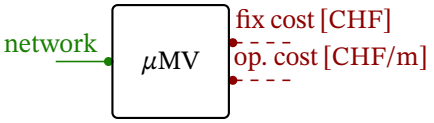


Figure 158: μMV design problem.

Lemma 11.12. d_{MM} is a well-defined design problem.

Proof. $C_{\mathrm{M,f}}, C_{\mathrm{M,o}}$ are monotone functions of the μMV's achievable speed. Leveraging Lemma 11.11, we know that the serviced network is a monotone function of the speed. □

The Subway design problem

The public transit design problem comprises the selection of the labeled network on which to operate, now resumed in the choice of fleet size for the subway system. Specifically, we assume the service frequency φ_j to scale monotonically with the size of the train fleet n_{S}. In the linear case, one has:

$$\frac{\varphi_j}{\varphi_{j,\mathrm{base}}} = \frac{n_{\mathrm{S}}}{n_{\mathrm{S,base}}},$$

where $\varphi_{j,\mathrm{base}}$ and $n_{\mathrm{S,base}}$ are respective existing baselines. Given a train fleet size, one obtains the resulting network as follows:

$$\begin{aligned} \mathrm{red}_{\mathrm{P}} : \left\langle \overline{\mathbb{N}}, \leq \right\rangle &\to \langle \mathbf{G}, \preceq_{\mathrm{G}} \rangle \\ n_{\mathrm{S}} &\mapsto \langle \mathbf{V}_{\mathrm{P}}, \mathbf{A}_{\mathrm{P}}, c \rangle, \end{aligned}$$

where

$$c = \left\langle \pi_s c_{\mathrm{P}}, t_{\mathrm{WS}} + \frac{n_{\mathrm{S,base}}}{2 n_{\mathrm{S}} \varphi_{j,\mathrm{base}}}, \pi_{\upsilon_{\mathrm{L}}} c_{\mathrm{P}}, \pi_e c_{\mathrm{P}}, \pi_k c_{\mathrm{P}} \right\rangle.$$

Lemma 11.13. The map $\mathrm{red}_{\mathrm{P}}$ is monotone.

See proof on page 245.

We relate a train fleet of size n_{S} to the fixed costs $C_{\mathrm{S,f}}$ (accounting for train and infrastructural costs) and to the operational costs $C_{\mathrm{S,o}}$ (accounting for energy consumption, vehicles depreciation, and train operators' wages). Given the

passengers-independent public transit operation in today's cities, we assume the operational costs $C_{S,o}$ to be mileage independent and to only vary with the size of the fleet. Assuming an average train's life of l_S, and a baseline subway fleet of $n_{S,baseline}$ trains, costs are

$$C_S = \frac{C_{S,f}}{l_S} \cdot n_{S,a} + C_{S,o}.$$

Moreover, operating a fleet of trains entails the CO_2 emissions

$$m_{CO_2,S,tot} = m_{CO_2,S} \cdot n_S.$$

Co-design formalization The public transit design problem, denoted d_P, provides the functionality $\mathcal{G}_P \in \mathbf{G}$ (i.e., the functionality of servicing a specific network with a specific performance) and requires the resources $C_S \in \overline{\mathbb{R}}_{\geq 0}$ and $m_{CO_2,S,tot} \in \overline{\mathbb{R}}_{\geq 0}$. The implementations space $\mathbf{I}_P$ consists of different train acquisition choices. Formally: $d_P : \mathbf{G} \nrightarrow \overline{\mathbb{R}}^2_{\geq 0}$ (Fig. 159).

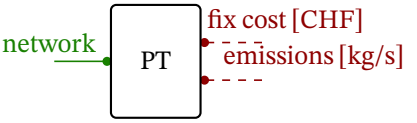


Figure 159: μMV design problem.

Lemma 11.14. d_P is a well-defined design problem.

Proof. First, notice that C_S and $m_{CO_2,S,tot}$ are monotone functions of n_S. Furthermore, leveraging Lemma 11.13, we know that the serviced network relates monotonically to n_S. □

The Intermodal Mobility System design problem (Version 1)

The first version of the intermodal mobility system design problem considers demand satisfaction as a functionality, and does not consider micromobility.

To successfully satisfy a given set of travel requests, we require the following resources:

- the network resulting from the design of AVs, $\mathcal{G}_{AV} = \langle \mathbf{V}_{AV}, \mathbf{A}_{AV}, c_{AV} \rangle$,
- the network resulting from the design of public transit $\mathcal{G}_P = \langle \mathbf{V}_P, \mathbf{A}_P, c_P \rangle$,
- the number of available AVs per fleet $n_{V,max}$,
- the average travel time of a trip

$$t_{avg} := \frac{1}{\alpha_{tot}} \sum_{\substack{m \in \mathcal{M}, \\ \langle i,j \rangle \in \mathbf{A}_W \cup \mathbf{A}_{AV} \cup \mathbf{A}_P \cup \mathbf{A}_C}} \pi_t c(i,j) \cdot f_m(i,j),$$

with

$$\alpha_{tot} := \sum_{m \in \mathcal{M}} \alpha_m, \tag{67}$$

▷ the total distance driven by the AVs per unit time

$$s_{\mathrm{V,tot}} := \sum_{\langle i,\, j\rangle \in \mathbf{A}_{\mathrm{AV}}} \pi_s c_{\mathrm{AV}}(i,j) \cdot f_{\mathrm{tot,V}}(i,j), \tag{68}$$

▷ the total AVs CO_2 emissions per unit time

$$m_{\mathrm{CO_2,V,tot}} := \gamma \cdot \sum_{\langle i,\, j\rangle \in \mathbf{A}_{\mathrm{AV}}} \pi_e c_{\mathrm{AV}} \cdot f_{\mathrm{tot,V}}(i,j). \tag{69}$$

We assume that AVs are routed to maximize the customers' welfare, defined without loss of generality as the average travel time t_{avg}. Hence, we link functionality and resources of the mobility system design problem through the optimization problem:

$$\begin{aligned} &\min_{\substack{\{f_m\}_m \\ f_{0,\mathrm{V}}}} t_{\mathrm{avg}} \\ &\text{s.t. Eq.(63)}, \\ &\quad \text{Eq.(64)} \quad \forall \langle i,j\rangle \in \mathbf{A}_{\mathrm{AV}}, \\ &\qquad \sum_{\langle i,\, j\rangle \in \mathbf{A}_{\mathrm{AV}}} f_{\mathrm{tot,V}}(i,j) \cdot \pi_t c_{\mathrm{AV}}(i,j) \le n_{\mathrm{V,max}}, \end{aligned} \tag{70}$$

where we express the number of vehicles on arc $\langle i,\, j\rangle$ as the multiplication of the total vehicles flow on the arc and its travel time.

Co-design formalization The intermodal mobility system design problem has as functionality the satisfied requests $q \in \mathbf{Q}$ and the mentioned resources. Furthermore, $\mathbf{I}_{\mathrm{O}}$ consists of specific intermodal scenarios.

Lemma 11.15. $d_{\mathrm{IAMOD,1}}$ is a well-defined design problem.

See proof on page 246.

The Intermodal Mobility System design problem (Version 2)

The second version of the intermodal mobility system design problem still considers demand satisfaction as a functionality, now including μM options. To successfully satisfy a given set of travel requests, we require the following resources:

▷ $\mathcal{G}_{\mathrm{AV}} = \langle \mathbf{V}_{\mathrm{AV}}, \mathbf{A}_{\mathrm{AV}}, c_{\mathrm{AV}} \rangle$ as in Section 11.3,

▷ $\mathcal{G}_{\mathrm{P}} = \langle \mathbf{V}_{\mathrm{P}}, \mathbf{A}_{\mathrm{P}}, c_{\mathrm{P}} \rangle$ as in Section 11.3,

▷ the network resulting from the design of μMs $\mathcal{G}_{\mathrm{MM}} = \langle \mathbf{V}_{\mathrm{MM}}, \mathbf{A}_{\mathrm{MM}}, c_{\mathrm{MM}} \rangle$,

▷ $n_{\mathrm{V,max}}$ as in Section 11.3,

▷ the number of available μMVs per fleet $n_{\mathrm{M,max}}$,

- the (adapted) average travel time of a trip

$$t_{\text{avg}} := \frac{1}{\alpha_{\text{tot}}} \sum_{\substack{m \in \mathcal{M}, \\ \langle i,j \rangle \in \mathbf{A}_{\text{W}} \cup \mathbf{A}_{\text{AV}} \cup \mathbf{A}_{\text{MM}} \cup \mathbf{A}_{\text{P}} \cup \mathbf{A}_{\text{C}}}} \pi_t c(i,j) \cdot f_m(i,j),$$

  with α_{tot} as in (67),
- $s_{\text{V,tot}}$ as in (68),
- the total distance driven by the μMVs per unit time

$$s_{\text{M,tot}} := \sum_{\langle i,j \rangle \in \mathbf{A}_{\text{MM}}} \pi_s c_{\text{MM}}(i,j) \cdot f_{\text{tot,M}}(i,j),$$

- $m_{\text{CO}_2\text{,V,tot}}$ as in (69),
- the total μMVs CO_2 emissions per unit time

$$m_{\text{CO}_2\text{,M,tot}} := \gamma \cdot \sum_{\langle i,j \rangle \in \mathbf{A}_{\text{MM}}} \pi_e c_{\text{MM}} \cdot f_{\text{tot,M}}(i,j),$$

  where γ relates energy consumption and CO_2 emissions.

We assume that AVs and μMVs are routed to maximize the customers' welfare, defined without loss of generality as the average travel time t_{avg}. Hence, we link functionality and resources of the mobility system design problem through the following optimization problem, extending (70):

$$\begin{aligned}
&\min_{\substack{\{f_m\}_m \\ f_{0,\text{V}} \\ f_{0,\text{M}}}} t_{\text{avg}} \\
&\text{s.t. Eq.(63)}, \\
&\quad \text{Eq.(64)} \quad \forall \langle i,j \rangle \in \mathbf{A}_{\text{AV}}, \\
&\qquad \sum_{\langle i,j \rangle \in \mathbf{A}_{\text{AV}}} f_{\text{tot,V}}(i,j) \cdot \pi_t c_{\text{AV}}(i,j) \leq n_{\text{V,max}}, \\
&\qquad \sum_{\langle i,j \rangle \in \mathbf{A}_{\text{MM}}} f_{\text{tot,M}}(i,j) \cdot \pi_t c_{\text{MM}}(i,j) \leq n_{\text{M,max}},
\end{aligned} \tag{71}$$

where we express the number of vehicles on arc $\langle i,j \rangle$ as the multiplication of the total vehicles flow on the arc and its travel time.

Co-design formalization The intermodal mobility system design problem has as functionality $Q \in \mathcal{Q}$ and the mentioned resources. Furthermore, $\mathbf{I}_{\text{O}}$ consists of specific intermodal scenarios.

Lemma 11.16. $d_{\text{IAMOD,2}}$ is a well-defined design problem.

Proof. The proofs is parallel the proof of Lemma 11.15. □

The Intermodal Mobility System design problem (Version 3)

We extend the setting presented in Section 11.3 by including a new functionality. Specifically, the intermodal mobility system design problem not only provides demand satisfaction as a functionality, but also provides the revenue ρ arising from the mobility offer, which reads:

$$\rho = p_{\mathrm{AV}} s_{\mathrm{V,tot}} + p_{\mathrm{P}} \sum_{\langle i,j\rangle \in \mathbf{A}_{\mathrm{C}} \cap \mathbf{V}_{\mathrm{W}} \times \mathbf{V}_{\mathrm{P}}} f_m(i,j),$$

where p_{AV} is a distance-based price to use AVs and p_{P} is a fixed entry price for the subway system. Accordingly, we modify the optimization problem to account for both average travel time and average cost of fare:

$$\begin{aligned} \min_{\substack{\{f_m\}_m \\ f_{0,\mathrm{V}}}} \; & V_{\mathrm{T}} t_{\mathrm{avg}} + \frac{1}{\alpha_{\mathrm{tot}}} \rho \\ \text{s.t. Eq.(63)}, & \\ \text{Eq.(64)} \quad & \forall \langle i,j\rangle \in \mathbf{A}_{\mathrm{AV}}, \\ & \sum_{\langle i,j\rangle \in \mathbf{A}_{\mathrm{AV}}} f_{\mathrm{tot,V}}(i,j) \cdot \pi_t c_{\mathrm{AV}}(i,j) \leq n_{\mathrm{V,max}}, \end{aligned} \tag{72}$$

where V_{T} is the value of time.

Co-design formalization This new version of the intermodal mobility system design problem has as functionality the satisfied requests $Q \in \mathcal{Q}$ and the total revenue $\rho \in \overline{\mathbb{R}}_{\geq 0}$ and the mentioned resources. Furthermore, $\mathbf{I}_{\mathrm{O}}$ consists of specific intermodal scenarios (including specific price choices).

Lemma 11.17. $d_{\mathrm{IAMOD,3}}$ is a well-defined design problem.

Proof. The proofs is parallel the proof of Lemma 11.15. □

The Mobility design problem (Version 1)

The functionality of the system is to satisfy the customers' demand. Formally, the functionality provided by the co-design problem with implementation (CDPI) is the set of travel requests and coincides with the functionalities of d_{I_1}. To provide the mobility service, three resources are required. First, on the customers' side, we require the average travel time defined in Section 11.3. Second, on the side of the central authority, the resource is the total transportation cost of the intermodal

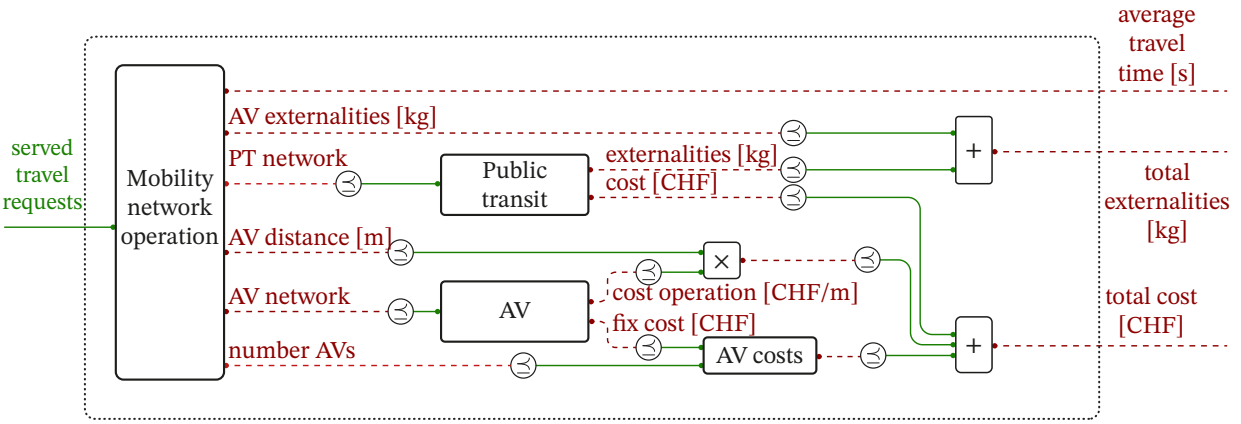


Figure 160: Design problem for the mobility system (version 1).

mobility system. Assuming an average AV's life of l_{V}, an average μMV's life of l_{M}, we express the total costs as

$$C_{\mathrm{tot}} = C_{\mathrm{V}} + C_{\mathrm{S}},$$

where C_{V} is the AVs-related cost

$$C_{\mathrm{V}} = \frac{C_{\mathrm{V,f}}}{l_{\mathrm{V}}} \cdot n_{\mathrm{V,max}} + C_{\mathrm{V,o}} \cdot s_{\mathrm{V,tot}},$$

and C_{S} is the public transit-related cost. Third, on the environmental side, we consider the total CO_2 emissions

$$m_{\mathrm{CO_2,tot}} = m_{\mathrm{CO_2,V,tot}} + m_{\mathrm{CO_2,S,tot}}.$$

Co-design formalization The formal diagram, following the logical interconnections, is reported in Fig. 160.

Lemma 11.18. $d_{\mathrm{Mob_1}}$ is a well-defined design problem.

> *Proof.* The design problem is valid, since they consist of the valid composition of design problems [122]. □

The Mobility design problem (Version 2)

As in Section 11.3, the functionality provided by the design problem is the set of travel requests. To provide the mobility service, three resources are required. First, on the customers' side, we require an average travel time, defined in (11.3). Second, on the side of the central authority, the resource is the total transportation

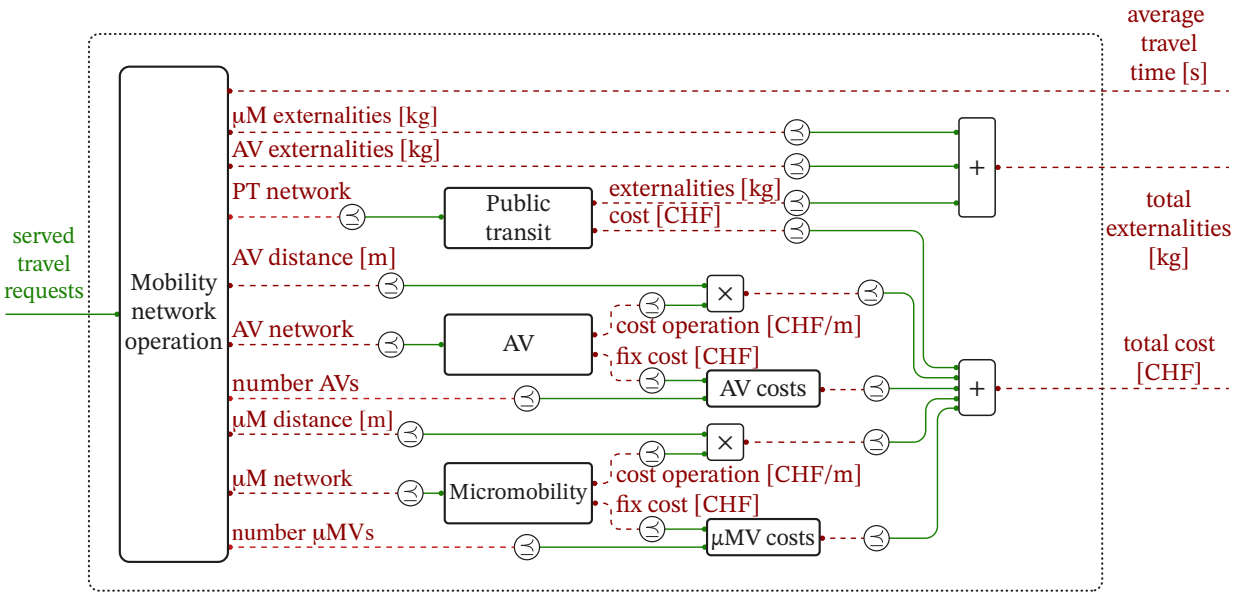


Figure 161: Design problem for the mobility system (version 2).

cost of the intermodal mobility system. To the cost defined in Section 11.3, we add the cost related to μM. Assuming an average μMV's life of l_{M} we get

$$C_{\mathrm{tot}} = C_{\mathrm{V}} + C_{\mathrm{M}} + C_{\mathrm{S}},$$

where C_{M} is the μMV-related cost

$$C_{\mathrm{M}} = \frac{C_{\mathrm{M,f}}}{l_{\mathrm{M}}} \cdot n_{\mathrm{M,max}} + C_{\mathrm{M,o}} \cdot s_{\mathrm{M,tot}},$$

Third, we add the μM-related emissions to the ones computed in Section 11.3:

$$m_{\mathrm{CO_2,tot}} = m_{\mathrm{CO_2,V,tot}} + m_{\mathrm{CO_2,M,tot}} + m_{\mathrm{CO_2,S,tot}}.$$

Co-design formalization Formally: $d_{\mathrm{Mob_2}} : \mathcal{Q} \nrightarrow \overline{\mathbb{R}}_{\geq 0}^{\,3}$. The MDPI is reported in Fig. 161.

Lemma 11.19. $d_{\mathrm{Mob_2}}$ is a well-defined design problem.

The Mobility design problem (Version 3)

We now extend the setting presented in Section 11.3 by including the structure presented in Section 11.3. Specifically, the functionality provided by the design problem coincides with the functionalities of $d_{\mathrm{I_3}}$ (i.e., includes travel requests and

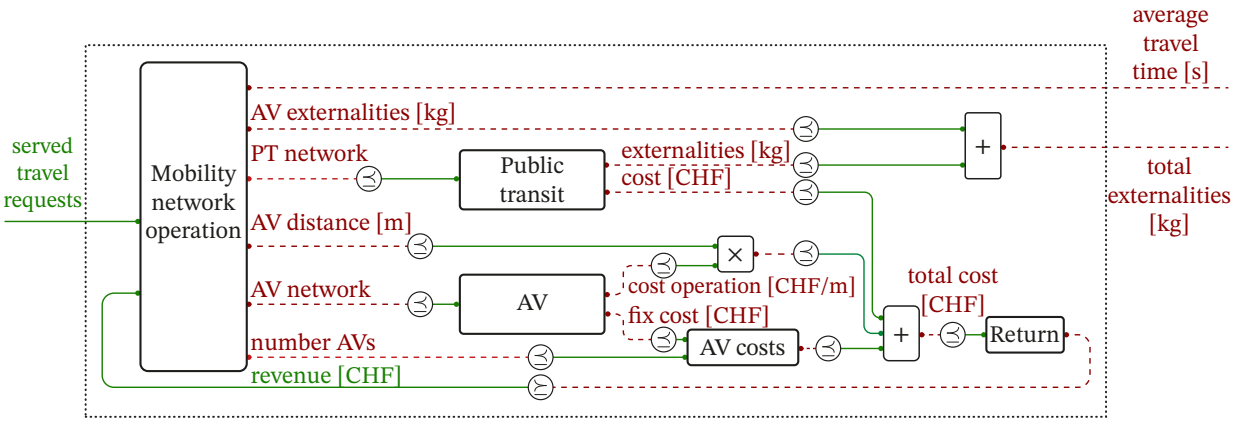


Figure 162: Design problem for the mobility system (version 3).

total revenue). Furthermore, to provide the functionalities the three resources introduced in Section 11.3 are required: t_{avg} C_{tot}, and $m_{\text{CO}_2,\text{tot}}$. We introduce a feedback loop, by requiring the total revenue ρ to at least cover a fraction χ of the total costs, i.e., $\rho \geq \chi \cdot C_{\text{tot}}$.

Co-design formalization Formally: $d_{\text{Mob}_3} : \mathcal{Q} \nrightarrow \overline{\mathbb{R}}_{\geq 0}^{\,2}$. The MDPI is reported in Fig. 162.

Lemma 11.20. d_{Mob_3} is a well-defined design problem.

Remark 11.21. Note that the intermodal mobility framework is only one of the many feasible ways to map total demand to travel time, costs, and emissions. Specifically, practitioners can replace the corresponding design problem (here specified via a multi-commodity flow model and an optimization problem) with different models (e.g., MATSim [222]), as long as the light condition of monotonicity of the design problem is preserved. In this sense, the framework is user-friendly, allowing users to plug in different models and analyze the results.

11.4 Results

In this section, we showcase the co-design framework on the case of Washington D.C., USA, leveraging real mobility data.

Design of Experiments

Our example study is based on the urban area of Washington D.C., USA. The city road network and its features are imported from OpenStreetMap [223], whilst the public transit network together with its schedules are extracted from GTFS [224].

Original demand data is obtained by merging origin-destination pairs of the morning peak of May first, 2017, provided by taxi companies [225] and the Washington Metropolitan Area Transit Authority (WMATA) [226]. On the public transportation side, we focus our studies on the MetroRail system and its design. To take account of the recently increased presence of ride-hailing companies, the taxi demand rate is scaled by a factor of 5 [227]. The complete demand dataset includes 16,430 distinct origin-destination pairs, describing travel requests. To account for congestion effects, the nominal road capacity is computed as in [228] and an average baseline usage of 93 % is assumed, in line with [229]. We assume an AV fleet composed of battery electric BEV-250 mile AVs [230]. We summarize the main parameters characterizing our case studies together with their bibliographic sources in Table 8.

In the remainder of this section, we solve the co-design problems. Beside our basic setting (S1), we evaluate the sensitivity of the design strategies to different models of automation costs of AVs (S2–S4) assess the impact of emerging μM solutions, showing how one can easily include new modes of transportation in the framework (S5), and investigate pricing strategies in (S6). We summarize the considered mobility solutions and their complementarity in Table 9.

S1 - Basic setting: We consider the co-design of the mobility system by means of AMoD and public transportation systems (Section 11.3), and do not include μM solutions (cf. S5). Specifically, we co-design the system by means of the AV fleet size, achievable free-flow speed (see Lemma 11.8), and subway service frequency (see Lemma 11.13): The municipality is allowed to (i) deploy an AV fleet of size $n_{\mathrm{V,max}} \in \{0, 500, 1{,}000, \dots, 5{,}000\}$ vehicles, (ii) choose the single AV achievable speed (determining the serviced mobility network) $v_{\mathrm{V}} \in \{20\,\mathrm{mph}, 25\,\mathrm{mph}, \dots, 50\,\mathrm{mph}\}$, and (iii) increase the subway service frequency φ_j by a factor of 0 %, 50 %, or 100 %. In line with recent literature [231]–[235], [247], we assume an average achievable-velocity-independent cost of automation.

S2 - Speed-dependent automation costs: To relax the potentially unrealistic assumption of a velocity-independent automation cost, we consider a performance-dependent cost structure, detailed in Table 8. The large variance in sensing technologies available on the market and their performances suggests that AV costs are, in fact, performance-dependent [128], [248]. Indeed, the technology currently required to safely operate an autonomous vehicle at 50 mph is substantially more sophisticated, and therefore more expensive, than the one needed at 20 mph. Furthermore, the frenetic evolution of automation techniques will inevitably reduce automation costs: Experts forecast a massive automation cost reduction (up to 90 %) in the next decade, principally due to mass-production of AVs sensing technology [249], [250]. Therefore, we perform our studies with current

Table 8: Parameters, variables, numbers, and units for the case studies.

Parameter		Variable	Value							Units	Source
Road usage		u_{ij}	93							%	[229]
			S1	**S2 (2022)**	**S2 (2025)**	**S3**	**S4**	**S5 (2022)**	**S5 (2025), S6**		
AVs operational cost		$C_{V,o}$	0.084	0.084	0.062	0.084	0.50	0.084	0.062	USD/mile	[230], [231]
Vehicle cost		C_V	32	32	26	32	32	32	26	kUSD/car	[230]
	20 mph		15	20	3.7	500	0	20	3.7	kUSD/car	[231]–[235]
	25 mph		15	30	4.4	500	0	300	4.4	kUSD/car	[231]–[235]
	30 mph		15	55	6.2	500	0	55	6.2	kUSD/car	[231]–[235]
AV automation cost	35 mph	$C_{V,a}$	15	90	8.7	500	0	90	8.7	kUSD/car	[231]–[235]
	40 mph		15	115	9.8	500	0	115	9.8	kUSD/car	[231]–[235]
	45 mph		15	130	12	500	0	130	12	kUSD/car	[231]–[235]
	50 mph		15	150	13	500	0	150	13	kUSD/car	[231]–[235]
AV life		l_V	5	5	5	5	5	5	5	year	[230]
CO_2 per Joule		γ	0.14	0.14	0.14	0.14	0.14	0.14	0.14	g/kJ	[236]
Time $\mathcal{G}_W$ to $\mathcal{G}_{R,V}$		t_{WV}	300	300	300	300	300	300	300	s	-
Time $\mathcal{G}_{R,V}$ to $\mathcal{G}_W$		t_{VW}	60	60	60	60	60	60	60	s	-
Speed limit fraction		β	1/1.3	1/1.3	1/1.3	1/1.3	1/1.3	1/1.3	1/1.3	–	[220]
			ES		**SB**		**M**		**FCM**		
μMV operational cost		$C_{M,o}$	0.79		1.58		2.05		1.20	USD/mile	[237]–[239]
μMV cost		$C_{M,f}$	550		8,860		1,000		3,000	USD/μMV	[238]–[240]
μMV achievable speed		$v_{M,ij}$	15		10		15		15	mph	-
μMV life		l_M	0.085		7.0		10.0		10.0	year	[238]–[240]
μMV emissions		$m_{CO_2,M,tot}$	0.101		0.033		0.158		0.033	kg/mile	[238], [241]–[243]
Time from $\mathcal{G}_W$ to $\mathcal{G}_{R,M}$		t_{WM}	60		60		60		60	s	-
Time from $\mathcal{G}_{R,M}$ to $\mathcal{G}_W$		t_{MW}	60		60		60		60	s	-
	100 %					148,000,000				USD/year	[244]
Subway operational cost	150 %	$C_{S,o}$				222,000,000				USD/year	[244]
	200 %					295,000,000				USD/year	[244]
Subway fixed cost		$C_{S,f}$				14,500,000				USD/train	[245]
Train life		l_S				30				year	[245]
Subway emissions per train		$m_{CO_2,S,tot}$				140,000				kg/year	[246]
Train fleet baseline		$n_{S,base}$				112				train	[245]
Subway service frequency		$\varphi_{j,baseline}$				1/6				1/min	-
Time $\mathcal{G}_W$ to $\mathcal{G}_P$		t_{WS}				60				s	-

(2022) automation costs as well as with their projections for the upcoming years (2025) [230], [247], [250].

S3 - High automation costs: We assess the impact of high automation costs. In particular, we assume a performance-independent automation cost of 0.5 Mil USD/car, capturing the extremely high research and development costs that AVs companies are facing today [251], as well as insurance costs and infrastructural investments. The latter, often referred to as "autonomy-enabling infrastructure", would allow high driving speeds, and could consist of dedicated roads, equipped with sensors and cloud computing capabilities, enhancing the performance of AVs.

S4 - MoD setting: We analyze the current Mobility-on-Demand (MoD) case. The cost structure of MoD systems is characterized by lower vehicle costs (due to lack of automation) and higher operation costs, mainly due to drivers' salaries.

S5 - Impact of new transportation modes: We show the modularity of our framework by evaluating the impact of μM solutions on urban mobility (Section 11.3). We consider ESs (e.g., Lime in DC), SBs (e.g., Capi-

Table 9: Comparison of the considered mobility solutions.

	Mobility Type	Emissions	Cost	Speed	Reliability
Taxi	Point-to-point	High	High operational cost, medium fixed cost	High	Up to availability and congestion
AV	Point-to-point	High	Low operational cost, high fixed cost	High	Up to availability and congestion
μMV	Point-to-point	Medium	Medium operational cost, low fixed cost	Low/Medium	Up to availability
Walking	Point-to-point	No emissions	Free	Low	High
Subway	Fixed hubs and routes	Low	Low	Medium	High

tal Bikeshare in DC), Ms (e.g., Revel in DC), and FCMs. In addition to the design parameters introduced in the basic setting, we design the specific μM solution $M \in \{\text{ES,SB,M,FCM}\}$ and the μM fleet size $n_{\text{M,max}} \in \{0, 500, 1{,}000, \dots, 5{,}000\}$ vehicles (see Lemma 11.11). We study the joint deployment of μM solutions and AVs, and therefore consider the extended settings of 2022 and 2025.

S6 - Pricing: We show another extension of our framework to capture pricing strategies and infrastructure-contributing revenues (Section 11.3) in the 2022 setting. We consider AMoD service providers that choose from an exemplary set of prices $\{0.8, 1.2, 1.6, 2.4, 3.2\}$ (expressed in USD/mile) and public transit authorities choosing fare prices from the set $\{1.0, 2.0, 4.0, 6.0\}$ (expressed in USD per ride). Furthermore, we consider a municipality willing to cover 50% (just a particular choice) of the investment cost through the revenues of mobility services. (i.e., the revenue gained from travelers paying for the trips should at least be enough to cover 50% of the investment costs).

Basic setting

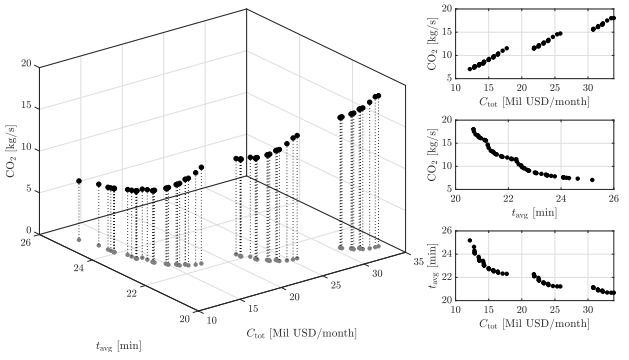
Fig. 163a reports the solution of the co-design problem through the antichain consisting of the total CO_2 emissions, average travel time, and total transportation cost. The design solutions are *rational* (and not comparable), since there exists no instance which simultaneously yields lower emissions, average travel time, and cost.

In the interest of clarity, we prefer a two-dimensional antichain representation, where emissions are included in the costs via a conversion factor of 40 USD/kg [252]. Note that this transformation preserves the monotonicity of the CDPI and therefore integrates in our framework. The two-dimensional antichain and the corresponding central authority's decisions are reported in Fig. 163b.

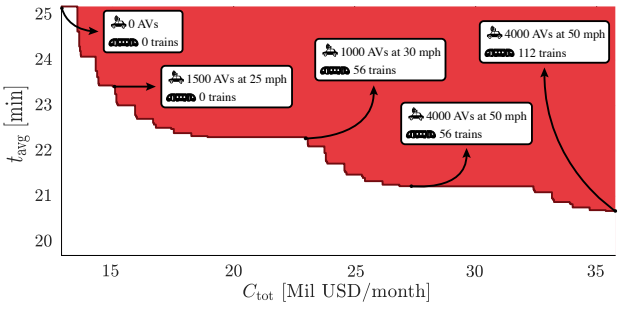
In general, as the municipality budget increases, the average travel time per trip required to satisfy the given demand decreases, reaching a minimum of about 20.7 min, with a monthly public expense of around 36 mil USD/month. This configuration corresponds to a fleet of 4,000 AVs able to drive at 50 mph, and to the doubling of the current MetroRail train fleet. Furthermore, the smallest

rational investment of 13 mil USD/month leads to a 22% higher average travel time, corresponding to the current situation, i.e., to a non-existent AVs fleet, and an unchanged subway infrastructure. Notably, an expense of 18 mil USD/month (50% lower than the highest rational investment) only increases the minimal required travel time by 8%, requiring a fleet of 3,000 AVs able to drive at 45 mph and no acquisition of trains. Conversely, an expense of 15 mil USD/month (just 2 mil USD/month higher than the minimal rational investment) provides a 2 min shorter travel time.

Finally, it is rational to improve the subway system starting from a budget of 23 mil USD/month, leading to a travel improvement of just 8%. This trend can be explained with the high train acquisition cost and increased operation costs, related to the reinforcement of the subway system. This phenomenon is expected to be even more marked for other cities, considering the moderate operation costs of the MetroRail subway system, due to its automation and related benefits.



(a) Left: Three-dimensional representation of antichain elements and their projection in the cost-time space. Right: Two-dimensional projections.

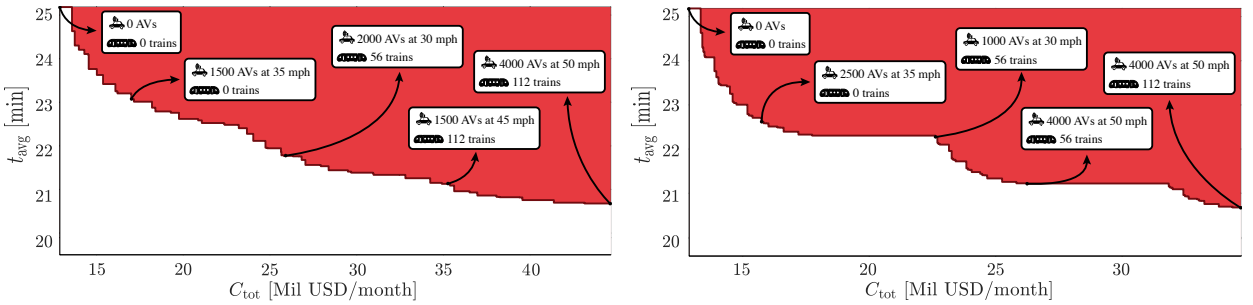


(b) Results for constant automation costs. We report the two-dimensional representation of the antichain elements: The pareto front is represented in dark red, and the upper set is the area above. We also report selected implementations corresponding to the highlighted antichain elements, in this case quantified in terms of achievable vehicle speed, AVs fleet size, and train fleet size.

Figure 163: Solution of the CDPI: Basic setting.

Speed-dependent automation costs

2022 We report the results in Fig. 164a. A comparison with our basic setting (cf. Fig. 163) confirms the trends concerning public expense. Indeed, a public expense of 26 mil USD/month (43% lower than the highest rational expense) only increases the average travel time by 5%, requiring a fleet of 2,000 AVs able to reach 30 mph and a subway reinforcement of 50%. Nevertheless, our comparison shows two substantial differences. First, the budget required for an average travel time of 13 min is 25% higher compared to S1. Second, the higher AV costs result in an average AVs fleet growth of 9%, an average velocity reduction of 15%, and



(a) Speed-dependent automation costs in 2022.

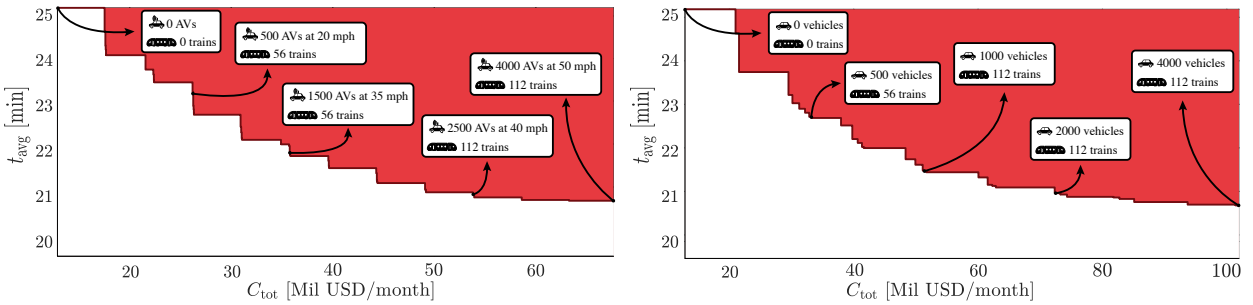
(b) Speed-dependent automation costs in 2025.

Figure 164: Results for the speed-dependent automation costs.

an average train fleet growth of 14%. The latter suggests a shift towards poorer AVs performance in favor of fleets reinforcements.

2025 The maximal rational budget is 23% lower than in the case of immediate deployment (Fig. 164b). Further, the reduction in autonomy costs incentifies the acquisition of more performant AVs, increasing the average vehicle speed by 14%. Hence, AVs and train fleets are 10% and 13% smaller.

High automation costs analysis



(a) Results for large automation costs.

(b) Results for the MoD case.

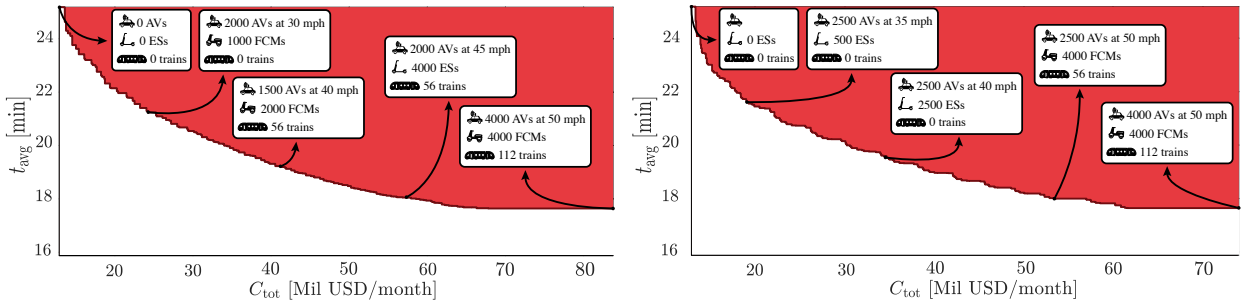
Figure 165: Results for large automation costs, and for the MoD case.

Fig. 165a shows the results for high automation costs. First, we observe a substantial shift towards larger train fleet sizes (65 % larger than in S1) and smaller AVs fleets (55 % smaller than in S1). Second, minimizing the average travel time entails an expense of approximately 68 Mil USD/month, basically doubling the investments observed in the basic setting.

MoD setting

We summarize the results for the MoD scenario in Fig. 165b. In particular, by comparing the MoD case with the 2025 setting, we can notice the game-changing properties that AVs introduce in the mobility ecosystem. In particular, the average train fleet size and the average vehicle fleet sizes increase by 130% and 66%, suggesting a clear transition in investments from public transit to AVs, and testifies to the interest in AMoD systems developed in the past years.

Impact of new transportation modes



(a) Impact of micromobility in 2022.

(b) Impact of micromobility in 2025.

Figure 166: Results for the impact of micromobility.

To assess the impact of μM solutions, we compare the arising design solutions, reported in Fig. 166, with their counterpart in S2 (cf. Fig. 164).

2022 Fig. 166a, together with Fig. 164a, demonstrates an overall benefit from μM solutions. For instance, the most time-efficient solution in S2 yields an average travel time of 20.7 min at an expense of 45 mil USD/month. The deployment of μM solutions lowers the average travel time achievable with the same expense by 10% (18.8 min) and allows for even lower average travel times, with a time-efficient solution of 17.6 min at an investment plan of 84 mil USD/month. Overall, the average AVs fleet size and the average train fleet size are 35% and 6% smaller, in favor of an average μM fleet of 2,280 μMVs.

2025 Fig. 166b, together with Fig. 164b, shows that the benefit of μM solutions is less marked than in 2022. For instance, an expense of 35 mil USD/month (same as the maximal expense in Fig. 164b) results in an average travel time of 19.5 min, i.e., only 6% lower than in the case without μM. Furthermore, we observe an average AVs fleet size enlargement of 17%, and an average train fleet size reduction of 27%. Finally, the comparison with the 2022 case highlights a μMVs fleet reduction of 23%, which suggests the comparative advantage of AVs in the future. Indeed,

the stronger the reduction of the cost of automation, the more investments in AVs are rational. The benefits of employing μM solutions could therefore just be temporary, and gradually vanish as the costs of automation of AVs decrease.

Pricing

We report the results in Fig. 167. In particular, we report the Pareto front between system performance and emissions (the two resources of the considered MDPI), as well as design choices for selected Pareto-optimal solutions, now including prices for AMoD and public transit services. We report three key observations. First, the most performing solution (which satisfies the cost-contributing constraint) features the usage of 4,000 AVs able to drive at 45 mph and an increment of 50% of the public transit fleets. The large usage of AVs is not only due to their efficiency, but also to the low price of 0.8 USD/mile. While this choice does not fully exploit the action space of the municipality (one could have larger fleets, more performant AVs, and more trains), it is the last one for which the weighted costs do not exceed the revenue. Second, we observe fewer solutions featuring an augmented train fleet, mainly because of the related onerous investments related (i.e., more AVs, not necessarily very performing, can bridge the system performance gap). Finally, comparing the emissions in Fig. 167 and in Fig. 163a suggests that bounding the allowed mobility system costs also prevents design options which are more pollutant from being chosen.

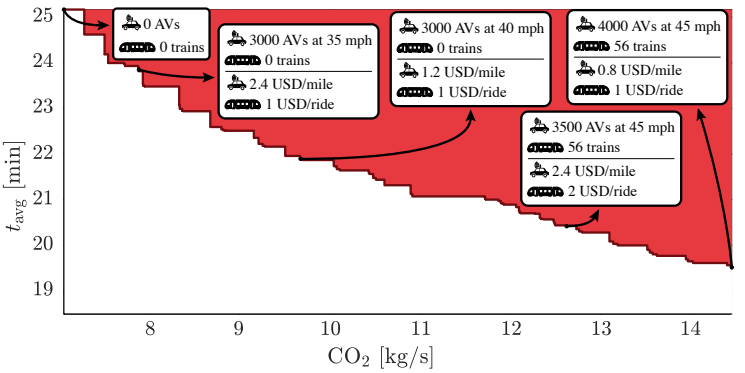


Figure 167: Pricing and revenues case study. The Pareto front is in terms of system performance (average travel time) and produced externalities.

Discussion

First, the presented case studies showcase the ability of our framework to extract the set of rational design strategies for a future mobility system, including AVs, μMVs, and public transit. This way, stakeholders such as mobility providers, transportation authorities, and policy makers can get transparent and interpretable insights on the impact of future interventions, inducing further reflection on

this complex socio-technical problem. Note that this kind of results is only one of the many factors affecting negotiations when interacting with stakeholders. Second, we perform a sensitivity analysis through the variation of autonomy cost structures, and show the capacity of our framework to capture various models. On the one hand, this reveals a clear transition from small fleets of fast AVs (in the case of low autonomy costs) to large fleets of slow AVs (in the case of high autonomy costs). On the other hand, our studies highlight that investments in the subway infrastructure are rational only when large budgets are available. Indeed, the high train acquisition and operation costs lead to a comparative advantage of AV-based mobility. Finally, our case studies suggest that the deployment of μM solutions is rational primarily on a short-term horizon: The lowering of automation costs could eventually make AVs the predominant actor in the future of urban mobility.

From autonomy to mobility via compositionality

12

The purpose of abstraction is not to be vague, but to create a new semantic level in which one can be absolutely precise.
—Edsger Dijkstra[20]

Following the principle "your system is just a component in somebody else's system", in this chapter we embed the complex autonomy co-design models developed in Chapter 10 in the co-design models of a mobility system developed in Chapter 11 First, we will present the updated models (Section 12.1), and then we will show sample results for Washington D.C., USA (Section 12.2).

12.1 Models

In the mobility co-design problem we started from a simple model of an AV. We assumed a single vehicle to be represented by a relationship between achievable speed, fix and operational costs. We can now do better, by embedding a more sophisticated model of autonomy, as introduced in Chapter 10. There are multiple ways to perform this extension.

A first, naive way, could be to consider the design problem of an AV as in Section 10.2, and forget all the functionalities/resources but achievable speed and fix/operational costs. In a sense, this would fit the sophisticated model of autonomy to the mobility needs. However, we might miss important aspects of autonomy, such as the extra power consumption due to the autonomy stack.

Another way, could be to take the AV design problem in the mobility co-design problem, and enhance it with functionalities and resources from the design problem in Section 10.2. For instance, one can take into account the power consumption of the entire autonomy stack of one AV, as well as the related emissions produced. At the mobility system level, one can then account for them, in addition to the emissions already considered before (e.g., operations of the vehicles and subway system). This can be visualized as in Fig. 168, where we enhanced the last mobility co-design problem discussed in Chapter 11.

In this context, another interesting study would include diverse fleets, performing different tasks. For instance, think of a city like San Francisco, featuring both flat, and hilly topologies. To achieve this, one could modify the mobility model, to account for heterogeneous fleets of AVs, and frame the optimization problem as a search for optimal fleet compositions given specific demand patterns [253]. Nevertheless, this is beyong the scope of this chapter, and we leave such problems

[20] Dijkstra was a Dutch computer scientist, recipient of the 1972 Turing Award for his contributions to structured programming languages.

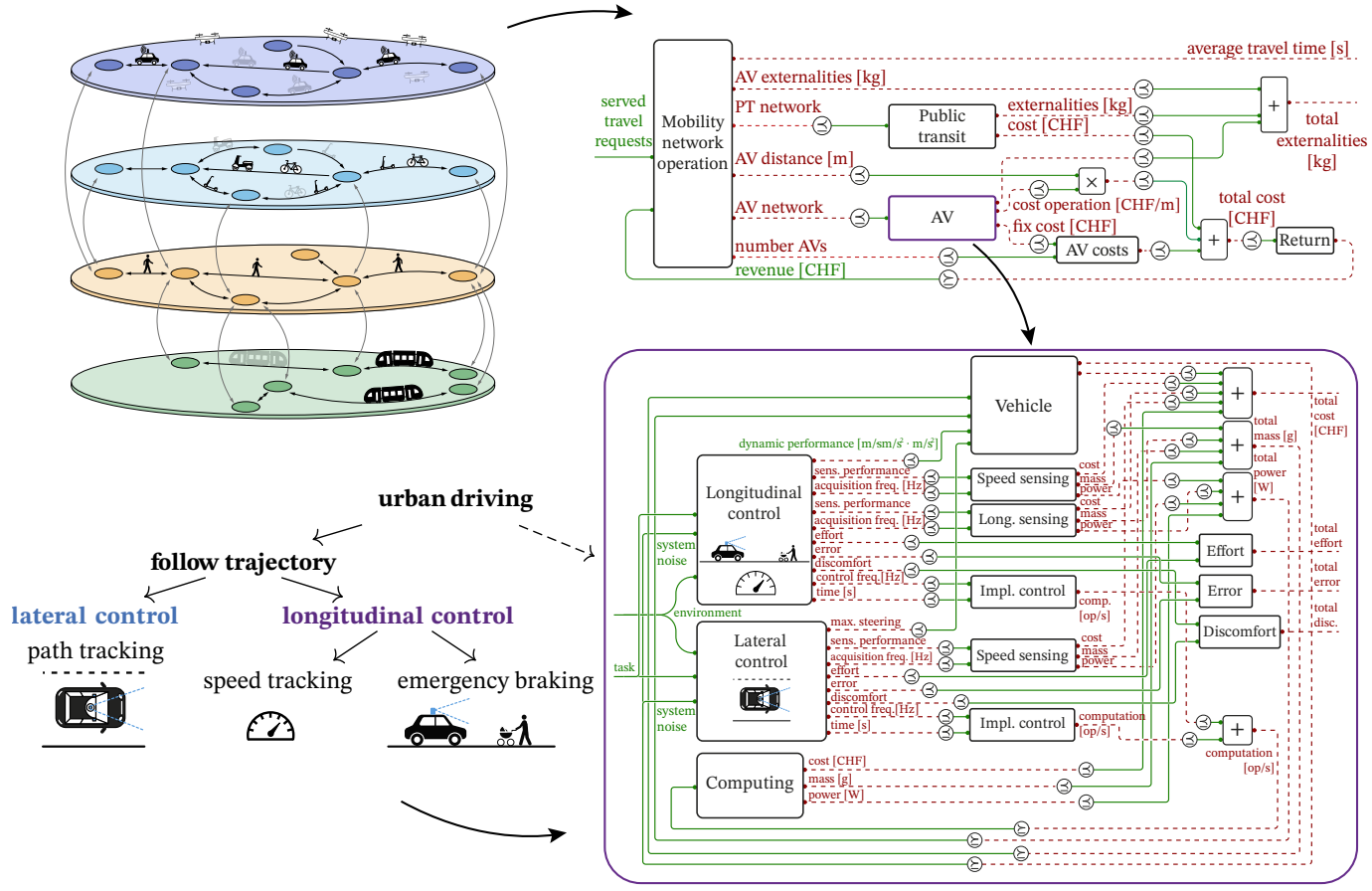


Figure 168: Compositionality of the co-design framework at work. We can take the co-design problem for the autonomy stack of an AV and embed it into the co-design problem of a mobility system.

to future investigations.

12.2 Results

Results can be produced in the very same way as we did so far. For instance, consider merging the case studies in Section 10.2, and the pricing case study in Chapter 11. Here, we can query the co-design problem for the mobility system with a specific travel demand for Washington, D.C., as well as task specifications for the autonomy stack. The resources to be minimized are, again, the average travel time in the mobility system, and the emissions produced by the mobility options. In a sense, this is a trade-off between the performance of the system and its sustainability.

The resulting Pareto front of solutions is reported in Fig. 169. Notably, one obtains the same kind of trade-offs presented in Chapter 11, but now eliciting more insights about the actual design of a single AV. Indeed, now not only we are

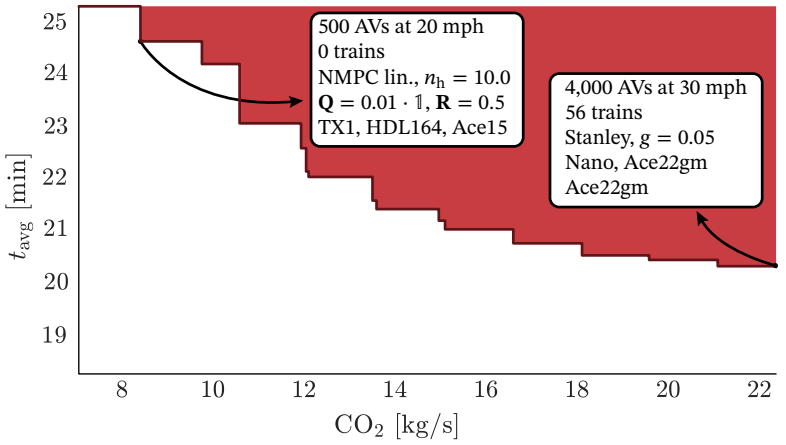


Figure 169: Pareto front for the optimal design of a mobiltiy system featuring AVs and public transit. The pareto front is in terms of system performance (average travel time) and produced externalities.

able to identify fleet sizes, speeds, and prices for every point on the antichain of optimal solutions, but we can also specifically identify details about the autonomy stack of a single AV composing the AMoD fleet. For instance, the first solution on the left consists in acquiring no extra trains and just 500 AVs, capable of driving at 20 mph. The AVs in this fleet will leverage a PID controller with certain parameters for longitudinal control, as well as a NMPC controller with other parameters for lateral control. They will be equipped with particular computing units and sensors. For the solution depicted on the right, instead, 4,000 AVs and 56 extra trains will be bought. The vehicles will be able to drive at 30 mph, featuring different controllers, computing units, and sensors. Here, the pricing strategy for the rides is 0.8 USD/mile for the AMoD service, and 1.0 USD/ride for public transit.

Open challenges and conclusions | Part C

Explicitly accounting for strategic interactions

13

Human beings are born solitary, but everywhere they are in chains – daisy chains – of interactivity. Social actions are makeshift forms, often courageous, sometimes ridiculous, always strange. And in a way, every social action is a negotiation, a compromise between "his", "her" or "their" wish and yours.

—Andy Warhol[21]

13.1 Need for co-design games

Up to this point, our discussion has revolved around intricate engineering design challenges and the ways in which our co-design theory has proven invaluable in the pursuit of optimal design solutions. These solutions often involve navigating a complex landscape of competing objectives. Within this co-design framework, we have emphasized its *collaborative* and *decentralized* nature. However, it is essential to acknowledge a fundamental assumption that underpinned our previous applications: the presumption that while multiple designers may model different components within a complex architecture, they all share a common interest in minimizing a predefined set of objectives. This assumption holds great significance and finds numerous practical applications in the field of engineering.

Consider, for instance, a team of engineers working on distinct facets of the autonomy stack of an AV. Their collective goal is to harmonize their efforts and discover the most efficient design policies for each component. This cooperative approach aligns well with the co-design concept: each expert can specify particular design problems within a complex diagram, and their expertises can be combined. However, when we shift our focus to a co-design problem within the context of a mobility system, the situation becomes more intricate. In this scenario, we have examined the issue from a municipal standpoint, but the stakeholders involved are multiple, each possessing diverse and, at times, conflicting objectives. These stakeholders encompass policy makers, mobility service providers, and customers, among others.

This observation underscores a critical point: the co-design theory we have presented is suitable for a specific subset of engineering design problems. Still, it necessitates further refinement to accomodate other categories of challenges involving multiple stakeholders engaged in *strategic design interactions*. To tackle such problems effectively, we must extend and modify the co-design theory to

[21] Warhol was an American visual artist, film director, producer, and leading figure in the pop art movement.

accomodate multiple participants with both shared and conflicting objectives. In this context, the conventional notion of a "problem" and a "solution" must undergo adaptation. Instead of merely optimizing outcomes, we must now consider a quest for *equilibria* – states in which no player has an incentive to alter their strategies.

The formal toolbox to address these types of challenges is rooted in game theory. However, it is important to note that existing methodologies cannot be readily applied "off-the-shelf" to address the specific issues highlighted above. These "new" games possess unique properties that set them apart.

Games with a co-design structure In this thesis, we recognized the versatility and expressivity of co-design problems. In the game theoretic setting, different players can take decisions which influence one or more design problems. One can essentially see two kinds of settings for interactions (the second generalizes the first one):

- Given a co-design diagram, players compete to design a particular component;
- Given a co-design diagram, players can take decisions which influence one or more design problems, and the payoffs of the players depend on (a subset) of the functionalities/resources of those design problems;

Handling these cases requires a theory of co-design games, which is yet to be developed. In the following, we present two preliminary steps took in this direction. First, we present the notion of games with partially ordered preferences, applied to multi-agent motion planning. Second, we present mobility games, to deal with the sequential nature of certain stakeholders' interactions. In both cases, we provide a synoptic description of the contributions. For details, please refer to the reported references.

13.2 Games with partially ordered payoffs

We have seen how posets play an important role in describing trade-offs in co-design. However, in game theory, their utilization has been somewhat limited, and the development of a comprehensive theory for games with partially ordered payoffs is missing. To address this gap, we have taken an initial step in our work on *posetal games* [254].

In this line of work, we delve into the study of games where players express preferences that are partially ordered with respect to specific metrics of interest. To illustrate this, consider the scenario of AVs navigating through urban environments (e.g., at intersections), where they must make trajectory choices based on their preferences over performance metrics. Fig. 170 provides a simplified example to visualize this concept within the context of urban driving. In this

scenario, every vehicle prioritizes collision avoidance as its top objective. Subsequently, an ambulance emphasizes minimization of travel time as its primary goal, while adhering strictly to traffic rules and ensuring a comfortable ride are secondary considerations. In contract, an elderly driver, while also desiring collision avoidance, places a higher value on obeying traffic regulations and ensuring a comfortable ride.

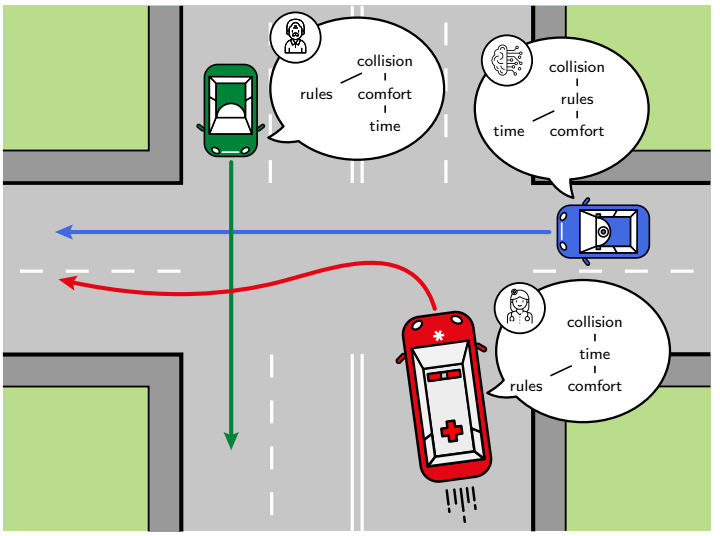


Figure 170: Cartoon representation of posetal games applied to AVs navigating an intersection.

The preferences regarding performance metrics naturally translate into preferences over the actions taken by the players. For instance, consider the scenario outlined in Fig. 171. Here, the AV0s top priority is to minimize its collision cost, which typically involves mitigating kinetic energy transfers during interactions. This takes precedence over the risk of a sever violation of traffic rules, such as cumulative time spent beyond the double lines, and the potential minimum clearance cost, related to violations of safety distance norms. In this specific context, trajectories b, c, and d, are preferred over trajectory a. Even though b and c are two different trajectories, they evaluate to be *indifferent* since they lead to the *same* outcomes. Both share the same collision, area violation, and clearance values. Finally, trajectory d has a worse area violation, but a better clearance than b, c. It is therefore *uncomparable* with respect to b, c, and constitutes an antichain with them.

Posetal games In this context, a posetal game is specified as follows:

- There is a finite set of players;
- Each player possesses a decision space.
- Given the action profile of all players, an outcome of the game can be deter-

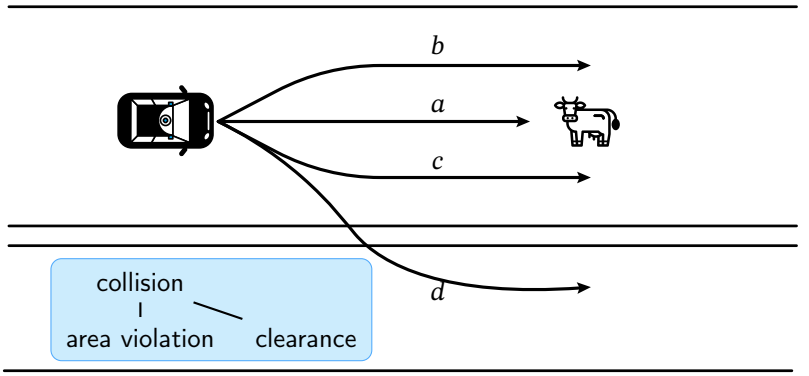


Figure 171: Example of priorities over metrics, which induce priorities over trajectories.

mined (e.g., what happens in an intersection if all players follow their chosen trajectories);

- Each player specifies a posetal preference over their actions.

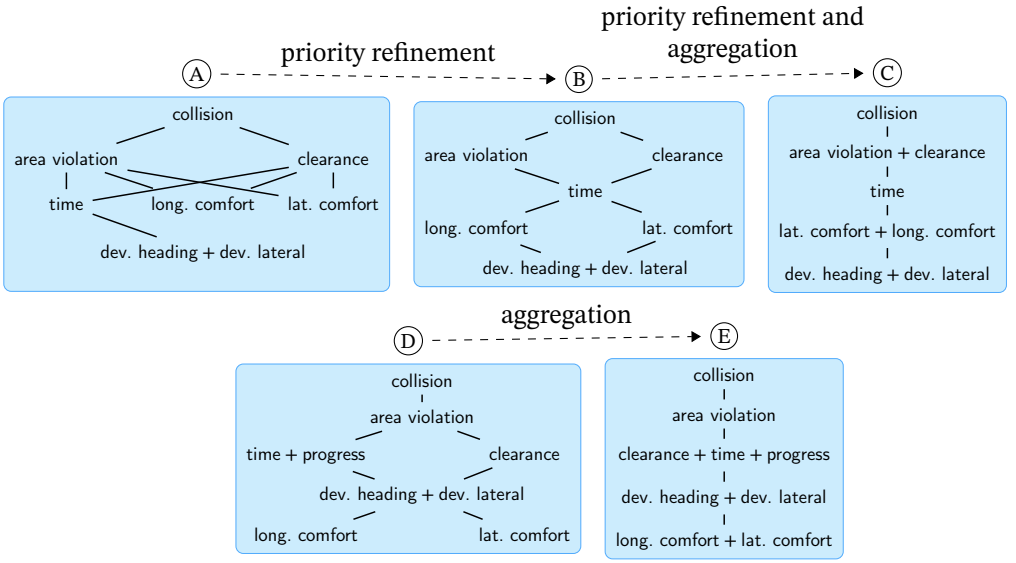
Results

Formally, we analyzed the presented class of games, providing the following theoretical contributions:

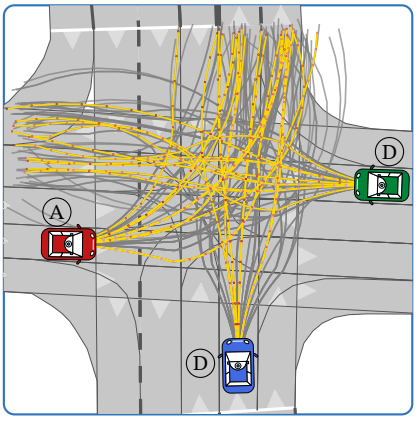
- We provide two sufficient conditions for the existence of a pure Nash Equilibria in posetal games with finite action sets. Interestingly, this depends on properties of the single metrics over which the agents specify a preference, but also on the combined preference structure of the players.
- We show that the set of equilibria of such a game is intimately related to the operations one performs on the preference structures. Particularly, any refining operation of a player's preference refines the set of equilibria.

To provide some intuition, consider Fig. 172.

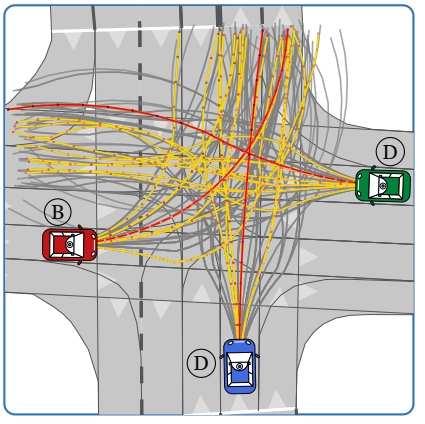
Here, we consider an intersection with three different vehicles, each which can choose between various trajectories. For different preference structures (Fig. 172a), we can compute the equilibria of the game. From left to right, we observe the *same* game played with *refined* preferences. Different kind of equilibria (refer to the paper for technicalities) are depicted using different colors. Interestingly, one of of the demonstrated properties is that refining the preferences in the game, actually refines (shrinks) the equilibria of that game. This can be visualized observing the plots from left to right.



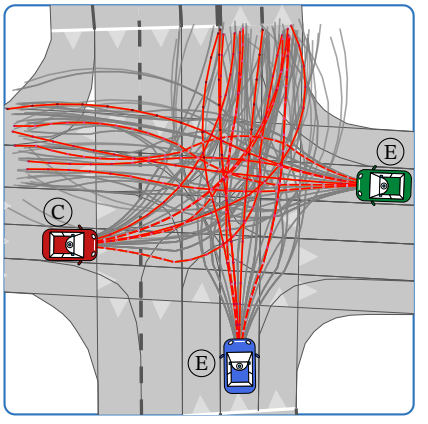
(a) Preferences used in the case study and their refinement.



(b) Preference in Fig. 172a yields 28 weak Nash equilibria, no strong Nash equilibria, and 17 admissible Nash equilibria.



(c) Preference in Fig. 172a yields 16 weak Nash equilibria, 1 strong Nash equilibria, and 10 admissible Nash equilibria.



(d) Preference in Fig. 172a yields no weak Nash equilibria, 7 strong Nash equilibria, and 3 admissible Nash equilibria.

Figure 172: Sample solution of a series of posetal games, in the setting of an intersection with three AVs.

13.3 Simultaneous and sequential decisions

In the problems considered so far, stakeholders engage in interactions which can occur simultaneously or sequentially, often exhibiting intricate and evolving patterns, depending on the specific time horizon being considered. To illustrate this dynamic, let's examine the interactions within an intermodal mobility system, involving key actors such as a municipality, a public transit agency, private mobility service providers, and the customers they serve (Fig. 173). The nature of these interactions can very significantly based on the time frame under consideration,

whether it be on a daily, monthly, yearly, or other temporal basis.

For instance, on a daily cycle customers make choices about where they need to go, when they need to travel, and how they plan to reach their destinations. Simultaneously, mobility service providers, operating within this daily context, are tasked with making real-time decisions, such as dynamically adjusting pricing structures (consider, for instance, the surge pricing during rainy weather in Singapore) and implementing operational strategies. These strategies might involve determining which vehicles should halt for recharging or maintenance to ensure smooth service delivery despite fluctuations in demand.

Conversely, when we extend our viewpoint to an annual time frame, the dynamics of interaction among stakeholders undergo a significant transformation. Customers, instead of making daily choices, might opt for longer-term commitments, such as signing up for annual travel plans, or subscription services which offer various travel benefits. Mobility service providers, instead, have the opportunity to strategically adjust their fleets and services to align with anticipated long-term demand patterns, customer preferences, and changing market conditions. Additionally, municipalities may come into play at this level, modifying regulations, or introducing tax incentives and policies which have a more profound impact on the broader mobility landscape.

Mobility games

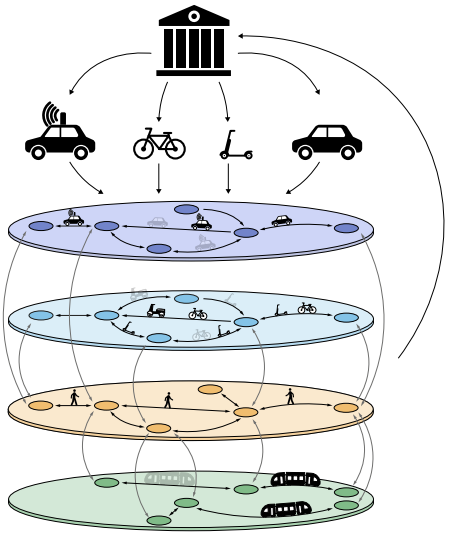


Figure 173: Scheme of interactions for sequential mobility games.

In this line of work, we leverage game theory to formulate and solve problems involving strategic (both simultaneous and sequential) interactions of the stakeholders of mobility systems [255]–[257]. At the heart of our methodology is the formulation of a sequential game which unfolds in a carefully orchestrated manner. In this game, the municipality assumes the leading role, making pivotal decisions that set the stage for the entire system. These decisions may encompass choices regarding taxes, incentives, pricing structures for public transit, or the establishment of regulations which shape the mobility landscape. Subsequently, all mobility service providers operating within the jurisdiction respond strategically. They must navigate through a complex decision space, making choices about their vehicle fleets, service offerings, and pricing strategies. Finally, customers, as the ultimate beneficiaries and decision-makers within this ecosystem, react to the choices made by the municipality and mobility providers.

It is crucial to highlight that each stakeholder within this strategic game is driven by distinct objectives and priorities, which add layers of complexity to the decision-making process. For instance, the municipality's primary concern may revolve around minimizing externalities, such as emissions, or optimizing the overall social welfare and performance of the mobility system. In contract, mobility service providers are inherently profit-driven, focusing on strategies that maximize their

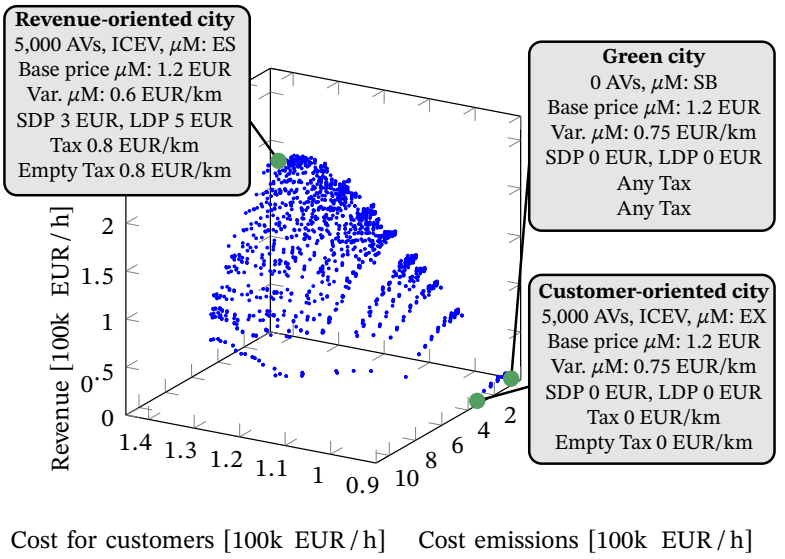


Figure 174: Equilibria of the game with respect to cost for customers, cost of emissions, and public revenue. Each point is a Nash equilibrium of the simultaneous game between mobility service providers. The equilibrium of the sequential game directly results from the weights of the three metrics in municipality's social welfare.

return on investment and competitiveness within the market.

Sample results for Berlin, Germany Here, we present a glimpse of the results obtained from one the case studies we conducted, focusing on the dynamic urban landscape of Berlin, Germany. The stakeholders feature a municipality, an AMoD operator, a micromobility operator, and a standard taxi company.

The municipal authority can determine pricing structures for public transit, as well as implement taxes on the AMoD operator. Meanwhile mobility providers, including the AMoD operator, micromobility service provider, and the traditional taxi company, have their unique spheres of influence. They make strategic decisions regarding the specifics of their services, such as determining fleet sizes and types. For instance, they can choose the combustion type of vehicles, or choosing among options like e-scooters, shared bikes, and electric mopeds for micrmobility. These providers also set pricing strategies which cater to the demands of the market and customer preferences.

Our analysis takes into account a rich dataset of 130,000 realistic travel requests, reflecting the preferences of customers who aim to minimize their personal costs, which include both fares and the value of their time. Furthermore, we integrate the actual road network of Berlin, factoring in calibrated congestion effects to ensure the realism and accuracy of our simulations.

Given the framework, we can for instance compute the Nash equilibria of the simultaneous game between mobility providers, and report them with respect to three metrics defining social welfare: cost for the customers, emissions, and public revenue. Given this plot, one can then select specific profiles for the municipality,

ending in one particular equilibrium. Then, for each equilibrium, we can provide the corresponding strategies for all the players involved. There are numerous trade-offs. For instance, a municipality minimizing emissions incurs in emission costs of 4,711 EUR/h, and should make public transport free and ban AMoD vehicles. A customer-centric city also opts for free public transport, and does not introduce taxes, at the price of no public revenue.

Extending modeling capabilities and solution algorithms

14

I've always been more interested in the future than in the past.
— Grace Hopper[22]

We have shown how the current co-design framework helps in solving complex co-design problems in different fields. In the following, we briefly mention a sample of the extensions to be developed in the future. In particular, we focus on new modeling techniques, solution techniques, and new applications.

14.1 Extending modeling techniques

Spatio-temporal resources So far, we have not attributed a time-related aspect to resources and functionalities. For instance, imagine a factory, consisting of many different machines, transforming materials, with performance metrics which depend on time, kind of job, etc. In this context, one could think about expressing dynamic feasibility relations, between temporal sequences of functionalities and resources. This would extend the notion of query as well. For instance, one could be interested in the minimal sequences of resources, such that a certain functionality is achieved by a certain time instant (e.g., when considering a particular order that has to be shipped out). Alternatively, one might care about the frequency at which a functionality is produced. Similar arguments can be made for "spatiality".

Connection to "classic" optimization problems In this thesis, we have explored the concept of modeling specific types of optimization problems, such as convex optimization problems, as design problems. This approach holds significant promise as it enables us to integrate established optimization techniques into the co-design framework, facilitating their interconnection. Looking forward, our aim is to delve deeper into this area of study. We plan to examine a range of "classic" optimization problems and their co-design formulations, with a keen interest in understanding how these formulations might influence solution techniques.

14.2 Extending solution techniques

Computation-aware solutions Up to this point, our approach has involved creating a specific co-design diagram, determining how to populate each individual design problem within it, and then addressing a particular query. In the

[22] Hopper was an American computer scientist, mathematician, and USA Navy rear admiral.

future, it would be interesting to conduct this process with a computation-aware approach. This would involve designing algorithms that autonomously determine which model to sample based on the existing partial solution. Such an approach holds particular significance for design problems where obtaining specific implementations is resource-intensive, as seen in simulation- or optimization-driven design challenges. This research direction also encompasses the enhancement of specific solutions through the utilization of techniques like differentiable simulation.

Negative design problems In the past, we have established a framework that deals with negative outcomes in category theory, in contrast to the existing positive results, as documented in our previous work [258]. One noteworthy application of this framework pertains to negative design problems, which are essentially situations where infeasibility relations exist, much like their positive counterparts in design problems. An intriguing avenue of exploration lies in leveraging these negative design problems for the creation of more efficient computational strategies. This could involve harnessing both positive and negative results to enhance the efficiency of computations.

14.3 New applications

Throughout this thesis, our primary focus has centered on the co-design of embodied intelligence and mobility systems. In these specific applications, our aim is to expand upon the existing models that underpin them. For instance, in the context of autonomy design, we are keen on delving deeper into the intricate relationship between perception and decision-making, potentially incorporating planning considerations into our co-design models. However, it's important to note that these applications represent just a subset of the domains that stand to benefit from our approach. Looking ahead, we find it intriguing to explore novel areas. These may encompass various fields, such as space systems (with deep historical ties to traditional engineering design literature), broader infrastructure systems (for instance, within the realms of energy and mobility ecosystems), and even the automotive sector (including the design of hybrid-electric engines), among others.

Conclusions 15

A ship in harbor is safe, but that is not what ships are built for.
—John A. Shedd[23]

Within this thesis, we have introduced the challenges associated with the design of complex systems, and we have underscored the pressing need for novel tools that can effectively frame and tackle these intricate design problems. In particular, we have introduced a monotone theory of co-design, a framework with a well-formalized structure rooted in the language of category theory. This framework boasts a range of compelling attributes: it is formal, compositional, modular, and computationally manageable, all while remaining user-friendly and intellectually tractable.

Our exploration has extended to the application of this framework in the context of co-designing embodied intelligent systems, spanning from the individual domain of autonomous vehicles to the orchestration of entire mobility systems. Through a series of case studies, we have showcased the framework's capacity to yield actionable insights for designers. Furthermore, we have furnished a structured procedure that facilitates reasoning about the intricacies of the design process.

While our findings are certainly promising and exciting, we acknowledge that there is room for improvement in the methodology. This realization has led us to identify a plethora of intriguing research avenues that deserve exploration in the future.

[23] Shedd was an American writer.

Back matter | Part D

Proofs 16

16.1 Proofs related to Part A

Proof of Lemma 2.16. Consider partial orders $\mathbf{P}, \mathbf{Q}$ and any three maps $f, g, h : \mathbf{P} \to \mathbf{Q}$. Clearly $f \preceq_{\mathbf{Q}^\mathbf{P}} f$. Furthermore, if $f \preceq_{\mathbf{Q}^\mathbf{P}} g$ and $g \preceq_{\mathbf{Q}^\mathbf{P}} h$ (i.e., $f(p) \preceq_\mathbf{Q} g(p)$ and $g(p) \preceq_\mathbf{Q} h(p)$, $\forall p \in \mathbf{P}$), then $f(p) \preceq_\mathbf{Q} h(p)$ $\forall p \in \mathbf{P}$, implying $f \preceq_{\mathbf{Q}^\mathbf{P}} h$. Finally, if $f \preceq_{\mathbf{Q}^\mathbf{P}} g$ and $g \preceq_{\mathbf{Q}^\mathbf{P}} f$, one has $f = g$. □

Proof of Lemma 2.32. Consider $p_1, p_2 \in \mathbf{P}$, $q_1, q_2 \in \mathbf{Q}$. By assuming that f and g are monotone, we have

$$\frac{p_1 \preceq_\mathbf{P} p_2}{f(p_1) \preceq_\mathbf{Q} f(p_2)}$$

and

$$\frac{q_1 \preceq_\mathbf{Q} q_2}{g(q_1) \preceq_\mathbf{R} g(q_2)}.$$

By substituting the above in the map composition formula, we have

$$\frac{p_1 \preceq_\mathbf{P} p_2}{(f \fatsemi g)(p_1) \preceq_\mathbf{R} (f \fatsemi g)(p_2)},$$

which is the monotonicity condition for the composite map $(f \fatsemi g)$. □

Proof of Lemma 2.37. Assume that y and z are both least upper bounds of $\mathbf{S} \subseteq \mathbf{P}$. In other words, one knows $x \preceq_\mathbf{P} y$ and $x \preceq_\mathbf{P} z$ for all $x \in \mathbf{S}$. However, one also has $y \preceq_\mathbf{P} z$ and $z \preceq_\mathbf{P} y$ (from y, z assumed to be both least upper bounds). Because of antisymmetry, this implies $y = z$ and proves the uniqueness of least upper bounds in a poset. □

Proof of Lemma 2.46. Suppose $y \in \uparrow \mathbf{S}$ and $z \in \mathbf{P}$, and suppose $y \preceq_\mathbf{P} z$. By definition there exists a x such that $x \preceq_\mathbf{P} y$, meaning that $x \preceq_\mathbf{P} z$. Thus, $z \in \uparrow \mathbf{S}$, as was to be shown. □

Proof of Lemma 2.47. Consider the posets $\langle \mathsf{Pow}\,\mathbf{P}, \subseteq \rangle$ and $\langle \mathsf{USets}\,\mathbf{P}, \supseteq \rangle$, and two sets of sets $\mathbf{S_1}, \mathbf{S_2} \in \mathsf{Pow}\,\mathbf{P}$. It is clear that given $\mathbf{S_1} \subseteq \mathbf{S_2}$, we have

$$\{y \in \mathbf{P} \mid \exists x \in \mathbf{S_1} : x \preceq_\mathbf{P} y\} \subseteq \{y \in \mathbf{P} \mid \exists x \in \mathbf{S_2} : x \preceq_\mathbf{P} y\}.$$

Therefore, $\uparrow \mathbf{S_1} \subseteq \uparrow \mathbf{S_2}$, satisfying the antitone map property for $\uparrow$. □

Proof of Lemma 2.49. Consider the posets $\langle \mathsf{Pow}\,\mathbf{P}, \subseteq \rangle$ and $\langle \mathsf{LSets}\,\mathbf{P}, \subseteq \rangle$, and let $\mathbf{S_1}, \mathbf{S_2} \in \mathsf{Pow}\,\mathbf{P}$. It is clear

that given $\mathbf{S_1} \subseteq \mathbf{S_2}$, we have

$$\{y \in \mathbf{P} \mid \exists x \in \mathbf{S_1} : y \preceq_{\mathbf{P}} x\} \subseteq \{y \in \mathbf{P} \mid \exists x \in \mathbf{S_2} : y \preceq_{\mathbf{P}} x\}.$$

Therefore, $\downarrow \mathbf{S_1} \subseteq \downarrow \mathbf{S_2}$, satisfying the monotonicity property for $\downarrow$. □

Proof of Lemma 2.50. Fix an element $a \in \mathbf{A}$. From $\uparrow \mathbf{A} = \uparrow \mathbf{B}$ we know that in particular $\mathbf{A} \subseteq \uparrow \mathbf{B}$. This means that for our fixed $a \in \mathbf{A}$ there exists $b \in \mathbf{B}$ such that $b \preceq a$. From $\uparrow \mathbf{A} = \uparrow \mathbf{B}$ it also follows that $\mathbf{B} \subseteq \uparrow \mathbf{A}$, so to the $b \in \mathbf{B}$ given above, there exists $a' \in \mathbf{A}$ such that $a' \preceq b$. In total, we have $a' \preceq b \preceq a$, and since $\mathbf{A}$ is an antichain, we must have $a' = a$. This implies that $a' = b = a$. In particular, we have $a \in \mathbf{B}$.

The above shows that $\mathbf{A} \subseteq \mathbf{B}$. To show $\mathbf{B} \subseteq \mathbf{A}$, we can fix any $b \in \mathbf{B}$ and repeat the above argumentation, now with the roles of $\mathbf{A}$ and $\mathbf{B}$ exchanged. □

Proof of Lemma 2.58. Consider the poset $U\mathbf{P} = \langle \mathsf{USets}\,\mathbf{P}, \preceq_{U\mathbf{P}} \rangle$ and $\mathbf{A}, \mathbf{B} \in \mathsf{USets}\,\mathbf{P}$.

First, we need to show that $\mathbf{A} \cap \mathbf{B} \in \mathsf{USets}\,\mathbf{P}$. To this extent, we need to show that, for all $a \in \mathbf{A} \cap \mathbf{B}$ and for all $a \preceq_{\mathbf{P}} b$, it holds $b \in \mathbf{A} \cap \mathbf{B}$. We have $\mathbf{A} \in \mathsf{USets}\,\mathbf{P}$ and $\mathbf{B} \in \mathsf{USets}\,\mathbf{P}$, meaning that by definition, if $a \in \mathbf{A} \cap \mathbf{B}$, we have $a \in \mathbf{A} \wedge a \in \mathbf{B}$. It follows that $b \in \mathsf{USets}\,\mathbf{P}$ and $b \in \mathsf{USets}\,\mathbf{Q}$. Therefore, $b \in \mathsf{USets}\,\mathbf{P} \cap \mathbf{B}$ and, thus, $\mathbf{A} \cap \mathbf{B} \in \mathsf{USets}\,\mathbf{P}$. Furthermore, we need to show that $\mathbf{A} \cap \mathbf{B}$ is the least upper bound of $\mathbf{A}$ and $\mathbf{B}$. Let $\mathbf{C} \in \mathsf{USets}\,\mathbf{P}$ such that

$$\mathbf{A} \preceq_{U\mathbf{P}} \mathbf{C} \preceq_{U\mathbf{P}} (\mathbf{A} \cap \mathbf{B}) \iff \mathbf{A} \supseteq \mathbf{C} \supseteq (\mathbf{A} \cap \mathbf{B})$$

and

$$\mathbf{B} \preceq_{U\mathbf{P}} \mathbf{C} \preceq_{U\mathbf{P}} (\mathbf{A} \cap \mathbf{B}) \iff \mathbf{B} \supseteq \mathbf{C} \supseteq (\mathbf{A} \cap \mathbf{B}).$$

Using the fact that intersection preserves inclusions, we have

$$\begin{aligned} & (\mathbf{A} \cap \mathbf{B}) \supseteq (\mathbf{C} \cap \mathbf{C}) \supseteq (\mathbf{A} \cap \mathbf{B}) \\ \iff & (\mathbf{A} \cap \mathbf{B}) \supseteq \mathbf{C} \supseteq (\mathbf{A} \cap \mathbf{B}) \\ \iff & \mathbf{C} = (\mathbf{A} \cap \mathbf{B}). \end{aligned}$$

Therefore, $\mathbf{A} \cap \mathbf{B}$ is the least upper bound of $\mathbf{A}$ and $\mathbf{B}$.

Second, we need to show that $\mathbf{A} \cup \mathbf{B} \in \mathsf{USets}\,\mathbf{P}$, meaning that for all $a \in \mathbf{A} \cup \mathbf{B}$, $a \preceq_{\mathbf{P}} b$ implies $b \in \mathbf{A} \cup \mathbf{B}$. We have $\mathbf{A} \in \mathsf{USets}\,\mathbf{P}$ and $\mathbf{B} \in \mathsf{USets}\,\mathbf{P}$, meaning that by definition, if $a \in \mathbf{A} \cup \mathbf{B}$, we have either $a \in \mathbf{A}$ or $a \in \mathbf{B}$. If $a \in \mathbf{A}$, then $b \in \mathsf{USets}\,\mathbf{A}$. If $a \in \mathbf{B}$, then $b \in \mathsf{USets}\,\mathbf{B}$. Either way, $b \in \mathbf{A} \cup \mathbf{B}$ and, thus, $\mathbf{A} \cup \mathbf{B} \in \mathsf{USets}\,\mathbf{P}$. Furthermore, we need to show that $\mathbf{A} \cup \mathbf{B}$ is the greatest lower bound of $\mathbf{A}$ and $\mathbf{B}$. Let $\mathbf{C} \in \mathsf{USets}\,\mathbf{P}$ such that

$$\mathbf{A} \cup \mathbf{B} \preceq_{U\mathbf{P}} \mathbf{C} \preceq_{U\mathbf{P}} \mathbf{A} \iff \mathbf{A} \cup \mathbf{B} \supseteq \mathbf{C} \supseteq \mathbf{A}$$

and

$$\mathbf{A} \cup \mathbf{B} \preceq_{U\mathbf{P}} \mathbf{C} \preceq_{U\mathbf{P}} \mathbf{B} \iff \mathbf{A} \cup \mathbf{B} \supseteq \mathbf{C} \supseteq \mathbf{B}.$$

Using the fact that union preserves inclusions, we have

$$\begin{aligned} & (\mathbf{A} \cup \mathbf{B}) \cup (\mathbf{A} \cup \mathbf{B}) \supseteq (\mathbf{C} \cup \mathbf{C}) \supseteq (\mathbf{A} \cup \mathbf{B}) \\ \iff & \mathbf{A} \cup \mathbf{B} \supseteq \mathbf{C} \supseteq (\mathbf{A} \cup \mathbf{B}) \\ \iff & \mathbf{C} = (\mathbf{A} \cup \mathbf{B}). \end{aligned}$$

Therefore, $\mathbf{A} \cup \mathbf{B}$ is the greatest lower bound of $\mathbf{A}$ and $\mathbf{B}$.

We have therefore proved that $U\mathbf{P} = \langle \mathsf{USets}\,\mathbf{P}, \preceq_{U\mathbf{P}} \rangle$ is a lattice. To show that it is bounded, we notice that $\emptyset \subseteq \mathbf{C}$ for any $\mathbf{C} \in \mathsf{USets}\,\mathbf{P}$, meaning that $\emptyset$ is the top. Furthermore, we notice that $\mathbf{C} \subseteq \mathbf{P}$ for any $\mathbf{C} \in \mathsf{USets}\,\mathbf{P}$, meaning that $\mathbf{P}$ is a bottom. Therefore, the lattice is bounded. □

Proof of Lemma 2.59. Consider the poset $L\mathbf{P} = \langle \mathsf{LSets}\,\mathbf{P}, \preceq_{L\mathbf{P}} \rangle$ and $\mathbf{A}, \mathbf{B} \in \mathsf{LSets}\,\mathbf{P}$.

First, we need to show that $\mathbf{A} \cup \mathbf{B} \in \mathsf{LSets}\,\mathbf{P}$. That is, $b \preceq_{\mathbf{P}} a$ implies $b \in \mathbf{A} \cup \mathbf{B}$. We have $\mathbf{A} \in \mathsf{LSets}\,\mathbf{P}$ and $\mathbf{B} \in \mathsf{LSets}\,\mathbf{P}$, meaning that by definition, if $a \in \mathbf{A} \cup \mathbf{B}$, either $a \in \mathbf{A}$ or $a \in \mathbf{B}$. If $a \in \mathbf{A}$, then $b \in \mathbf{A}$. If $a \in \mathbf{B}$, then $b \in \mathsf{LSets}\,\mathbf{P}$. It follows that $b \in \mathbf{A} \cup \mathbf{B}$ and, thus, all $\mathbf{A} \cup \mathbf{B}\mathsf{LSets}\,\mathbf{P}$. Furthermore, we need to show that $\mathbf{A} \cup \mathbf{B}$ is the least upper bound of $\mathbf{A}$ and $\mathbf{B}$. Consider $\mathbf{C} \in \mathsf{LSets}\,\mathbf{P}$ such that

$$\mathbf{A} \preceq_{L\mathbf{P}} \mathbf{C} \preceq_{L\mathbf{P}} \mathbf{A} \cup \mathbf{B} \iff \mathbf{A} \subseteq \mathbf{C} \subseteq \mathbf{A} \cup \mathbf{B}$$

and

$$\mathbf{B} \preceq_{L\mathbf{P}} \mathbf{C} \preceq_{L\mathbf{P}} \mathbf{A} \cup \mathbf{B} \iff \mathbf{B} \subseteq \mathbf{C} \subseteq \mathbf{A} \cup \mathbf{B}.$$

Using the fact that union preserves inclusions, we have

$$\begin{aligned} & (\mathbf{A} \cup \mathbf{B}) \subseteq (\mathbf{C} \cup \mathbf{C}) \subseteq (\mathbf{A} \cup \mathbf{B}) \\ \iff & (\mathbf{A} \cup \mathbf{B}) \subseteq \mathbf{C} \subseteq (\mathbf{A} \cup \mathbf{B}) \\ \iff & \mathbf{C} = (\mathbf{A} \cup \mathbf{B}). \end{aligned}$$

Therefore, $\mathbf{A} \cup \mathbf{B}$ is the least upper bound of $\mathbf{A}$ and $\mathbf{B}$.

Second, we need to show that $\mathbf{A} \cap \mathbf{B} \in \mathsf{LSets}\,\mathbf{P}$. That is, $b \preceq_{\mathbf{P}} a$ implies $b \in \mathbf{A} \cap \mathbf{B}$. We have $\mathbf{A} \in \mathsf{LSets}\,\mathbf{P}$ and $\mathbf{B} \in \mathsf{LSets}\,\mathbf{P}$, meaning that by definition, if $a \in \mathbf{A} \cap \mathbf{B}$, we have $a \in \mathbf{A} \wedge a \in \mathbf{B}$. Since $\mathbf{A}, \mathbf{B} \in \mathsf{LSets}\,\mathbf{P}$, this implies $b \in \mathbf{A} \wedge b \in \mathbf{B}$ and, thus, $b \in \mathbf{A} \cap \mathbf{B}$. Consider $\mathbf{C} \in \mathsf{LSets}\,\mathbf{P}$ such that

$$\mathbf{A} \cap \mathbf{B} \preceq_{L\mathbf{P}} \mathbf{C} \preceq_{L\mathbf{P}} \mathbf{A} \iff \mathbf{A} \cap \mathbf{B} \subseteq \mathbf{C} \subseteq \mathbf{A}$$

and

$$\mathbf{A} \cap \mathbf{B} \preceq_{L\mathbf{P}} \mathbf{C} \preceq_{L\mathbf{P}} \mathbf{B} \iff \mathbf{A} \cap \mathbf{B} \subseteq \mathbf{C} \subseteq \mathbf{B}.$$

Using the fact that intersection preserves inclusions, we have

$$\begin{aligned} & (\mathbf{A} \cap \mathbf{B}) \cap (\mathbf{A} \cap \mathbf{B}) \subseteq (\mathbf{C} \cap \mathbf{C}) \subseteq (\mathbf{A} \cap \mathbf{B}) \\ \iff & \mathbf{A} \cap \mathbf{B} \subseteq \mathbf{C} \subseteq (\mathbf{A} \cap \mathbf{B}) \\ \iff & \mathbf{C} = (\mathbf{A} \cap \mathbf{B}). \end{aligned}$$

Therefore, $\mathbf{A} \cap \mathbf{B}$ is the greatest lower bound of $\mathbf{A}$ and $\mathbf{B}$.

We have therefore proved that $L\mathbf{P} = \langle \mathsf{LSets}\,\mathbf{P}, \preceq_{L\mathbf{P}} \rangle$ is a lattice. To show that it is bounded, we notice that $\emptyset \subseteq \mathbf{C}$ for any $\mathbf{C} \in \mathsf{LSets}\,\mathbf{P}$, meaning that $\emptyset$ is the bottom. Furthermore, we notice that $\mathbf{C} \subseteq \mathbf{P}$ for any $\mathbf{C} \in \mathsf{LSets}\,\mathbf{P}$, meaning that $\mathbf{P}$ is a top. Therefore, the lattice is bounded. □

Proof of Lemma 4.3. We need to prove that $\mathbf{d}_g$ is a monotone map. For this, consider $\langle f^*, r \rangle \in \mathbf{F}^{\mathrm{op}} \times \mathbf{R}$ and $\langle \tilde{f}^*, \tilde{r} \rangle \in \mathbf{F}^{\mathrm{op}} \times \mathbf{R}$ with $\langle f^*, r \rangle \preceq \langle \tilde{f}^*, \tilde{r} \rangle$. In other words, we have $\tilde{f} \preceq f$ and $r \preceq \tilde{r}$.

- ▹ Case 1: $\mathbf{d}_g(f^*, r) = \bot$.

  In this case, $\mathbf{d}_g(f^*, r) \preceq \mathbf{d}_g(\tilde{f}^*, \tilde{r})$ holds, no matter which value $\mathbf{d}_g(\tilde{f}^*, \tilde{r})$ takes.
- ▹ Case 2: $\mathbf{d}_g(f^*, r) = \top$.

  In this case, it follows that $g(f) \preceq r$. Using this, we observe that

  $$g(\tilde{f}) \preceq g(f) \preceq r \preceq \tilde{r},$$

  so in particular $g(\tilde{f}) \preceq \tilde{r}$, which means that $\mathbf{d}_g(\tilde{f}^*, \tilde{r}) = \top$. In particular we then have $\mathbf{d}_g(f^*, r) \preceq \mathbf{d}_g(\tilde{f}^*, \tilde{r})$.

□

Proof of Lemma 4.5. We prove the two directions

1. Let's define $f : \mathbf{A} \to \mathbf{B}$. Given a design problem $\mathbf{d} : \mathbf{F}^{\mathrm{op}} \times \mathbf{R} \to_{\mathbf{Pos}} \mathbf{Bool}$, we define $\mathbf{K}_{\mathbf{d}} := f(\mathbf{d}) \subseteq \mathbf{F}^{\mathrm{op}} \times \mathbf{R}$ via

   $$\langle f^*, r \rangle \in \mathbf{K}_{\mathbf{d}} \Leftrightarrow \mathbf{d}(f^*, r) = \top.$$

   Now we need to prove that $\mathbf{K}_{\mathbf{d}} = f(\mathbf{d})$ is indeed an upperset. For this, consider $\langle f^*, r \rangle \in \mathbf{K}_{\mathbf{d}}$ and $\langle \tilde{f}^*, \tilde{r} \rangle \in \mathbf{F}^{\mathrm{op}} \times \mathbf{R}$ with $\langle f^*, r \rangle \preceq \langle \tilde{f}^*, \tilde{r} \rangle$. Because $\mathbf{d}$ is monotone, we have $\mathbf{d}(f^*, r) \preceq \mathbf{d}(\langle \tilde{f}^*, \tilde{r} \rangle)$. Since $\mathbf{d}(f^*, r) = \top$, it follows that $\mathbf{d}(\langle \tilde{f}^*, \tilde{r} \rangle) = \top$ must also hold, since the only element in **Bool** which is larger or equal to $\top$ is $\top$ itself. But $\mathbf{d}(\langle \tilde{f}^*, \tilde{r} \rangle) = \top$ means precisely that $\langle \tilde{f}^*, \tilde{r} \rangle \in \mathbf{K}_{\mathbf{d}}$, as was to be shown.
2. Let's define $g : \mathbf{B} \to \mathbf{A}$. Given an upperset $\mathbf{K} \subseteq \mathbf{F}^{\mathrm{op}} \times \mathbf{R}$, we define $\mathbf{d}_{\mathbf{K}} := g(\mathbf{K}) : \mathbf{F}^{\mathrm{op}} \times \mathbf{R} \to_{\mathbf{Pos}}$ **Bool** via

   $$\mathbf{d}_{\mathbf{K}}(f^*, r) = \top \Leftrightarrow \langle f^*, r \rangle \in \mathbf{K}.$$

   Now we need to prove that $\mathbf{d}_{\mathbf{K}}$ is a monotone map. For this, consider $\langle f^*, r \rangle \in \mathbf{F}^{\mathrm{op}} \times \mathbf{R}$ and $\langle \tilde{f}^*, \tilde{r} \rangle \in \mathbf{F}^{\mathrm{op}} \times \mathbf{R}$ with $\langle f^*, r \rangle \preceq \langle \tilde{f}^*, \tilde{r} \rangle$.

▷ Case 1: $\mathbf{d}_{\mathbf{K}}(f^*, r) = \bot$.

In this case, $\mathbf{d}_{\mathbf{K}}(f^*, r) \preceq \mathbf{d}_{\mathbf{K}}(\langle \tilde{f}^*, \tilde{r} \rangle)$ holds, no matter which value $\mathbf{d}_{\mathbf{K}}(\langle \tilde{f}^*, \tilde{r} \rangle)$ takes.

▷ Case 2: $\mathbf{d}_{\mathbf{K}}(f^*, r) = \top$.

In this case, we have that $\langle f^*, r \rangle \in \mathbf{K}$. Since $\mathbf{K}$ is an upper set and $\langle f^*, r \rangle \preceq \langle \tilde{f}^*, \tilde{r} \rangle$, it follows that also $\langle \tilde{f}^*, \tilde{r} \rangle \in \mathbf{K}$, which means that $\mathbf{d}_{\mathbf{K}}(\tilde{f}^*, \tilde{r}) = \top$. In particular, we thus have $\mathbf{d}_{\mathbf{K}}(f^*, r) \preceq \mathbf{d}_{\mathbf{K}}(\langle \tilde{f}^*, \tilde{r} \rangle)$.

3. We prove that f and g are inverse to one another.

▷ Given a design problem $\mathbf{d}$, we have

$$((f \fatsemi g)(\mathbf{d}))(f^*, r) = \top \Leftrightarrow \langle f^*, r \rangle \in f(\mathbf{d}) \Leftrightarrow \mathbf{d}(f^*, r) = \top.$$

Thus, $(f \fatsemi g)(\mathbf{d}) = \mathbf{d}$.

▷ Given an upperset $\mathbf{K}$, we have

$$\langle f^*, r \rangle \in f(g(\mathbf{K})) \Leftrightarrow g(\mathbf{K})(f^*, r) = \top \Leftrightarrow \langle f^*, r \rangle \in \mathbf{K}.$$

Thus, $(g \fatsemi f)(\mathbf{K}) = \mathbf{K}$.

□

Proof of Lemma 4.9. One can derive this starting from Theorem 5.4 in [136]. With $\boldsymbol{e}_t = \boldsymbol{x}_t - \hat{\boldsymbol{x}}_t$ representing the estimation error one has:

$$\begin{aligned}
\lim_{t\to\infty} \mathbb{E}\{\boldsymbol{x}_t^\intercal \mathbf{Q}_0 \boldsymbol{x}_t\} &= \lim_{t\to\infty} \mathsf{Tr}\left(\mathbf{Q}_0 \mathbb{E}\{\boldsymbol{x}_t \boldsymbol{x}_t^\intercal\}\right) \\
&= \lim_{t\to\infty} \mathsf{Tr}\left(\mathbf{Q}_0 \mathbb{E}\{(\hat{\boldsymbol{x}}_t + \boldsymbol{e}_t)(\hat{\boldsymbol{x}}_t + \boldsymbol{e}_t)^\intercal\}\right) \\
&= \lim_{t\to\infty} \mathsf{Tr}\left(\mathbf{Q}_0 \mathbb{E}\{\hat{\boldsymbol{x}}_t \hat{\boldsymbol{x}}_t^\intercal + \boldsymbol{e}_t \boldsymbol{e}_t^\intercal\}\right) \\
&= \mathsf{Tr}\left(\mathbf{Q}_0 (\mathbf{F} + \boldsymbol{\Sigma})\right),
\end{aligned}$$

where $\mathbf{F}$ can be computed using the closed-loop dynamics in (6). By applying the optimal control $\boldsymbol{u}_t^\star$, we have:

$$\begin{aligned}
\lim_{t\to\infty} \mathbb{E}\{\boldsymbol{u}_t^\intercal \mathbf{R}_0 \boldsymbol{u}_t\} &= \lim_{t\to\infty} \mathbb{E}\{\hat{\boldsymbol{x}}_t^\intercal \mathbf{K}^\intercal \mathbf{R}_0 \mathbf{K} \hat{\boldsymbol{x}}_t\} \\
&= \mathsf{Tr}\left(\mathbf{S}\mathbf{B}^* \mathbf{R}^{-1} \mathbf{R}_0 \mathbf{R}^{-1} \mathbf{B} \mathbf{S} \mathbf{F}\right).
\end{aligned}$$

□

Proof of Lemma 4.10. First, $\alpha \leq \alpha'$ implies $\mathbf{Q} = \alpha \mathbf{Q}_0 \preceq \alpha' \mathbf{Q}_0 = \mathbf{Q}'$ and $\mathbf{R} = \mathbf{R}_0/\alpha \succeq \mathbf{R}' = \mathbf{R}_0/\alpha'$. For the first part, following Lemma 4.9, we need to prove that

$$\alpha \leq \alpha' \implies \mathsf{Tr}\left(\mathbf{Q}_0 \left(\boldsymbol{\Sigma} + \mathbf{F}\right)\right)|_{\alpha} \geq \mathsf{Tr}\left(\mathbf{Q}_0 \left(\boldsymbol{\Sigma} + \mathbf{F}\right)\right)|_{\alpha'}.$$

$\mathbf{\Sigma}$ is independent from α. From (6) it is clear that $\alpha \leq \alpha' \Rightarrow \mathbf{F}_1 \succeq \mathbf{F}_2$, which confirms the statement. For the second part, following Lemma 4.9, it is easy to prove that

$$\alpha \leq \alpha' \implies \mathsf{Tr}\left(\mathbf{SB}\alpha^2\mathbf{R}_0^{-1}\mathbf{B}^*\mathbf{SF}\right)|_\alpha \leq \mathsf{Tr}\left(\mathbf{SB}\alpha'^2\mathbf{R}_0^{-1}\mathbf{B}^*\mathbf{SF}\right)|_{\alpha'}\ .$$

□

Proof of Lemma 4.12. First, we know that $\bar{\mathbf{\Sigma}}$ satisfies (5) :

$$\mathbf{A\Sigma} + \mathbf{\Sigma A}^* - \mathbf{\Sigma C}^*\mathbf{V}^{-1}\mathbf{C\Sigma} + \mathbf{W} := \mathbf{M}(\mathbf{V}, \mathbf{W}).$$

By letting $\langle \mathbf{V}_1, \mathbf{W}_1 \rangle \preceq \langle \mathbf{V}_2, \mathbf{W}_2 \rangle$, and using the notation $\mathbf{M}_i = \mathbf{M}(\mathbf{V}_i, \mathbf{W}_i)$, $i \in \{1, 2\}$, from Theorem 3 in [137] we know that $\mathbf{M}_2 - \mathbf{M}_1 \succeq \mathbf{0} \implies \bar{\mathbf{\Sigma}}(\mathbf{V}_1, \mathbf{W}_1) \preceq \bar{\mathbf{\Sigma}}(\mathbf{V}_2, \mathbf{W}_2)$. We have:

$$\mathbf{M}_2 - \mathbf{M}_1 = -\mathbf{\Sigma C}^*\left(\mathbf{V}_2^{-1} - \mathbf{V}_1^{-1}\right)\mathbf{C\Sigma} + \mathbf{W}_2 - \mathbf{W}_1 \succeq \mathbf{0},$$

where we use that $\mathbf{V}_1^{-1} - \mathbf{V}_2^{-1} \succ \mathbf{0}$ and $\mathbf{W}_2 - \mathbf{W}_1 \succ \mathbf{0}$ (Lemma 4.11). Therefore, $\bar{\mathbf{\Sigma}}$ is monotone in $\mathbf{V}$ and $\mathbf{W}$. □

Proof of Lemma 4.13. We need to prove:

$$\langle \mathbf{V}, \mathbf{W} \rangle \preceq \langle \mathbf{V}', \mathbf{W}' \rangle \implies P_{\text{effort}}(\mathbf{V}, \mathbf{W}) \leq P_{\text{effort}}(\mathbf{V}', \mathbf{W}'),$$
$$\langle \mathbf{V}, \mathbf{W} \rangle \preceq \langle \mathbf{V}', \mathbf{W}' \rangle \implies P_{\text{track}}(\mathbf{V}, \mathbf{W}) \leq P_{\text{track}}(\mathbf{V}', \mathbf{W}').$$

From Lemma 4.12, we know $\mathbf{\Sigma}(\mathbf{V}, \mathbf{W}) \preceq \mathbf{\Sigma}(\mathbf{V}', \mathbf{W}')$. Since $\mathbf{F}$ solves (6), we know $\mathbf{F}(\mathbf{V}, \mathbf{W}) \preceq \mathbf{F}(\mathbf{V}', \mathbf{W}')$. As no other term of P_{effort} or P_{track} depends on $\mathbf{V}, \mathbf{W}$, we are done. □

Proof of Lemma 4.16. We consider the continuous-time dynamics given in Def. 4.6, and sample this process with sampling period δ. The input $\boldsymbol{u}_t$ is constant over the sampling period. We can write the solution of the dynamics as

$$\boldsymbol{x}_t = \Phi(t, k\delta)\boldsymbol{x}_{k\delta} + \Gamma(t, k\delta)\boldsymbol{u}_{k\delta}, \tag{73}$$

where $\Phi(t, k\delta)$ satisfies

$$\frac{\mathrm{d}}{\mathrm{d}t}\Phi(t, k\delta) = \mathbf{A}\Phi(t, k\delta), \quad \Phi(k\delta, k\delta) = \mathbf{I}, \tag{74}$$

and

$$\Gamma(t, k\delta) = \int_{k\delta}^{t} \Phi(t, s)\boldsymbol{B}\mathrm{d}s. \tag{75}$$

The sampled version of the dynamics is

$$\boldsymbol{x}_{k\delta+\delta} = \Phi(t, k\delta)\boldsymbol{x}_{k\delta} + \Gamma(t, k\delta)\boldsymbol{u}_{k\delta} + \boldsymbol{w}_{k\delta}$$
$$\boldsymbol{y}_{k\delta} = \boldsymbol{C}\boldsymbol{x}_{k\delta} + \boldsymbol{v}_{k\delta},$$

with covariances $\mathbf{W}_\mathrm{d} = \int_0^\delta e^{\mathbf{A}\tau}\mathbf{W}e^{\mathbf{A}^\intercal\tau}\mathrm{d}\tau$, and $\mathbf{V}_\mathrm{d}$. We can now manipulate the continuous-time cost provided in Def. 4.6:

$$\begin{aligned} J &= \lim_{T\to\infty}\frac{1}{T}\mathbb{E}\{\int_0^T \left((\boldsymbol{x}_t^\intercal\mathbf{Q}\boldsymbol{x}_t)+(\boldsymbol{u}_t^\intercal\mathbf{R}\boldsymbol{u}_t)\right)\mathrm{d}t\} \\ &= \lim_{N\to\infty}\frac{1}{N}\mathbb{E}\{\sum_{k=0}^{N-1}\int_{k\delta}^{k\delta+\delta}\left((\boldsymbol{x}_t^\intercal\mathbf{Q}\boldsymbol{x}_t)+(\boldsymbol{u}_t^\intercal\mathbf{R}\boldsymbol{u}_t)\right)\mathrm{d}t\}. \end{aligned} \tag{76}$$

We can now use (73) to get:

$$\int_{k\delta}^{k\delta+\delta}\left((\boldsymbol{x}_t^\intercal\mathbf{Q}\boldsymbol{x}_t)+(\boldsymbol{u}_t^\intercal\mathbf{R}\boldsymbol{u}_t)\right)\mathrm{d}t = \boldsymbol{x}_{k\delta}^\intercal\mathbf{Q}_\mathrm{d}\boldsymbol{x}_{k\delta}+\boldsymbol{u}_{k\delta}^\intercal\mathbf{R}_\mathrm{d}\boldsymbol{u}_{k\delta},$$

$$\begin{aligned} \mathbf{Q}_\mathrm{d} &= \int_{k\delta}^{k\delta+\delta}\Phi^\intercal(s,k\delta)\mathbf{Q}\Phi(s,k\delta)\mathrm{d}s, \\ \mathbf{R}_\mathrm{d} &= \int_{k\delta}^{k\delta+\delta}(\Gamma^\intercal(s,k\delta)\mathbf{Q}\Gamma(s,k\delta)+\mathbf{R})\,\mathrm{d}s. \end{aligned}$$

Hence, (76) is $\lim_{N\to\infty}\frac{1}{N}\mathbb{E}\{\sum_{k=0}^{N-1}\boldsymbol{x}_{k\delta}^\intercal\mathbf{Q}_\mathrm{d}\boldsymbol{x}_{k\delta}+\boldsymbol{u}_{k\delta}^\intercal\mathbf{R}_\mathrm{d}\boldsymbol{u}_{k\delta}\}$, which indeed corresponds to the cost of a discrete-time LQG (Def. 4.15). Solving (74) one finds $\mathbf{A}_\mathrm{d} = \Phi(t,k\delta) = e^{\mathbf{A}\delta}$, and one can hence write (75) as $\mathbf{B}_\mathrm{d} = \Gamma(\delta,0) = \int_0^\delta e^{\mathbf{A}s}\mathbf{B}\mathrm{d}s$. □

Proof of Lemma 4.21. Consider $\langle\mathbf{V}_\mathrm{d},\mathbf{W}_\mathrm{d}\rangle \preceq \langle\mathbf{V}'_\mathrm{d},\mathbf{W}'_\mathrm{d}\rangle$. From the *Comparison Theorem* in [259], we know that the solution $\bar{\mathbf{\Gamma}}$ of (8) is monotonic in both $\mathbf{V}_\mathrm{d}$ and $\mathbf{W}_\mathrm{d}$. Furthermore, the solution of (7) does not depend on $\mathbf{V}_\mathrm{d},\mathbf{W}_\mathrm{d}$. Finally, $\mathbf{F}_\mathrm{d}$ solves the discrete Lyapunov equation (9) and analogously to Lemma 4.13, one can prove $\mathbf{F}(\mathbf{V}_\mathrm{d},\mathbf{W}_\mathrm{d}) \preceq \mathbf{F}(\mathbf{V}'_\mathrm{d},\mathbf{W}'_\mathrm{d})$. □

Proof of Lemma 5.3. We need to show that $(\mathbf{d}\fatsemi\mathbf{e})(p^*,r)$ is monotone in p^* and r. Because $\mathbf{d}$ represents a design problem, $\mathbf{d}(p^*,q)$ is monotone in p^*, and similarly $\mathbf{e}(q^*,r)$ is monotone in r. The conjunction "$\wedge$" is monotone in both variables, and likewise the "$\vee$" operation. □

Proof of Lemma 5.4. Consider $\mathbf{d}:\ \mathbf{P}\nrightarrow\mathbf{Q}$, $\mathbf{e}:\ \mathbf{Q}\nrightarrow\mathbf{R}$, $\mathbf{g}:\ \mathbf{R}\nrightarrow\mathbf{S}$. To show that the operation is

associative, we can use distributivity and commutativity in **Bool**:

$$\begin{aligned}((\mathbf{d} \fatsemi \mathbf{e}) \fatsemi \mathbf{g})(p^*, s) &= \bigvee_{r \in \mathbf{R}} \left(\bigvee_{q \in \mathbf{Q}} \mathbf{d}(p^*, q) \wedge \mathbf{e}(q^*, r) \right) \wedge \mathbf{g}(r^*, s) \\ &= \bigvee_{r \in \mathbf{R}} \left(\bigvee_{q \in \mathbf{Q}} \mathbf{d}(p^*, q) \wedge \mathbf{e}(q^*, r) \wedge \mathbf{g}(r^*, s) \right) \\ &= \bigvee_{q \in \mathbf{Q}} \mathbf{d}(p^*, q) \wedge \left(\bigvee_{r \in \mathbf{R}} \mathbf{e}(q^*, r) \wedge \mathbf{g}(r^*, s) \right) \\ &= (\mathbf{d} \fatsemi (\mathbf{e} \fatsemi \mathbf{g}))(p^*, s).\end{aligned}$$

□

Proof of Lemma 6.7. Given $\mathbf{d} : \mathbf{P} \nrightarrow \mathbf{Q}$, we need to show:

$$\mathrm{id}_{\mathbf{P}} \fatsemi \mathbf{d} = \mathbf{d} = \mathbf{d} \fatsemi \mathrm{id}_{\mathbf{Q}}.$$

In the following, we prove $\mathrm{id}_{\mathbf{P}} \fatsemi \mathbf{d} = \mathbf{d}$. Proving $\mathbf{d} \fatsemi \mathrm{id}_{\mathbf{Q}} = \mathbf{d}$ is similar. Consider the poset **Bool**. Since for $x, y \in \mathbf{Bool}$,

$$\frac{x \cong y}{x = y},$$

(also referred to as skeletality [148]), we just need to show that $\mathbf{d} \preceq \mathrm{id}_{\mathbf{P}} \fatsemi \mathbf{d}$ and $\mathrm{id}_{\mathbf{P}} \fatsemi \mathbf{d} \preceq \mathbf{d}$. Here, $\mathbf{d} \preceq \mathbf{e}$ means $\mathbf{d}(p^*, q) \preceq_{\mathbf{Bool}} \mathbf{e}(p^*, q)$ for all $p \in \mathbf{P}$, $q \in \mathbf{Q}$. We have

$$\begin{aligned}\mathbf{d}(p^*, q) &= \top \wedge \mathbf{d}(p^*, q) \\ &= \mathrm{id}_{\mathbf{P}}(p^*, p) \wedge \mathbf{d}(p^*, q) \\ &\preceq \bigvee_{p' \in \mathbf{P}} \mathrm{id}_{\mathbf{P}}(p^*, p') \wedge \mathbf{d}(p'^*, q) \\ &= (\mathrm{id}_{\mathbf{P}} \fatsemi \mathbf{d})(p^*, q).\end{aligned}$$

For the other direction, we need to show that $\mathrm{id}_{\mathbf{P}} \fatsemi \mathbf{d} \preceq \mathbf{d}$:

$$\bigvee_{p' \in \mathbf{P}} \mathrm{id}_{\mathbf{P}}(p^*, p') \wedge \mathbf{d}(p'^*, q) \preceq \mathbf{d}(p^*, q).$$

This holds if and only if $\mathrm{id}_{\mathbf{P}}(p^*, p') \wedge \mathbf{d}(p'^*, q) \preceq \mathbf{d}(p^*, q)$ for some $p' \in \mathbf{P}$. If there is no such p', then the inequality holds ($\bot \preceq \bot$ and $\bot \preceq \top$). If there is such an element p', it means that $\mathrm{id}_{\mathbf{P}}(p^*, p') = \top$ and $\mathbf{d}(p'^*, q) = \top$. We know that

$$\frac{\mathrm{id}_{\mathbf{P}}(p^*, p') = \top}{p \preceq p'}$$

and hence $\mathbf{d}(p^*, q) = \top$. □

Proof of Lemma 6.13. First, consider posets $\mathbf{P}, \mathbf{Q} \in \mathrm{Ob}_{\mathbf{DP}}$. We show that $\mathrm{id}_{\mathbf{P}} \otimes \mathrm{id}_{\mathbf{Q}} = \mathrm{id}_{\mathbf{P}\times\mathbf{Q}}$. It holds

$$\begin{aligned}(\mathrm{id}_{\mathbf{P}} \otimes \mathrm{id}_{\mathbf{Q}})(\langle p_1, q_1\rangle^*, \langle p_2, q_2\rangle) &= \mathrm{id}_{\mathbf{P}}(p_1{}^*, p_2) \wedge \mathrm{id}_{\mathbf{Q}}(q_1{}^*, q_2)\\ &= (p_1 \preceq_{\mathbf{P}} p_2) \wedge (q_1 \preceq_{\mathbf{Q}} q_2)\\ &= \langle p_1, q_1\rangle \preceq_{\mathbf{P}\times\mathbf{Q}} \langle p_2, q_2\rangle\\ &= \mathrm{id}_{\mathbf{P}\times\mathbf{Q}}(\langle p_1, q_1\rangle^*, \langle p_2, q_2\rangle)\,.\end{aligned}$$

Furthermore, consider the design problems

$$\mathbf{d}_1 : \mathbf{P}_1 \nrightarrow \mathbf{Q}_1, \quad \mathbf{d}_2 : \mathbf{P}_2 \nrightarrow \mathbf{Q}_2, \quad \mathbf{e}_1 : \mathbf{Q}_1 \nrightarrow \mathbf{R}_1, \quad \mathbf{e}_2 : \mathbf{Q}_2 \nrightarrow \mathbf{R}_2.$$

We need to show that

$$\underbrace{((\mathbf{d}_1 \fatsemi \mathbf{e}_1) \otimes (\mathbf{d}_2 \fatsemi \mathbf{e}_2))}_{\star} = ((\mathbf{d}_1 \otimes \mathbf{d}_2) \fatsemi (\mathbf{e}_1 \otimes \mathbf{e}_2))\,.$$

It holds

$$\begin{aligned}\star\,(\langle p_1, p_2\rangle^*, \langle r_1, r_2\rangle) &= (\mathbf{d}_1 \fatsemi \mathbf{e}_1)(p_1{}^*, r_1) \wedge (\mathbf{d} \fatsemi \mathbf{e}_2)(p_2{}^*, r_2)\\ &= \left(\bigvee_{q_1 \in \mathbf{Q}_1} (\mathbf{d}_1(p_1{}^*, q_1) \wedge \mathbf{e}_1(q_1{}^*, r_1))\right) \wedge \left(\bigvee_{q_2 \in \mathbf{Q}_2} (\mathbf{d}_2(p_2{}^*, q_2) \wedge \mathbf{e}_2(q_2{}^*, r_2))\right)\\ &= \bigvee_{q_1 \in \mathbf{Q}_1} \bigvee_{q_2 \in \mathbf{Q}_2} (\mathbf{d}_1(p_1{}^*, q_1) \wedge \mathbf{e}_1(q_1{}^*, r_1) \wedge \mathbf{d}_1(p_2{}^*, q_2) \wedge \mathbf{e}_2(q_2{}^*, r_2))\\ &= \bigvee_{\langle q_1, q_2\rangle \in \mathbf{Q}_1\times\mathbf{Q}_2} (\mathbf{d}_1(p_1{}^*, q_1) \wedge \mathbf{d}_2(p_2{}^*, q_2) \wedge \mathbf{e}_1(q_1{}^*, r_1) \wedge \mathbf{e}_2(q_2{}^*, r_2))\\ &= ((\mathbf{d}_1 \otimes \mathbf{d}_2) \fatsemi (\mathbf{e}_1 \otimes \mathbf{e}_2))(\langle p_1, p_2\rangle^*, \langle r_1, r_2\rangle)\,.\end{aligned}$$

Therefore, $\otimes$ is functorial in **DP**. □

Proof of Lemma 6.14. We prove this for $\mathsf{lu}_{\mathbf{P}}$, for any $\mathbf{P} \in \mathrm{Ob}_{\mathbf{DP}}$. The proof for $\mathsf{ru}_{\mathbf{P}}$ is analogous. Define the inverse of the left unitor:

$$\begin{aligned}\mathsf{lu}_{\mathbf{P}}^{-1} : \mathbf{P}^{\mathrm{op}} \times (\mathbf{1} \times \mathbf{P}) &\to_{\mathbf{Pos}} \mathbf{Bool},\\ \langle p_1{}^*, \langle \bullet, p_2\rangle\rangle &\mapsto p_1 \preceq_{\mathbf{P}} p_2.\end{aligned}$$

Clearly, this is a valid design problem, i.e., $\mathsf{lu}_{\mathbf{P}}^{-1} \in \mathrm{Hom}_{\mathbf{DP}}(\mathbf{P}; \mathbf{1} \times \mathbf{P})$. We have:

$$\begin{aligned}(\mathsf{lu}_{\mathbf{P}}^{-1} \fatsemi \mathsf{lu}_{\mathbf{P}})(p_1{}^*, p_2) &= \bigvee_{p \in \mathbf{P}} \mathsf{lu}_{\mathbf{P}}^{-1}(p_1{}^*, \langle \bullet, p \rangle) \wedge \mathsf{lu}_{\mathbf{P}}(\langle \bullet, p \rangle^*, p_2) \\ &= \bigvee_{p \in \mathbf{P}} (p_1 \preceq_{\mathbf{P}} p) \wedge (p \preceq_{\mathbf{P}} p_2) \\ &= p_1 \preceq_{\mathbf{P}} p_2 \\ &= \mathrm{id}_{\mathbf{P}}(p_1{}^*, p_2),\end{aligned}$$

and

$$\begin{aligned}(\mathsf{lu}_{\mathbf{P}} \fatsemi \mathsf{lu}_{\mathbf{P}}^{-1})(p_1{}^*, p_2) &= \bigvee_{p \in \mathbf{P}} \mathsf{lu}_{\mathbf{P}}(\langle \bullet, p_1 \rangle^*, p) \wedge \mathsf{lu}_{\mathbf{P}}^{-1}(p^*, \langle \bullet, p_2 \rangle) \\ &= \bigvee_{p \in \mathbf{P}} (p_1 \preceq_{\mathbf{P}} p) \wedge (p \preceq_{\mathbf{P}} p_2) \\ &= p_1 \preceq_{\mathbf{P}} p_2 \\ &= \mathrm{id}_{\mathbf{1} \times \mathbf{P}}(p_1{}^*, p_2) \\ &\simeq \mathrm{id}_{\mathbf{P}}(p_1{}^*, p_2).\end{aligned}$$

□

Proof of Lemma 6.15. We define the inverse of the associator. For brevity, we omit the pedix, and denote the condition $\langle\langle p, q\rangle, r\rangle \in (\mathbf{P} \times \mathbf{Q}) \times \mathbf{R}$ by † and the condition $\langle p, \langle q, r\rangle\rangle \in \mathbf{P} \times (\mathbf{Q} \times \mathbf{R})$ by ‡:

$$\begin{aligned}\mathsf{as}^{-1} : (\mathbf{P} \times (\mathbf{Q} \times \mathbf{R}))^{\mathrm{op}} \times ((\mathbf{P} \times \mathbf{Q}) \times \mathbf{R}) &\to_{\mathbf{Pos}} \mathbf{Bool}, \\ \langle p_1, \langle q_1, r_1\rangle\rangle^*, \langle\langle p_2, q_2\rangle, r_2\rangle &\mapsto p_1 \preceq p_2 \wedge q_1 \preceq q_2 \wedge r_1 \preceq r_2.\end{aligned}$$

We have:

$$\begin{aligned}&\left(\mathsf{as}^{-1} \fatsemi \mathsf{as}\right)(\langle p_1, \langle q_1, r_1\rangle\rangle^*, \langle p_2, \langle q_2, r_2\rangle\rangle) \\ &= \bigvee_{\dagger} \mathsf{as}^{-1}(\langle p_1, \langle q_1, r_1\rangle\rangle^*, \langle\langle p, q\rangle, r\rangle) \wedge \mathsf{as}(\langle\langle p, q\rangle, r\rangle^*, \langle p_2, \langle q_2, r_2\rangle\rangle) \\ &= \bigvee_{\dagger} p_1 \preceq p \wedge q_1 \preceq q \wedge r_1 \preceq r \wedge p \preceq p_2 \wedge q \preceq q_2 \wedge r \preceq r_2 \\ &= (p_1 \preceq p_2) \wedge (q_1 \preceq q_2) \wedge (r_1 \preceq r_2) \\ &= \mathrm{id}_{(\mathbf{P} \times \mathbf{Q}) \times \mathbf{R}}(\langle\langle p_1, q_1\rangle, r_1\rangle^*, \langle\langle p_2, q_2\rangle, r_2\rangle).\end{aligned}$$

and

$$
\begin{aligned}
&\left(\mathsf{as} \fatsemi \mathsf{as}^{-1}\right)(\langle\langle p_1, q_1\rangle, r_1\rangle^*, \langle\langle p_2, q_2\rangle, r_2\rangle) \\
&= \bigvee_{\ddagger} \mathsf{as}(\langle\langle p_1, q_1\rangle, r_1\rangle^*, \langle p, \langle q, r\rangle\rangle) \wedge \mathsf{as}^{-1}(\langle p, \langle q, r\rangle\rangle^*, \langle\langle p_2, q_2\rangle, r_2\rangle) \\
&= \bigvee_{\ddagger} p_1 \preceq p \wedge q_1 \preceq q \wedge r_1 \preceq r \wedge p \preceq p_2 \wedge q \preceq q_2 \wedge r \preceq r_2 \\
&= (p_1 \preceq p_2) \wedge (q_1 \preceq q_2) \wedge (r_1 \preceq r_2) \\
&= \mathsf{id}_{\mathbf{P}\times(\mathbf{Q}\times\mathbf{R})}(\langle p_1, \langle q_1, r_1\rangle\rangle^*, \langle p_2, \langle q_2, r_2\rangle\rangle) \\
&\simeq \mathsf{id}_{(\mathbf{P}\times\mathbf{Q})\times\mathbf{R}}(\langle\langle p_1, q_1\rangle, r_1\rangle^*, \langle\langle p_2, q_2\rangle, r_2\rangle).
\end{aligned}
$$

□

Proof of Lemma 6.16. We have already checked functoriality of the monoidal product, and that the other constituents are valid. We therefore just need to check the triangle and the pentagon identities (Fig. 175).

Triangle identity We want to show that

$$\mathsf{as}_{\mathbf{P},\mathbf{1},\mathbf{Q}} \fatsemi (\mathsf{id}_{\mathbf{P}} \otimes \mathsf{lu}_{\mathbf{Q}}) = \mathsf{ru}_{\mathbf{P}} \otimes \mathsf{id}_{\mathbf{Q}}.$$

We start by looking at the left-hand side of the equation. One has (by dropping the pedix of the associator for brevity, and by denoting the condition $\langle p, \langle \bullet, q\rangle\rangle \in \mathbf{P} \times (\mathbf{1} \times \mathbf{Q})$ by †):

$$
\begin{aligned}
&(\mathsf{as} \fatsemi (\mathsf{id}_{\mathbf{P}} \otimes \mathsf{lu}_{\mathbf{Q}}))(\langle\langle p_1, \bullet\rangle, q_1\rangle^*, \langle p_2, q_2\rangle) \\
&= \bigvee_{\dagger} \mathsf{as}(\langle\langle p_1, \bullet\rangle, q_1\rangle^*, \langle p, \langle \bullet, q\rangle\rangle) \wedge (\mathsf{id}_{\mathbf{P}} \otimes \mathsf{lu}_{\mathbf{Q}})(\langle p, \langle \bullet, q\rangle\rangle^*, \langle p_2, q_2\rangle) \\
&= \bigvee_{\dagger} p_1 \preceq p \wedge q_1 \preceq q \wedge p \preceq p_2 \wedge q \preceq q_2 \\
&= p_1 \preceq p_2 \wedge q_1 \preceq q_2 \\
&= (\mathsf{ru}_{\mathbf{P}} \otimes \mathsf{id}_{\mathbf{Q}})(\langle\langle p_1, \bullet\rangle, q_1\rangle^*, \langle p_2, q_2\rangle),
\end{aligned}
$$

proving the identity.

Pentagon identity We want to show that

$$\mathsf{as}_{\mathbf{P}\times\mathbf{Q},\mathbf{R},\mathbf{S}} \fatsemi \mathsf{as}_{\mathbf{P},\mathbf{Q},\mathbf{R}\times\mathbf{S}} = (\mathsf{as}_{\mathbf{P},\mathbf{Q},\mathbf{R}} \otimes \mathsf{id}_{\mathbf{S}}) \fatsemi \mathsf{as}_{\mathbf{P},\mathbf{Q}\times\mathbf{R},\mathbf{S}} \fatsemi (\mathsf{id}_{\mathbf{P}} \otimes \mathsf{as}_{\mathbf{Q},\mathbf{R},\mathbf{S}}).$$

These are a bit of a mouthful, so let's spell out their signatures for clarity:

$$\mathsf{as}_{\mathbf{P}\times\mathbf{Q},\mathbf{R},\mathbf{S}} : (((\mathbf{P}\times\mathbf{Q})\times\mathbf{R})\times\mathbf{S}) \nrightarrow ((\mathbf{P}\times\mathbf{Q})\times(\mathbf{R}\times\mathbf{S})),$$
$$\mathsf{as}_{\mathbf{P},\mathbf{Q},\mathbf{R}\times\mathbf{S}} : ((\mathbf{P}\times\mathbf{Q})\times(\mathbf{R}\times\mathbf{S})) \nrightarrow (\mathbf{P}\times(\mathbf{Q}\times(\mathbf{R}\times\mathbf{S}))),$$
$$\mathsf{as}_{\mathbf{P},\mathbf{Q},\mathbf{R}} \otimes \mathsf{id}_{\mathbf{S}} : (((\mathbf{P}\times\mathbf{Q})\times\mathbf{R})\times\mathbf{S}) \nrightarrow (\mathbf{P}\times(\mathbf{Q}\times(\mathbf{R}\times\mathbf{S}))),$$
$$\mathsf{as}_{\mathbf{P},\mathbf{Q}\times\mathbf{R},\mathbf{S}} : ((\mathbf{P}\times(\mathbf{Q}\times\mathbf{R}))\times\mathbf{S}) \nrightarrow (\mathbf{P}\times((\mathbf{Q}\times\mathbf{R})\times\mathbf{S})),$$
$$\mathsf{id}_{\mathbf{P}} \otimes \mathsf{as}_{\mathbf{Q},\mathbf{R},\mathbf{S}} : (\mathbf{P}\times((\mathbf{Q}\times\mathbf{R})\times\mathbf{S})) \nrightarrow (\mathbf{P}\times(\mathbf{Q}\times(\mathbf{R}\times\mathbf{S}))).$$

One has, by denoting the condition $\langle\langle p, q\rangle, \langle r, s\rangle\rangle \in (\mathbf{P}\times\mathbf{Q})\times(\mathbf{R}\times\mathbf{S})$ by †:

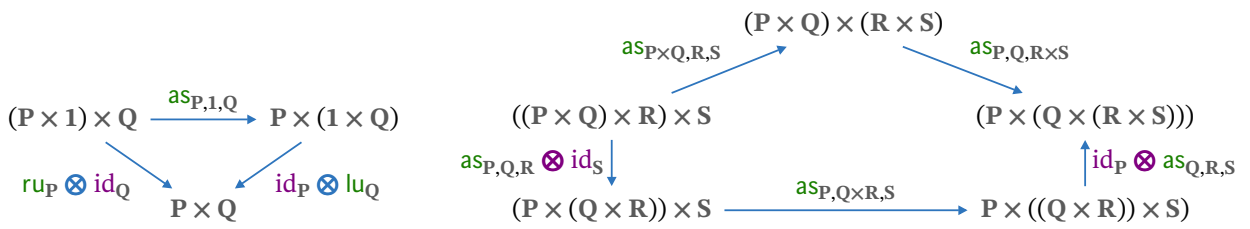
$$\begin{aligned}
&(\mathsf{as}_{\mathbf{P}\times\mathbf{Q},\mathbf{R},\mathbf{S}} \fatsemi \mathsf{as}_{\mathbf{P},\mathbf{Q},\mathbf{R}\times\mathbf{S}})(\langle\langle\langle p_1, q_1\rangle, r_1\rangle, s_1\rangle^*, \langle p_2, \langle q_2, \langle r_2, s_2\rangle\rangle\rangle) \\
&= \bigvee_{\dagger} \mathsf{as}_{\mathbf{P}\times\mathbf{Q},\mathbf{R},\mathbf{S}}(\langle\langle\langle p_1, q_1\rangle, r_1\rangle, s_1\rangle^*, \langle\langle p, q\rangle, \langle r, s\rangle\rangle) \wedge \mathsf{as}_{\mathbf{P},\mathbf{Q},\mathbf{R}\times\mathbf{S}}(\langle\langle p, q\rangle, \langle r, s\rangle\rangle^*, \langle p_2, \langle q_2, \langle r_2, s_2\rangle\rangle\rangle) \\
&= \bigvee_{\dagger} p_1 \preceq p \wedge q_1 \preceq q \wedge r_1 \preceq r \wedge s_1 \preceq s \wedge p \preceq p_2 \wedge q \preceq q_2 \wedge r \preceq r_2 \wedge s \preceq s_2 \\
&= p_1 \preceq p_2 \wedge q_1 \preceq q_2 \wedge r_1 \preceq r_2 \wedge s_1 \preceq s_2,
\end{aligned}$$

given the associativity of $\wedge$.

Furthermore, one has, by denoting the condition $\langle\langle p, \langle q, r\rangle\rangle, s\rangle \in (\mathbf{P}\times(\mathbf{Q}\times\mathbf{R}))\times\mathbf{S}$ as ‡:

$$\begin{aligned}
&((\mathsf{as}_{\mathbf{P},\mathbf{Q},\mathbf{R}} \otimes \mathsf{id}_{\mathbf{S}}) \fatsemi (\mathsf{as}_{\mathbf{P},\mathbf{Q}\times\mathbf{R},\mathbf{S}}))(\langle\langle\langle p_1, q_1\rangle, r_1\rangle, s_1\rangle^*, \langle p_2, \langle\langle q_2, r_2\rangle, s_2\rangle\rangle) \\
&= \bigvee_{\ddagger} (\mathsf{as}_{\mathbf{P},\mathbf{Q},\mathbf{R}} \otimes \mathsf{id}_{\mathbf{S}})(\langle\langle\langle p_1, q_1\rangle, r_1\rangle, s_1\rangle^*, \langle\langle p, \langle q, r\rangle\rangle, s\rangle) \wedge \\
&\quad (\mathsf{as}_{\mathbf{P},\mathbf{Q}\times\mathbf{R},\mathbf{S}})(\langle\langle p, \langle q, r\rangle\rangle, s\rangle^*, \langle p_2, \langle\langle q_2, r_2\rangle, s_2\rangle\rangle) \\
&\quad = \bigvee_{\ddagger} p_1 \preceq p \wedge q_1 \preceq q \wedge r_1 \preceq r \wedge s_1 \preceq s \wedge p \preceq p_2 \wedge q \preceq q_2 \wedge r \preceq r_2 \wedge s \preceq s_2 \\
&\quad = p_1 \preceq p_2 \wedge q_1 \preceq q_2 \wedge r_1 \preceq r_2 \wedge s_1 \preceq s_2,
\end{aligned}$$

and the second composition of the right-hand side works similarly. Therefore, $\langle \mathbf{DP}, \otimes, \mathbf{1}, \mathsf{lu}, \mathsf{ru}, \mathsf{as}\rangle$ is a monoidal category. □



(a) Triangle identity for **DP**.

(b) Pentagon identity for **DP**.

Figure 175: Identities for the monoidal structure on **DP**.

Proof of Lemma 6.21. The inverse of $\mathsf{br}_{\mathbf{P},\mathbf{Q}}$ is $\mathsf{br}_{\mathbf{Q},\mathbf{P}}$. To show the isomorphism, one has:

$$\begin{aligned}
&(\mathsf{br}_{\mathbf{P},\mathbf{Q}} \fatsemi \mathsf{br}_{\mathbf{Q},\mathbf{P}})(\langle p_1, q_1\rangle^*, \langle p_2, q_2\rangle)\\
&= \bigvee_{\langle q,\, p\rangle \in \mathbf{Q}\times\mathbf{P}} \mathsf{br}_{\mathbf{P},\mathbf{Q}}(\langle p_1, q_1{}^*, \langle q, p\rangle\rangle) \wedge \mathsf{br}_{\mathbf{Q},\mathbf{P}}(\langle q, p\rangle^*, \langle p_2, q_2\rangle)\\
&= \bigvee_{\langle q,\, p\rangle \in \mathbf{Q}\times\mathbf{P}} p_1 \preceq p \wedge q_1 \preceq q \wedge p \preceq p_2 \wedge q \preceq q_2\\
&= p_1 \preceq q_1 \wedge p_2 \preceq q_2\\
&= \mathsf{id}_{\mathbf{P}\times\mathbf{Q}}\\
&\simeq \mathsf{id}_{\mathbf{Q}\times\mathbf{P}}\\
&= (\mathsf{br}_{\mathbf{Q},\mathbf{P}} \fatsemi \mathsf{br}_{\mathbf{P},\mathbf{Q}})(\langle q_1, p_1\rangle^*, \langle q_2, p_2\rangle).
\end{aligned}$$

For naturality, consider $\mathbf{d} : \mathbf{P}_1 \nrightarrow \mathbf{Q}_1$, $\mathbf{e} : \mathbf{P}_2 \nrightarrow \mathbf{Q}_2$. We first evaluate $((\mathbf{d} \otimes \mathbf{e}) \fatsemi \mathsf{br}_{\mathbf{Q}_1,\mathbf{Q}_2}) : \mathbf{P}_1 \times \mathbf{P}_2 \nrightarrow \mathbf{Q}_2 \times \mathbf{P}_2$. One has:

$$\begin{aligned}
&((\mathbf{d} \otimes \mathbf{e}) \fatsemi \mathsf{br}_{\mathbf{Q}_1,\mathbf{Q}_2})(\langle p_1, p_2{}^*, \langle q_2, q_1\rangle\rangle)\\
&= \bigvee_{\langle q,\, q'\rangle \in \mathbf{Q}_1\times\mathbf{Q}_2} (\mathbf{d} \otimes \mathbf{e})(\langle p_1, p_2\rangle^*, \langle q, q'\rangle) \wedge \mathsf{br}_{\mathbf{Q}_1,\mathbf{Q}_2}(\langle q, q'\rangle^*, \langle q_2, q_1\rangle)\\
&= \bigvee_{\langle q,\, q'\rangle \in \mathbf{Q}_1\times\mathbf{Q}_2} \mathbf{d}(p_1{}^*, q) \wedge \mathbf{e}(p_2{}^*, q') \wedge q \preceq q_1 \wedge q' \preceq q_2
\end{aligned}$$

We now evaluate $(\mathsf{br}_{\mathbf{P}_1,\mathbf{P}_2} \fatsemi (\mathbf{e} \otimes \mathbf{d})) : \mathbf{P}_1 \times \mathbf{P}_2 \nrightarrow \mathbf{Q}_2 \times \mathbf{P}_2$. One has:

$$\begin{aligned}
&(\mathsf{br}_{\mathbf{P}_1,\mathbf{P}_2} \fatsemi (\mathbf{e} \otimes \mathbf{d}))(\langle p_1, p_2\rangle^*, \langle q_2, q_1\rangle)\\
&= \bigvee_{\langle p',\, p\rangle \in \mathbf{P}_2\times\mathbf{P}_1} \mathsf{br}_{\mathbf{P}_1,\mathbf{P}_2}(\langle p_1, p_2\rangle^*, \langle p', p\rangle) \wedge (\mathbf{e} \otimes \mathbf{d})(\langle p', p\rangle^*, \langle q_2, q_1\rangle)\\
&= \bigvee_{\langle p',\, p\rangle \in \mathbf{P}_2\times\mathbf{P}_1} p_1 \preceq p \wedge p_2 \preceq p' \wedge \mathbf{e}(p', q_2) \wedge \mathbf{d}(p, q_1).
\end{aligned}$$

Given the monotonicity properties of $\mathbf{d}, \mathbf{e}$, the above equations are equivalent, proving naturality. □

Proof of Lemma 6.22. Lemma 6.21 also shows the symmetry condition. We start by showing the Hexagon identities. For the first, we need to prove:

$$\underbrace{\underbrace{\mathsf{as}_{\mathbf{P},\mathbf{Q},\mathbf{R}} \fatsemi \mathsf{br}_{\mathbf{P},\mathbf{Q}\times\mathbf{R}}}_{①} \fatsemi \mathsf{as}_{\mathbf{Q},\mathbf{R},\mathbf{P}}}_{②} = \underbrace{\underbrace{(\mathsf{br}_{\mathbf{P},\mathbf{Q}} \otimes \mathsf{id}_{\mathbf{R}}) \fatsemi \mathsf{as}_{\mathbf{Q},\mathbf{P},\mathbf{R}}}_{③} \fatsemi (\mathsf{id}_{\mathbf{Q}} \otimes \mathsf{br}_{\mathbf{P},\mathbf{R}})}_{④}.$$

First of all, both maps share the signature $(\mathbf{P} \times \mathbf{Q}) \times \mathbf{R} \nrightarrow \mathbf{Q} \times (\mathbf{R} \times \mathbf{P})$. We have, by denoting the

condition $\langle p, \langle q, r\rangle\rangle \in \mathbf{P} \times (\mathbf{Q} \times \mathbf{R})$ by †:

$$
\begin{aligned}
&①(\langle\langle p_1, q_1\rangle, r_1\rangle^*, \langle\langle q_2, r_2\rangle, p_2\rangle) \\
&= \bigvee_{\dagger} \mathsf{as}_{\mathbf{P},\mathbf{Q},\mathbf{R}}(\langle\langle\langle p_1, q_1\rangle, r_1\rangle^*, \langle p, \langle q, r\rangle\rangle) \wedge \mathsf{br}_{\mathbf{P},\mathbf{Q}\times\mathbf{R}}(\langle p, \langle q, r\rangle\rangle^*, \langle\langle q_2, r_2\rangle, p_2\rangle) \\
&= \bigvee_{\dagger} p_1 \preceq p \wedge q_1 \preceq q \wedge r_1 \preceq r \wedge p \preceq p_2 \wedge q \preceq q_2 \wedge r \preceq r_2 \\
&= p_1 \preceq p_2 \wedge q_1 \preceq q_2 \wedge r_1 \preceq r_2.
\end{aligned}
$$

Furthermore, by denoting the condition $\langle\langle q, r\rangle, p\rangle \in (\mathbf{Q} \times \mathbf{R}) \times \mathbf{P}$ by ‡, one has:

$$
\begin{aligned}
&②(\langle\langle p_1, q_1\rangle, r_1\rangle^*, \langle q_2, \langle r_2, p_2\rangle\rangle) \\
&= \bigvee_{\ddagger} ①(\langle\langle p_1, q_1\rangle, r_1\rangle^*, \langle\langle q, r\rangle, p\rangle) \wedge \mathsf{as}_{\mathbf{Q},\mathbf{R},\mathbf{P}}(\langle\langle q, r\rangle, p\rangle^*, \langle q_2, \langle r_2, p_2\rangle\rangle) \\
&= \bigvee_{\ddagger} p_1 \preceq p \wedge q_1 \preceq q \wedge r_1 \preceq r \wedge p \preceq p_2 \wedge q \preceq q_2 \wedge r \preceq r_2 \\
&= p_1 \preceq p_2 \wedge q_1 \preceq q_2 \wedge r_1 \preceq r_2.
\end{aligned}
$$

Similarly, by denoting the condition $\langle\langle q, p\rangle, r\rangle \in (\mathbf{Q} \times \mathbf{P}) \times \mathbf{R}$ by ≀ one has:

$$
\begin{aligned}
&③(\langle\langle p_1, q_1\rangle, r_1\rangle^*, \langle r_2, \langle p_2, r_2\rangle\rangle) \\
&= \bigvee_{\wr} (\mathsf{br}_{\mathbf{P},\mathbf{Q}} \otimes \mathsf{id}_{\mathbf{R}})(\langle\langle p_1, q_1\rangle, r_1\rangle^*, \langle\langle q, p\rangle, r\rangle) \wedge \mathsf{as}_{\mathbf{Q},\mathbf{P},\mathbf{R}}(\langle\langle q, p\rangle, r\rangle^*, \langle r_2, \langle p_2, r_2\rangle\rangle) \\
&= \bigvee_{\wr} p_1 \preceq p \wedge q_1 \preceq q \wedge r_1 \preceq r \wedge p \preceq p_2 \wedge q \preceq q_2 \wedge r \preceq r_2 \\
&= p_1 \preceq p_2 \wedge q_1 \preceq q_2 \wedge r_1 \preceq r_2.
\end{aligned}
$$

Furthermore, by denoting the condition $\langle q, \langle p, r\rangle\rangle \in \mathbf{Q} \times (\mathbf{P} \times \mathbf{R})$ by ⋄ one has:

$$
\begin{aligned}
&④(\langle\langle p_1, q_1\rangle, r_1\rangle^*, \langle q_2, \langle r_2, p_2\rangle\rangle) \\
&= \bigvee_{\diamond} ③(\langle\langle p_1, q_1\rangle, r_1\rangle^*, \langle q, \langle p, r\rangle\rangle) \wedge (\mathsf{id}_{\mathbf{Q}} \otimes \mathsf{br}_{\mathbf{P},\mathbf{R}})(\langle q, \langle p, r\rangle\rangle^*, \langle q_2, \langle r_2, p_2\rangle\rangle) \\
&= \bigvee_{\diamond} p_1 \preceq p \wedge q_1 \preceq q \wedge r_1 \preceq r \wedge p \preceq p_2 \wedge q \preceq q_2 \wedge r \preceq r_2 \\
&= p_1 \preceq p_2 \wedge q_1 \preceq q_2 \wedge r_1 \preceq r_2,
\end{aligned}
$$

proving the original statement. The other Hexagon identity follows analogously. To show the first

triangle identity, we write

$$\begin{aligned}
&(\mathsf{br}_{\mathbf{1}\times\mathbf{P}} \fatsemi \mathsf{ru}_{\mathbf{P}})(\langle \bullet, p_1\rangle^*, p_2)\\
&= \bigvee_{\langle p, \bullet\rangle \in \mathbf{P}\times\mathbf{1}} \mathsf{br}_{\mathbf{1}\times\mathbf{P}}(\langle \bullet, p_1\rangle^*, \langle p, \bullet\rangle) \wedge \mathsf{ru}_{\mathbf{P}}(\langle p, \bullet\rangle^*, p_2)\\
&= \bigvee_{\langle p, \bullet\rangle \in \mathbf{P}\times\mathbf{1}} (\bullet \preceq \bullet) \wedge (p_1 \preceq p) \wedge (p \preceq p_2)\\
&= p_1 \preceq p_2\\
&= \mathsf{lu}_{\mathbf{P}}(\langle \bullet, p_1\rangle^*, p_2)\,.
\end{aligned}$$

□

Proof of Lemma 6.24. We prove the trace axioms one-by-one.

Naturality in P Consider $\mathbf{d} : \mathbf{P}\times\mathbf{R} \nrightarrow \mathbf{Q}\times\mathbf{R}$ and $\mathbf{e} : \mathbf{T} \nrightarrow \mathbf{P}$. We need to show:

$$\underbrace{\mathrm{Tr}^{\mathbf{R}}_{\mathbf{T},\mathbf{Q}}(\underbrace{(\mathbf{e}\otimes \mathrm{id}_{\mathbf{R}}) \fatsemi \mathbf{d}}_{①})}_{②} = \mathbf{e} \fatsemi \mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{Q}}(\mathbf{d}).$$

We start by ①. We have:

$$\begin{aligned}
&①(\langle t, r_1\rangle^*, \langle q, r_2\rangle)\\
&= \bigvee_{\langle p, r\rangle \in \mathbf{P}\times\mathbf{R}} (\mathbf{e}\otimes \mathrm{id}_{\mathbf{R}})(\langle t, r_1\rangle^*, \langle p, r\rangle) \wedge \mathbf{d}(\langle p, r\rangle^*, \langle q, r_2\rangle)\\
&= \bigvee_{\langle p, r\rangle \in \mathbf{P}\times\mathbf{R}} \mathbf{e}(t^*, p) \wedge r_1 \preceq r \wedge \mathbf{d}(\langle p, r\rangle^*, \langle q, r_2\rangle).
\end{aligned}$$

Consequently:

$$\begin{aligned}
②(\langle t^*, q\rangle) &= \bigvee_{r\in\mathbf{R}} ①(\langle t, r\rangle^*, \langle q, r\rangle)\\
&= \bigvee_{r\in\mathbf{R}} \bigvee_{\langle p, r'\rangle \in \mathbf{P}\times\mathbf{R}} \mathbf{e}(t^*, p) \wedge r \preceq r' \wedge \mathbf{d}(\langle p, r'\rangle^*, \langle q, r\rangle).
\end{aligned}$$

We now look at the right-hand side of the original equation. One has:

$$\begin{aligned}
&(\mathbf{e} \fatsemi \mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{Q}}(\mathbf{d}))(\langle t^*, q\rangle)\\
&= \bigvee_{p\in\mathbf{P}} \mathbf{e}(t^*, p) \wedge \bigvee_{r\in\mathbf{R}} \mathbf{d}(\langle p, r\rangle^*, \langle q, r\rangle),
\end{aligned}$$

which is equivalent to the previous one due to the monotonicity of $\mathbf{d}$.

Naturality in $\mathbf{Q}$ Consider $\mathbf{d} : \mathbf{P} \times \mathbf{R} \nrightarrow \mathbf{Q} \times \mathbf{R}$ and $\mathbf{e} : \mathbf{Q} \nrightarrow \mathbf{T}$. We need to show:

$$\underbrace{\mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{T}}(\underbrace{\mathbf{d} \fatsemi (\mathbf{e} \otimes \mathrm{id}_{\mathbf{R}})}_{①})}_{②} = \mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{Q}}(\mathbf{d}) \fatsemi \mathbf{e}$$

We start by ①. We have:

$$\begin{aligned} &①(\langle p, r_1\rangle^*, \langle t, r_2\rangle) \\ &= \bigvee_{\langle q, r\rangle \in \mathbf{Q}\times\mathbf{R}} \mathbf{d}(\langle p, r_1\rangle^*, \langle q, r\rangle) \wedge (\mathbf{e} \otimes \mathrm{id}_{\mathbf{R}})(\langle q, r\rangle^*, \langle t, r_2\rangle) \\ &= \bigvee_{\langle q, r\rangle \in \mathbf{Q}\times\mathbf{R}} \mathbf{d}(\langle p, r_1\rangle^*, \langle q, r\rangle) \wedge \mathbf{e}(q^*, t) \wedge r \preceq r_2. \end{aligned}$$

We can therefore evaluate ②. We have:

$$\begin{aligned} ②(p^*, t) &= \bigvee_{r \in \mathbf{R}} ①(\langle p, r\rangle^*, \langle t, r\rangle) \\ &= \bigvee_{r \in \mathbf{R}} \bigvee_{\langle q, r'\rangle \in \mathbf{Q}\times\mathbf{R}} \mathbf{d}(\langle p, r\rangle^*, \langle q, r'\rangle) \wedge \mathbf{e}(q^*, t) \wedge r' \preceq r. \end{aligned}$$

We can look at the right-hand side of the original statement to prove. We have:

$$(\mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{Q}}(\mathbf{d}) \fatsemi \mathbf{e})(p^*, t) = \bigvee_{q \in \mathbf{Q}} \bigvee_{r \in \mathbf{R}} \mathbf{d}(\langle p, r\rangle^*, \langle q, r\rangle) \wedge \mathbf{e}(q^*, t),$$

which, again, is equivalent to the previous equation due to monotonicity of $\mathbf{d}$.

Dinaturality in $\mathbf{R}$ Given $\mathbf{d} : \mathbf{P} \times \mathbf{R} \nrightarrow \mathbf{Q} \times \mathbf{S}$, $\mathbf{e} : \mathbf{S} \nrightarrow \mathbf{R}$, we want to prove

$$\mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{Q}}(\mathbf{d} \fatsemi (\mathrm{id}_{\mathbf{Q}} \otimes \mathbf{e})) = \mathrm{Tr}^{\mathbf{S}}_{\mathbf{P},\mathbf{Q}}((\mathrm{id}_{\mathbf{P}} \otimes \mathbf{e}) \fatsemi \mathbf{d}). \tag{77}$$

We start by elaborating the inner part of the left-hand side of (77). One has:

$$\begin{aligned} &(\mathbf{d} \fatsemi (\mathrm{id}_{\mathbf{Q}} \otimes \mathbf{e}))(\langle p, r_1\rangle^*, \langle q, r_2\rangle) \\ &= \bigvee_{\langle q', s\rangle \in \mathbf{Q}\times\mathbf{S}} \mathbf{d}(\langle p, r_1\rangle^* \langle q', s\rangle) \wedge (\mathrm{id}_{\mathbf{Q}} \otimes \mathbf{e})(\langle q', s\rangle^*, \langle q, r_2\rangle) \\ &= \bigvee_{\langle q', s\rangle \in \mathbf{Q}\times\mathbf{S}} \mathbf{d}(\langle p, r_1\rangle^* \langle q', s\rangle) \wedge q' \preceq q \wedge \mathbf{e}(s^*, r_2). \end{aligned}$$

The complete evaluation of the left-hand side of (77) reads:

$$\begin{aligned}
&\mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{Q}}(\mathbf{d} \fatsemi (\mathrm{id}_{\mathbf{Q}} \otimes \mathbf{e}))(p^*, q) \\
&= \bigvee_{r \in \mathbf{R}} (\mathbf{d} \fatsemi (\mathrm{id}_{\mathbf{Q}} \otimes \mathbf{e}))(\langle p, r\rangle^*, \langle q, r\rangle) \\
&= \bigvee_{r \in \mathbf{R}} \bigvee_{\langle q', s\rangle \in \mathbf{Q}\times\mathbf{S}} \mathbf{d}(\langle p, r\rangle^* \langle q', s\rangle) \wedge q' \preceq q \wedge \mathbf{e}(s^*, r).
\end{aligned}$$

We now evaluate the inner part of the right-hand side of (77):

$$\begin{aligned}
&((\mathrm{id}_{\mathbf{P}} \otimes \mathbf{e}) \fatsemi \mathbf{d})(\langle p, s_1\rangle^*, \langle q, s_2\rangle) \\
&= \bigvee_{\langle p', r\rangle \in \mathbf{P}\times\mathbf{R}} (\mathrm{id}_{\mathbf{P}} \otimes \mathbf{e})(\langle p, s_1\rangle^*, \langle p', r\rangle) \wedge \mathbf{d}(\langle p', r\rangle^*, \langle q, s_2\rangle) \\
&= \bigvee_{\langle p', r\rangle \in \mathbf{P}\times\mathbf{R}} p \preceq p' \wedge \mathbf{e}({s_1}^*, r) \wedge \mathbf{d}(\langle p', r\rangle^*, \langle q, s_2\rangle).
\end{aligned}$$

The complete evaluation of the right-hand side of (77) gives:

$$\begin{aligned}
&\mathrm{Tr}^{\mathbf{S}}_{\mathbf{P},\mathbf{Q}}((\mathrm{id}_{\mathbf{P}} \otimes \mathbf{e}) \fatsemi \mathbf{d})(p^*, q) \\
&= \bigvee_{s \in \mathbf{S}} ((\mathrm{id}_{\mathbf{P}} \otimes \mathbf{e}) \fatsemi \mathbf{d})(\langle p, s\rangle^*, \langle q, s\rangle) \\
&= \bigvee_{s \in \mathbf{S}} \bigvee_{\langle p', r\rangle \in \mathbf{P}\times\mathbf{R}} p \preceq p' \wedge \mathbf{e}(s^*, r) \wedge \mathbf{d}(\langle p', r\rangle^*, \langle q, s\rangle)
\end{aligned}$$

The equivalence of the expressions comes from the monotonicity of $\mathbf{d}$.

Vanishing I For any design problem $\mathbf{d} : \mathbf{P} \times \mathbf{1} \nrightarrow \mathbf{Q} \times \mathbf{1}$, we want to prove:

$$\mathrm{Tr}^{\mathbf{1}}_{\mathbf{P},\mathbf{Q}}(\mathbf{d}) = \mathsf{ru}_{\mathbf{P}}^{-1} \fatsemi \mathbf{d} \fatsemi \mathsf{ru}_{\mathbf{Q}}.$$

By developing the right-hand side, one has:

$$\begin{aligned}
&(\mathsf{ru}_{\mathbf{P}}^{-1} \fatsemi \mathbf{d} \fatsemi \mathsf{ru}_{\mathbf{Q}})({p_1}^*, q_1) \\
&= \bigvee_{\langle p, \bullet\rangle \in \mathbf{P}\times\mathbf{1}} \mathsf{ru}_{\mathbf{P}}^{-1}({p_1}^*, \langle p, \bullet\rangle) \wedge (\mathbf{d} \fatsemi \mathsf{ru}_{\mathbf{Q}})(\langle p, \bullet\rangle^*, q_1) \\
&= \bigvee_{\langle p, \bullet\rangle \in \mathbf{P}\times\mathbf{1}} p_1 \preceq p \wedge \bigvee_{\langle q, \bullet\rangle \in \mathbf{Q}\times\mathbf{1}} \mathbf{d}(\langle p, \bullet\rangle^*, \langle q, \bullet\rangle) \wedge \mathsf{ru}_{\mathbf{Q}}(\langle q, \bullet\rangle^*, q_1) \\
&= \bigvee_{\langle p, \bullet\rangle \in \mathbf{P}\times\mathbf{1}, \langle q, \bullet\rangle \in \mathbf{Q}\times\mathbf{1}} p_1 \preceq p \wedge q \preceq q_1 \wedge \mathbf{d}(\langle p, \bullet\rangle^*, \langle q, \bullet\rangle) \\
&= \mathbf{d}(\langle p_1, \bullet\rangle^*, q_1) \\
&= \mathrm{Tr}^{\mathbf{1}}_{\mathbf{P},\mathbf{Q}}(\mathbf{d})({p_1}^*, q_1),
\end{aligned}$$

by monotonicity of $\mathbf{d}$.

Vanishing II Given $\mathbf{d} : (\mathbf{P} \times \mathbf{R}) \times \mathbf{S} \nrightarrow (\mathbf{Q} \times \mathbf{R}) \times \mathbf{S}$, we want to prove:

$$\mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{Q}}(\mathrm{Tr}^{\mathbf{S}}_{\mathbf{P}\times\mathbf{R},\mathbf{Q}\times\mathbf{R}}(\mathbf{d})) = \mathrm{Tr}^{\mathbf{R}\times\mathbf{S}}_{\mathbf{P},\mathbf{Q}}(\mathsf{as}^{-1}_{\mathbf{P},\mathbf{R},\mathbf{S}} \fatsemi \mathbf{d} \fatsemi \mathsf{as}_{\mathbf{Q},\mathbf{R},\mathbf{S}}). \tag{78}$$

We start by evaluating the left-hand side of (78). We have:

$$\mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{Q}}(\mathrm{Tr}^{\mathbf{S}}_{\mathbf{P}\times\mathbf{R},\mathbf{Q}\times\mathbf{R}}(\mathbf{d}))(p^*, q) = \bigvee_{r\in\mathbf{R}} \bigvee_{s\in\mathbf{S}} \mathbf{d}(\langle\langle p, r\rangle, s\rangle^*, \langle\langle q, r\rangle, s\rangle)$$

Furthermore, the inner part of the right-hand side of (78) reads:

$$\begin{aligned}
&(\mathsf{as}^{-1}_{\mathbf{P},\mathbf{R},\mathbf{S}} \fatsemi \mathbf{d} \fatsemi \mathsf{as}_{\mathbf{Q},\mathbf{R},\mathbf{S}})(\langle p_1, \langle r_1, s_1\rangle\rangle^*, \langle q_2, \langle r_2, s_2\rangle\rangle) \\
&= \bigvee_{\langle\langle p, r\rangle, s\rangle\in(\mathbf{P}\times\mathbf{R})\times\mathbf{S}} \mathsf{as}^{-1}_{\mathbf{P},\mathbf{R},\mathbf{S}}(\langle p_1, \langle r_1, s_1\rangle\rangle^*, \langle\langle p, r\rangle, s\rangle) \wedge (\mathbf{d} \fatsemi \mathsf{as}_{\mathbf{Q},\mathbf{R},\mathbf{S}})(\langle\langle p, r\rangle, s\rangle^*, \langle q_2, \langle r_2, s_2\rangle\rangle) \\
&= \bigvee_{\langle\langle p, r\rangle, s\rangle\in(\mathbf{P}\times\mathbf{R})\times\mathbf{S}} p_1 \preceq p \wedge r_1 \preceq r \wedge s_1 \preceq s \wedge \bigvee_{\langle\langle q, r'\rangle, s'\rangle\in(\mathbf{Q}\times\mathbf{R})\times\mathbf{S}} \mathbf{d}(\langle\langle p, r\rangle, s\rangle^*, \langle\langle q, r'\rangle, s'\rangle) \\
&\wedge q \preceq q_2 \wedge r' \preceq r_2 \wedge s' \preceq s_2.
\end{aligned}$$

Therefore, the right-hand side of (78) reads:

$$\begin{aligned}
&\mathrm{Tr}^{\mathbf{R}\times\mathbf{S}}_{\mathbf{P},\mathbf{Q}}(\mathsf{as}^{-1}_{\mathbf{P},\mathbf{R},\mathbf{S}} \fatsemi \mathbf{d} \fatsemi \mathsf{as}_{\mathbf{Q},\mathbf{R},\mathbf{S}})(p_1{}^*, q_1) \\
&= \bigvee_{\langle r, s\rangle\in\mathbf{R}\times\mathbf{S}} (\mathsf{as}^{-1}_{\mathbf{P},\mathbf{R},\mathbf{S}} \fatsemi \mathbf{d} \fatsemi \mathsf{as}_{\mathbf{Q},\mathbf{R},\mathbf{S}})(\langle p_1, \langle r, s\rangle\rangle^*, \langle q_1, \langle r, s\rangle\rangle)
\end{aligned}$$

By pluggin-in previous results, and leveraging the monotonicity of $\mathbf{d}$, one obtains the desired result.

Superposing Given $\mathbf{d} : \mathbf{P} \times \mathbf{R} \nrightarrow \mathbf{Q} \times \mathbf{R}$ we want to prove:

$$\mathrm{Tr}^{\mathbf{R}}_{\mathbf{S}\times\mathbf{P},\mathbf{S}\times\mathbf{Q}}(\mathsf{as}^{-1}_{\mathbf{S},\mathbf{P},\mathbf{R}} \fatsemi \mathsf{id}_{\mathbf{S}} \otimes \mathbf{d} \fatsemi \mathsf{as}_{\mathbf{S},\mathbf{Q},\mathbf{R}}) = \mathsf{id}_{\mathbf{S}} \otimes \mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{R}}(\mathbf{d}).$$

The proof is very similar in style to the one of Vanishing II.

Yanking We want to prove:

$$\mathrm{Tr}^{\mathbf{R}}_{\mathbf{R},\mathbf{R}}(\mathsf{br}_{\mathbf{R},\mathbf{R}}) = \mathsf{id}_{\mathbf{R}}.$$

By developing the left-hand side further, once has:

$$\begin{aligned}\mathrm{Tr}^{\mathrm{R}}_{\mathrm{R},\mathrm{R}}(\mathsf{br}_{\mathrm{R},\mathrm{R}})(r_1{}^*, r_2) &= \bigvee_{r\in\mathrm{R}} \mathsf{br}_{\mathrm{R},\mathrm{R}}(\langle r_1, r\rangle^*, \langle r_2, r\rangle)\\ &= \bigvee_{r\in\mathrm{R}} r_1 \preceq r \wedge r \preceq r_2\\ &= r_1 \preceq r_2\\ &= \mathsf{id}_{\mathrm{R}}(r_1{}^*, r_2).\end{aligned}$$

□

Proof of Lemma 6.28. First, we need to prove that $\mathsf{Hom}_{\mathbf{DP}}(\mathbf{P};\mathbf{Q})$ is a poset. To prove this, we check the following:

▹ *Reflexivity*: Given $\mathbf{d} \in \mathsf{Hom}_{\mathbf{DP}}(\mathbf{P};\mathbf{Q})$:

$$\frac{\top}{\mathbf{d} \preceq_{\mathbf{DP}} \mathbf{d}};$$

▹ *Antisymmetry*: Given $\mathbf{d}, \mathbf{e} \in \mathsf{Hom}_{\mathbf{DP}}(\mathbf{P};\mathbf{Q})$:

$$\frac{\mathbf{d} \preceq_{\mathbf{DP}} \mathbf{e} \quad \mathbf{e} \preceq_{\mathbf{DP}} \mathbf{d}}{\mathbf{d} = \mathbf{e}};$$

▹ *Transitivity*: Given $\mathbf{d}, \mathbf{e}, \mathbf{g} \in \mathsf{Hom}_{\mathbf{DP}}(\mathbf{P};\mathbf{Q})$:

$$\frac{\mathbf{d} \preceq_{\mathbf{DP}} \mathbf{e} \quad \mathbf{e} \preceq_{\mathbf{DP}} \mathbf{g}}{\mathbf{d} \preceq_{\mathbf{DP}} \mathbf{g}}.$$

Therefore, $\mathsf{Hom}_{\mathbf{DP}}(\mathbf{P};\mathbf{Q})$ is a poset. Furthermore, consider two design problems $\mathbf{d}, \mathbf{e} \in \mathsf{Hom}_{\mathbf{DP}}(\mathbf{P};\mathbf{Q})$. Their greatest lower bound (meet) is $\mathbf{d} \wedge \mathbf{e}$, since it is the greatest design problem implying both $\mathbf{d}$ and $\mathbf{e}$. Their least upper bound (join), instead, is $\mathbf{d} \vee \mathbf{e}$, since it is the least design problem implied by both $\mathbf{d}$ and $\mathbf{e}$. This proves that $\mathsf{Hom}_{\mathbf{DP}}$ is a lattice. To prove that it is bounded, we identify the top element as $\top_{\mathbf{P},\mathbf{Q}}$ (it is implied by all other design problems) and the bottom element as $\bot_{\mathbf{P},\mathbf{Q}}$ (it implies by all the other design problems). □

Proof of Lemma 6.31. Consider any $\mathbf{P}, \mathbf{Q} \in \mathsf{Ob}_{\mathbf{DP}}$ and $\mathsf{Hom}_{\mathbf{DP}}(\mathbf{P};\mathbf{Q})$. We have already shown that $\mathbf{A} = \mathsf{Hom}_{\mathbf{DP}}(\mathbf{P};\mathbf{Q})$ is a bounded lattice (Lemma 6.28). Now, take any subset $\mathbf{B}$ of $\mathbf{A}$. We define the following two design problems:

$$\begin{aligned}\bigvee_{\mathbf{d}\in\mathbf{B}} \mathbf{d} : \mathbf{P}^{\mathrm{op}} \times \mathbf{Q} &\to_{\mathbf{Pos}} \mathbf{Bool},\\ \langle p, q\rangle &\mapsto \exists \mathbf{d} \in \mathbf{B} : \mathbf{d}(p^*, q),\end{aligned}$$

and

$$\bigwedge_{\mathbf{d}\in\mathbf{B}} \mathbf{d} : \mathbf{P}^{\mathrm{op}} \times \mathbf{Q} \to_{\mathbf{Pos}} \mathbf{Bool},$$

$$\langle p^*, q\rangle \mapsto \forall \mathbf{d} \in \mathbf{B} : \mathbf{d}(p^*, q).$$

These are clearly design problems (given that $\mathbf{d}$ is a design problem) and given their signature they belong to $\mathbf{A}$. We will now argue that $\bigvee_{\mathbf{d}\in\mathbf{B}} \mathbf{d}$ is the supremum of $\mathbf{B}$ and $\bigwedge_{\mathbf{d}\in\mathbf{B}} \mathbf{d}$ is the infimum of $\mathbf{B}$.

$\bigvee_{\mathbf{d}\in\mathbf{B}} \mathbf{d}$ is the supremum of $\mathbf{B}$: First, for any $\mathbf{d} \in \mathbf{B}$, we know that $\mathbf{d} \preceq_{\mathbf{DP}} \mathbf{d} \vee \bigvee_{\mathbf{d}\in\mathbf{B}\setminus\mathbf{d}} \mathbf{d} = \bigvee_{\mathbf{d}\in\mathbf{B}} \mathbf{d}$, proving that $\bigvee_{\mathbf{d}\in\mathbf{B}} \mathbf{d}$ is an upper bound of $\mathbf{B}$. We now want to show that $\bigvee_{\mathbf{d}\in\mathbf{B}} \mathbf{d}$ is the least upper bound of $\mathbf{B}$: for any upper bound $\mathbf{e}$ of $\mathbf{B}$, we need to show $\bigvee_{\mathbf{d}\in\mathbf{B}} \mathbf{d} \preceq_{\mathbf{DP}} \mathbf{e}$. In other words, for any pair $\langle p^*, q\rangle \in \mathbf{P}^{\mathrm{op}} \times \mathbf{Q}$, we need to show $(\bigvee_{\mathbf{d}\in\mathbf{B}} \mathbf{d})(p^*, q) \preceq_{\mathbf{Bool}} \mathbf{e}(p^*, q)$. Fix any $\langle p^*, q\rangle$. If $(\bigvee_{\mathbf{d}\in\mathbf{B}} \mathbf{d})(p^*, q) = \bot$, the condition is trivially satisfied.

If, instead, $(\bigvee_{\mathbf{d}\in\mathbf{B}} \mathbf{d})(p^*, q) = \top$, there exists a $\mathbf{d} \in \mathbf{B}$ such that $\mathbf{d}(p^*, q) = \top$. Given that $\mathbf{e}$ is an upper bound of $\mathbf{B}$, this implies $\top = \mathbf{d}(p^*, q) \preceq_{\mathbf{Bool}} \mathbf{e}(p^*, q) = \top$, proving the condition.

$\bigwedge_{\mathbf{d}\in\mathbf{B}} \mathbf{d}$ is the infimum of $\mathbf{B}$: First, for any $\mathbf{d} \in \mathbf{B}$, we know that $\mathbf{d} \wedge \bigwedge_{\mathbf{d}\in\mathbf{B}\setminus\mathbf{d}} \mathbf{d} = \bigwedge_{\mathbf{d}\in\mathbf{B}} \mathbf{d} \preceq_{\mathbf{DP}} \mathbf{d}$, proving that $\bigwedge_{\mathbf{d}\in\mathbf{B}} \mathbf{d}$ is a lower bound of $\mathbf{B}$. We now want to show that $\bigwedge_{\mathbf{d}\in\mathbf{B}} \mathbf{d}$ is the greatest lower bound of $\mathbf{B}$: for any lower bound $\mathbf{e}$ of $\mathbf{B}$, we need to show $\mathbf{e} \preceq_{\mathbf{DP}} \bigwedge_{\mathbf{d}\in\mathbf{B}} \mathbf{d}$. In other words, for any pair $\langle p^*, q\rangle \in \mathbf{P}^{\mathrm{op}} \times \mathbf{Q}$, we need to show $\mathbf{e}(p^*, q) \preceq_{\mathbf{Bool}} (\bigwedge_{\mathbf{d}\in\mathbf{B}} \mathbf{d})(p^*, q)$. Fix any $\langle p^*, q\rangle$. If $(\bigwedge_{\mathbf{d}\in\mathbf{B}} \mathbf{d})(p^*, q) = \top$, the condition is trivially satisfied. If, instead, $(\bigwedge_{\mathbf{d}\in\mathbf{B}} \mathbf{d})(p^*, q) = \bot$, there is at least one $\mathbf{d} \in \mathbf{B}$ for which $\mathbf{d}(p^*, q) = \bot$. Given that $\mathbf{e}$ is a lower bound of $\mathbf{B}$, this implies $\bot = \mathbf{e}(p^*, q) \preceq_{\mathbf{Bool}} \mathbf{d}(p^*, q) = \bot$, proving the condition. □

Proof of Lemma 6.34. We have:

$$\begin{aligned}
&((\mathbf{d} \wedge \mathbf{e}) \vee \mathbf{g})(p^*, q)\\
&= (\mathbf{d} \wedge \mathbf{e})(p^*, q) \vee \mathbf{g}(p^*, q)\\
&= (\mathbf{d}(p^*, q) \wedge \mathbf{e}(p^*, q)) \vee \mathbf{g}(p^*, q)\\
&= (\mathbf{d}(p^*, q) \vee \mathbf{g}(p^*, q)) \wedge (\mathbf{e}(p^*, q) \vee \mathbf{g}(p^*, q))\\
&= ((\mathbf{d} \vee \mathbf{g}) \wedge (\mathbf{e} \vee \mathbf{g}))(p^*, q).
\end{aligned}$$

□

Proof of Lemma 6.35. We have:

$$\begin{aligned}
&((\mathbf{d} \vee \mathbf{e}) \wedge \mathbf{g})(p^*, q) \\
&= (\mathbf{d} \vee \mathbf{e})(p^*, q) \wedge \mathbf{g}(p^*, q) \\
&= (\mathbf{d}(p^*, q) \vee \mathbf{e}(p^*, q)) \vee \mathbf{g}(p^*, q) \\
&= (\mathbf{d}(p^*, q) \wedge \mathbf{g}(p^*, q)) \vee (\mathbf{e}(p^*, q) \wedge \mathbf{g}(p^*, q)) \\
&= ((\mathbf{d} \wedge \mathbf{g}) \vee (\mathbf{e} \wedge \mathbf{g}))(p^*, q).
\end{aligned}$$

□

Proof of Lemma 6.37. We have:

$$\begin{aligned}
&((\mathbf{d} \vee \mathbf{e}) \fatsemi \mathbf{g})(p^*, r) \\
&= \bigvee_{q \in \mathbf{Q}} (\mathbf{d} \vee \mathbf{e})(p^*, q) \wedge \mathbf{g}(q^*, r) \\
&= \bigvee_{q \in \mathbf{Q}} (\mathbf{d}(p^*, q) \vee \mathbf{e}(p^*, q)) \wedge \mathbf{g}(q^*, r) \\
&= \bigvee_{q \in \mathbf{Q}} (\mathbf{d}(p^*, q) \wedge \mathbf{g}(q^*, r)) \vee (\mathbf{e}(p^*, q) \wedge \mathbf{g}(q^*, r)) \\
&= ((\mathbf{d} \fatsemi \mathbf{g}) \vee (\mathbf{e} \fatsemi \mathbf{g}))(p^*, r).
\end{aligned}$$

□

Proof of Lemma 6.39. We have:

$$\begin{aligned}
&((\mathbf{d} \wedge \mathbf{e}) \fatsemi \mathbf{g})(p^*, r) \\
&= \bigvee_{q \in \mathbf{Q}} (\mathbf{d} \wedge \mathbf{e})(p^*, q) \wedge \mathbf{g}(q^*, r) \\
&= \bigvee_{q \in \mathbf{Q}} (\mathbf{d}(p^*, q) \wedge \mathbf{e}(p^*, q)) \wedge \mathbf{g}(q^*, r) \\
&= \bigvee_{q \in \mathbf{Q}} (\mathbf{d}(p^*, q) \wedge \mathbf{g}(q^*, r)) \wedge (\mathbf{e}(p^*, q) \wedge \mathbf{g}(q^*, r)) \\
&= ((\mathbf{d} \fatsemi \mathbf{g}) \wedge (\mathbf{e} \fatsemi \mathbf{g}))(p^*, r).
\end{aligned}$$

□

Proof of Lemma 6.41. We want to prove the following:

$$\frac{\langle \mathbf{d}, \mathbf{e} \rangle \preceq \left\langle \mathbf{d}', \mathbf{e}' \right\rangle}{(\mathbf{d} \fatsemi \mathbf{e}) \preceq_{\mathbf{DP}} (\mathbf{d}' \fatsemi \mathbf{e}')}$$

This is easy to show. Let's start from $\mathbf{d} \preceq_{\mathbf{DP}} \mathbf{d}'$, $\mathbf{e} \preceq_{\mathbf{DP}} \mathbf{e}'$. We have:

$$\begin{aligned}(\mathbf{d} \fatsemi \mathbf{e})(p^*, r) &= \bigvee_{q \in \mathbf{Q}} \mathbf{d}(p^*, q) \wedge \mathbf{e}(q^*, r) \\ &\preceq_{\mathbf{Bool}} \bigvee_{q \in \mathbf{Q}} \mathbf{d}'(p^*, q) \wedge \mathbf{e}'(q^*, r) \\ &= (\mathbf{d}' \fatsemi \mathbf{e}')(p^*, r).\end{aligned}$$

□

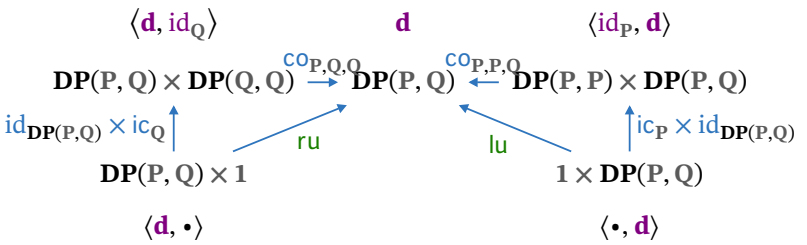
Proof of Lemma 6.42. We first define the identity-choosing morphism as:

$$\begin{aligned}\mathrm{ic}_{\mathbf{P}} : \mathbf{1} &\to_{\mathbf{Pos}} \mathrm{Hom}_{\mathbf{DP}}(\mathbf{P}; \mathbf{P}), \\ \bullet &\mapsto \mathrm{id}_{\mathbf{P}}.\end{aligned}$$

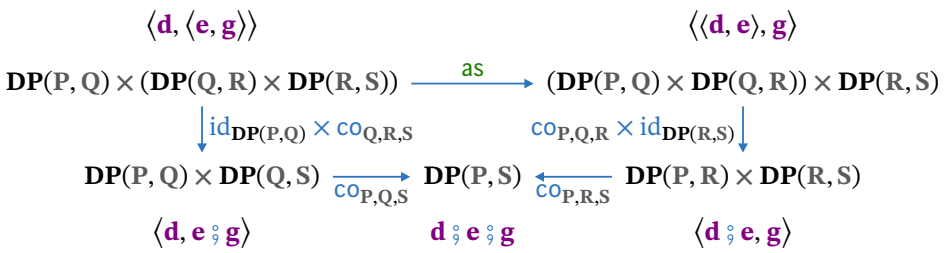
This is trivially monotone and hence a valid morphism in **Pos**. Furthermore, we define the composition morphism as:

$$\begin{aligned}\mathrm{co}_{\mathbf{P},\mathbf{Q},\mathbf{R}} : \mathrm{Hom}_{\mathbf{DP}}(\mathbf{P}; \mathbf{Q}) \times \mathrm{Hom}_{\mathbf{DP}}(\mathbf{Q}; \mathbf{R}) &\to_{\mathbf{Pos}} \mathrm{Hom}_{\mathbf{DP}}(\mathbf{P}; \mathbf{R}), \\ \langle \mathbf{d}, \mathbf{e} \rangle &\mapsto \mathbf{d} \fatsemi \mathbf{e}.\end{aligned},$$

which a valid morphism in **Pos**. We now want to check neutrality and associativity commuting diagrams, and in the following, we denote $\mathrm{Hom}_{\mathbf{DP}}(\mathbf{P}; \mathbf{Q})$ by $\mathbf{DP}(\mathbf{P}, \mathbf{Q})$ We start by neutrality, and we do this graphically:



Similarly, for associativity one has:



□

Proof of Lemma 7.4. Consider a design problem $\mathbf{d} : \mathbf{F} \nrightarrow \mathbf{R}$ and $f \preceq f'$. We know

$$\begin{aligned} H_{\mathbf{d}}(f) &= \{r \in \mathbf{R} : \mathbf{d}(f^*, r)\} \\ &\supseteq \{r \in \mathbf{R} : \mathbf{d}(f'^*, r)\} \\ &= H_{\mathbf{d}}(f'), \end{aligned}$$

showing monotonicity of H. Analogously, consider a DP $\mathbf{d} : \mathbf{F} \nrightarrow \mathbf{R}$ and $r \preceq r'$. We know

$$\begin{aligned} K_{\mathbf{d}}(r) &= \{f \in \mathbf{F} : \mathbf{d}(f^*, r)\} \\ &\subseteq \{f \in \mathbf{F} : \mathbf{d}(f^*, r')\} \\ &= K_{\mathbf{d}}(r'), \end{aligned}$$

showing monotonicity of K. □

Proof of Lemma 7.7. We prove that $\mathbf{Pos}_U$ is a category. The proof for $\mathbf{Pos}_L$ is analogous. In the following, we show unitality and associativity.

Unitality: Given $f : \mathbf{P} \to \mathbf{Q}$, we have:

$$\begin{aligned} (f \fatsemi \mathrm{id}_{\mathbf{Q}})^\star(p) &= \bigcup_{q \in f^\star(p)} \mathrm{id}_{\mathbf{Q}}{}^\star(q) \\ &= \bigcup_{q \in f^\star(p)} {\uparrow}\{q\} \\ &= \bigcup_{q \in f^\star(p)} \{q' \in \mathbf{Q} : q \preceq_{\mathbf{Q}} q'\}. \end{aligned}$$

We know that $f^\star(p)$ is an upper set:

$$\begin{aligned} f^\star(p) &= \bigcup_{q \in f^\star(p)} \{q\} \\ &= \bigcup_{q \in f^\star(p)} \{q' \in \mathbf{Q} : q \preceq_{\mathbf{Q}} q'\}. \end{aligned}$$

Therefore, $(f \fatsemi \mathrm{id}_{\mathbf{Q}})^\star(p) = f^\star(p)$ for all $p \in \mathbf{P}$. Similarly, we have:

$$\begin{aligned} (\mathrm{id}_{\mathbf{P}} \fatsemi f)^\star(p) &= \bigcup_{p' \in \mathrm{id}_{\mathbf{P}}{}^\star(p)} f^\star(p') \\ &= \bigcup_{p' \in {\uparrow}\{p\}} f^\star(p') \\ &= f^\star(p), \end{aligned}$$

where the last equality holds since $f^\star$ is a monotone function and $f^\star(p') \subseteq f^\star(p)$ for all $p' \in {\uparrow}\{p\}$.

Associativity: Consider three morphisms $f : \mathbf{P} \to \mathbf{Q}$, $g : \mathbf{Q} \to \mathbf{R}$, and $h : \mathbf{R} \to \mathbf{S}$. We have:

$$\begin{aligned}((f \fatsemi g) \fatsemi h)^\star(p) &= \bigcup_{r \in \left(\bigcup_{q \in f^\star(p)} g^\star(q)\right)} h^\star(r) \\ &= \bigcup_{q \in f^\star(p)} \bigcup_{r \in g^\star(q)} h^\star(r) \\ &= (f \fatsemi (g \fatsemi h))^\star(p).\end{aligned}$$

Therefore, $\mathbf{Pos}_U$ is a category. □

Proof of Lemma 7.8. To prove this, we need to define the needed functors and to show that they satisfy the listed properties. We choose the functors to be the ones that map a poset $\mathbf{P}$ in a category to its opposite version $\mathbf{P}^{\mathrm{op}}$ in another category. Given a morphism $f : \mathbf{P} \to \mathbf{Q}$ in $\mathbf{Pos}_U$, we have:

$$\begin{aligned}(\swarrow (f))^\star : \mathbf{P}^{\mathrm{op}} &\to_{\mathbf{Pos}} L\mathbf{Q}^{\mathrm{op}} \\ p &\mapsto f^\star(p).\end{aligned}$$

Given a morphism $g : \mathbf{P} \to \mathbf{Q}$ in $\mathbf{Pos}_L$, we have:

$$\begin{aligned}(\nearrow (g))^\star : \mathbf{P}^{\mathrm{op}} &\to_{\mathbf{Pos}} U\mathbf{Q}^{\mathrm{op}} \\ p &\mapsto g^\star(p).\end{aligned}$$

$\swarrow$ and $\nearrow$ are functors:

▷ *Preservation of identities*: Given $\mathbf{P} \in \mathrm{Ob}_{\mathbf{Pos}_U}$, we have:

$$\begin{aligned}(\swarrow (\mathrm{id}_{\mathbf{P}}))^\star &= \uparrow_{\mathbf{P}} \{p\} \\ &= \downarrow_{\mathbf{P}^{\mathrm{op}}} \{p\} \\ &= \mathrm{id}_{\mathbf{P}^{\mathrm{op}}}{}^\star,\end{aligned}$$

where $\mathrm{id}_{\mathbf{P}}$ is an identity morphism in $\mathbf{Pos}_U$, and $\mathrm{id}_{\mathbf{P}^{\mathrm{op}}}$ is an identity morphism in $\mathbf{Pos}_L$. Similarly, given $\mathbf{P} \in \mathrm{Ob}_{\mathbf{Pos}_L}$ we have:

$$\begin{aligned}(\nearrow (\mathrm{id}_{\mathbf{P}}))^\star &= \downarrow_{\mathbf{P}} \{p\} \\ &= \uparrow_{\mathbf{P}^{\mathrm{op}}} \{p\} \\ &= \mathrm{id}^\star_{\mathbf{P}^{\mathrm{op}}}.\end{aligned}$$

▷ *Preservation of composition*: This can be easily seen as follows. Given any $f \in \mathrm{Hom}_{\mathbf{Pos}_U}(\mathbf{P}; \mathbf{Q})$, $g \in \mathrm{Hom}_{\mathbf{Pos}_U}(\mathbf{Q}; \mathbf{R})$:

$$\begin{aligned}(\swarrow (f \fatsemi g))^\star &= (f \fatsemi g)^\star \\ &= (\swarrow (f) \fatsemi \swarrow (g))^\star.\end{aligned}$$

Similarly, given any $f \in \mathrm{Hom}_{\mathbf{Pos}_L}(\mathbf{P};\mathbf{Q})$, $g \in \mathrm{Hom}_{\mathbf{Pos}_L}(\mathbf{Q};\mathbf{R})$:

$$\begin{aligned}(\nearrow(f \fatsemi g))^\star &= (f \fatsemi g)^\star \\ &= (\nearrow(f) \fatsemi \nearrow(g))^\star.\end{aligned}$$

Compositions return identity functors: We want to show that by composing the two functors we obtain the identity functors in $\mathbf{Pos}_U$ and $\mathbf{Pos}_L$, respectively. Clearly, composing the two functors returns the identity on the objects, since for any poset $\mathbf{P}$, we have $(\mathbf{P}^{\mathrm{op}})^{\mathrm{op}} = \mathbf{P}$. The functors act on morphisms by "flipping the context", and "flipping"twice is the "same"as not flipping. □

Proof of Lemma 7.9. Consider two posets $\mathbf{P}, \mathbf{Q}$ and two respective upper sets $\mathbf{A}, \mathbf{B}$. We have

$$\frac{a \in \mathbf{A} \quad a \preceq_{\mathbf{P}} a'}{a' \in \mathbf{A}},$$

and

$$\frac{b \in \mathbf{B} \quad b \preceq_{\mathbf{Q}} b'}{b' \in \mathbf{B}}.$$

Therefore:

$$\frac{\langle a, b\rangle \in \mathbf{A} \times \mathbf{B} \quad \langle a, b\rangle \preceq_{\mathbf{P}\times\mathbf{Q}} \langle a', b'\rangle}{\langle a', b'\rangle \in \mathbf{A} \times \mathbf{B}},$$

which proves that $\mathbf{A} \times \mathbf{B}$ is an upper set. The proof for the product of lower sets is analogous. □

Proof of Lemma 7.10. First, we prove that $\mathrm{id}_{\mathbf{P}} \otimes \mathrm{id}_{\mathbf{Q}} = \mathrm{id}_{\mathbf{P}\times\mathbf{Q}}$. We have:

$$\begin{aligned}(\mathrm{id}_{\mathbf{P}} \otimes \mathrm{id}_{\mathbf{Q}})^\star(p, q) &= {\mathrm{id}_p}^\star \times {\mathrm{id}_q}^\star \\ &= \{\uparrow p\} \times \{\uparrow q\} \\ &= {\mathrm{id}_{\mathbf{P}\times\mathbf{Q}}}^\star.\end{aligned}$$

Now consider $f_1 : \mathbf{P}_1 \to \mathbf{Q}_1$, $f_2 : \mathbf{P}_2 \to \mathbf{Q}_2$, $g_1 : \mathbf{Q}_1 \to \mathbf{R}_1$, $g_2 : \mathbf{Q}_2 \to \mathbf{R}_2$. We want to prove:

$$(f_1 \fatsemi g_1) \otimes (f_2 \fatsemi g_2) = ((f_1 \otimes f_2) \fatsemi (g_1 \otimes g_2)).$$

By expanding the left-hand side, we have:

$$\begin{aligned}((f_1 \fatsemi g_1) \otimes (f_2 \fatsemi g_2))^\star(p_1, p_2) &= (f_1 \fatsemi g_1)^\star(p_1) \times (f_2 \fatsemi g_2)^\star(p_2) \\ &= \bigcup_{q \in {f_1}^\star(p_1)} g_1(q) \times \bigcup_{q \in {f_2}^\star(p_2)} g_2(q)\end{aligned}$$

By expanding the right-hand side, we have:

$$\begin{aligned}((f_1 \otimes f_2) \fatsemi (g_1 \otimes g_2))^\star(p_1, p_2) &= \bigcup_{\langle q, q' \rangle \in (f_1 \otimes f_2)^\star(p_1, p_2)} (g_1 \otimes g_2)(q, q') \\ &= \bigcup_{q \in {f_1}^\star(p_1)} \bigcup_{q' \in {f_2}^\star(p_2)} {g_1}^\star(q) \times {g_2}^\star(q'),\end{aligned}$$

which confirms functoriality. □

Proof of Lemma 7.11. First, we need define their inverses:

$$\begin{aligned}{\mathsf{lu}_{\mathbf{P}}^{-1}}^\star : \mathbf{P} &\to_{\mathbf{Pos}} U(\mathbf{1} \times \mathbf{P}), \\ p &\mapsto \mathbf{1} \times \uparrow\{p\},\end{aligned}$$

and

$$\begin{aligned}{\mathsf{ru}_{\mathbf{P}}^{-1}}^\star : \mathbf{P} &\to_{\mathbf{Pos}} U(\mathbf{P} \times \mathbf{1}), \\ p &\mapsto \uparrow\{p\} \times \mathbf{1}.\end{aligned}$$

We prove the isomorphism for the left unitor. The proof for the right unitor is analogous. We have:

$$\begin{aligned}(\mathsf{lu}_{\mathbf{P}} \fatsemi \mathsf{lu}_{\mathbf{P}}^{-1})^\star(\langle \bullet, p \rangle) &= \bigcup_{s \in \mathsf{lu}_{\mathbf{P}}^\star(\langle \bullet, p \rangle)} {\mathsf{lu}_{\mathbf{P}}^{-1}}^\star(s) \\ &= \mathbf{1} \times \uparrow\{p\} \\ &= \mathrm{id}_{\mathbf{1} \times \mathbf{P}}.\end{aligned}$$

Furthermore, we have:

$$\begin{aligned}(\mathsf{lu}_{\mathbf{P}}^{-1} \fatsemi \mathsf{lu}_{\mathbf{P}})^\star(p) &= \bigcup_{s \in {\mathsf{lu}_{\mathbf{P}}^{-1}}^\star(p)} {\mathsf{lu}_{\mathbf{P}}}^\star(s) \\ &= \uparrow\{p\} \\ &= \mathrm{id}_{\mathbf{P}} \\ &\simeq \mathrm{id}_{\mathbf{1} \times \mathbf{P}}.\end{aligned}$$

□

Proof of Lemma 7.12. We first define its inverse:

$$\begin{aligned}{\mathsf{as}_{\mathbf{P},\mathbf{Q},\mathbf{R}}^{-1}}^\star : \mathbf{P} \times (\mathbf{Q} \times \mathbf{R}) &\to_{\mathbf{Pos}} (U\mathbf{P} \times U\mathbf{Q}) \times U\mathbf{R}, \\ \langle p, \langle q, r \rangle \rangle &\mapsto (\uparrow\{p\} \times \uparrow\{q\}) \times \uparrow\{r\}.\end{aligned}$$

We have:

$$\begin{aligned}
&(\mathsf{as}_{\mathbf{P},\mathbf{Q},\mathbf{R}} \fatsemi \mathsf{as}^{-1}_{\mathbf{P},\mathbf{Q},\mathbf{R}})^\star(\langle\langle p, q\rangle, r\rangle)\\
&= \bigcup_{\langle p', \langle q', r'\rangle\rangle \in \mathsf{as}_{\mathbf{P},\mathbf{Q},\mathbf{R}}{}^\star(\langle\langle p, q\rangle, r\rangle)} \mathsf{as}^{-1}_{\mathbf{P},\mathbf{Q},\mathbf{R}}{}^\star(\langle p', \langle q', r'\rangle\rangle)\\
&= \uparrow\{p\} \times \uparrow\{q\} \times \uparrow\{r\}\\
&= \mathsf{id}_{(\mathbf{P}\times\mathbf{Q})\times\mathbf{R}}(\langle\langle p, q\rangle, r\rangle)\\
&\simeq \mathsf{id}_{\mathbf{P}\times(\mathbf{Q}\times\mathbf{R})}(\langle p, \langle q, r\rangle\rangle)\\
&= (\mathsf{as}^{-1}_{\mathbf{P},\mathbf{Q},\mathbf{R}} \fatsemi \mathsf{as}_{\mathbf{P},\mathbf{Q},\mathbf{R}})^\star(\langle p, \langle q, r\rangle\rangle)
\end{aligned}$$

□

Proof of Lemma 7.13. We have already proved that the monoidal product is functorial, and that the provided constituents are of valid form. We now show the triangle and pentagon identities.

Triangle identity We want to show:

$$\mathsf{as}_{\mathbf{P},\mathbf{1},\mathbf{Q}} \fatsemi (\mathsf{id}_{\mathbf{P}} \otimes \mathsf{lu}_{\mathbf{Q}}) = \mathsf{ru}_{\mathbf{P}} \otimes \mathsf{id}_{\mathbf{Q}}.$$

This clearly follows from:

$$\begin{aligned}
&(\mathsf{as}_{\mathbf{P},\mathbf{1},\mathbf{Q}} \fatsemi (\mathsf{id}_{\mathbf{P}} \otimes \mathsf{lu}_{\mathbf{Q}}))^\star(\langle\langle p, \bullet\rangle, q\rangle)\\
&= \bigcup_{\langle r, s\rangle \in \mathsf{as}_{\mathbf{P},\mathbf{1},\mathbf{Q}}{}^\star(\langle\langle p, \bullet\rangle, q\rangle)} \mathsf{id}_{\mathbf{P}}{}^\star(r) \times \mathsf{lu}_{\mathbf{Q}}{}^\star(s)\\
&= \uparrow\{p\} \times \uparrow\{q\}\\
&= (\mathsf{ru}_{\mathbf{P}} \otimes \mathsf{id}_{\mathbf{Q}})^\star(\langle\langle p, \bullet\rangle, q\rangle).
\end{aligned}$$

Pentagon identity We want to show:

$$\mathsf{as}_{\mathbf{P}\times\mathbf{Q},\mathbf{R},\mathbf{S}} \fatsemi \mathsf{as}_{\mathbf{P},\mathbf{Q},\mathbf{R}\times\mathbf{S}} = (\mathsf{as}_{\mathbf{P},\mathbf{Q},\mathbf{R}} \otimes \mathsf{id}_{\mathbf{S}}) \fatsemi \mathsf{as}_{\mathbf{P},\mathbf{Q}\times\mathbf{R},\mathbf{S}} \fatsemi (\mathsf{id}_{\mathbf{P}} \otimes \mathsf{as}_{\mathbf{Q},\mathbf{R},\mathbf{S}}).$$

For the left-hand side, we have:

$$\begin{aligned}
&(\mathsf{as}_{\mathbf{P}\times\mathbf{Q},\mathbf{R},\mathbf{S}} \fatsemi \mathsf{as}_{\mathbf{P},\mathbf{Q},\mathbf{R}\times\mathbf{S}})^\star(\langle\langle\langle p, q\rangle, r\rangle, s\rangle)\\
&= \bigcup_{\langle\langle p', q'\rangle, \langle r', s'\rangle\rangle \in (\uparrow\{p\}\times\uparrow\{q\})\times(\uparrow\{r\}\times\uparrow\{s\})} (\uparrow\{p'\} \times (\uparrow\{q'\} \times (\uparrow\{r'\} \times \uparrow\{s'\})))\\
&= \uparrow\{p\} \times (\uparrow\{q\} \times (\uparrow\{r\} \times \uparrow\{s\})).
\end{aligned}$$

For the right-hand side, we have:

$$(\mathsf{as}_{\mathbf{P},\mathbf{Q},\mathbf{R}} \otimes \mathsf{id}_{\mathbf{S}})^\star(\langle\langle\langle p, q\rangle, r\rangle, s\rangle) = (\uparrow\{p\} \times (\uparrow\{q\} \times \uparrow\{r\})) \times \uparrow\{s\},$$

and

$$(\mathrm{id}_{\mathbf{P}} \otimes \mathrm{as}_{\mathbf{Q},\mathbf{R},\mathbf{S}})^{\star}(\langle p, \langle\langle q, r\rangle, s\rangle\rangle) = \uparrow\{p\} \times (\uparrow\{q\} \times (\uparrow\{r\} \times \uparrow\{s\})).$$

Therefore, the first piece evaluates to:

$$\begin{aligned}
&((\mathrm{as}_{\mathbf{P},\mathbf{Q},\mathbf{R}} \otimes \mathrm{id}_{\mathbf{S}}) \fatsemi \mathrm{as}_{\mathbf{P},\mathbf{Q}\times\mathbf{R},\mathbf{S}})^{\star}(\langle\langle\langle p, q\rangle, r\rangle, s\rangle) \\
&= \bigcup_{\langle\langle p', \langle q', r'\rangle\rangle, s'\rangle \in (\uparrow\{p\}\times(\uparrow\{q\}\times\uparrow\{r\}))\times\uparrow\{s\}} (\uparrow\{p'\} \times ((\uparrow\{q'\} \times \uparrow\{r'\}) \times \uparrow\{s'\})) \\
&= \uparrow\{p\} \times ((\uparrow\{q\} \times \uparrow\{r\}) \times \uparrow\{s\}).
\end{aligned}$$

Finally, we have:

$$\begin{aligned}
&((\mathrm{as}_{\mathbf{P},\mathbf{Q},\mathbf{R}} \otimes \mathrm{id}_{\mathbf{S}}) \fatsemi \mathrm{as}_{\mathbf{P},\mathbf{Q}\times\mathbf{R},\mathbf{S}} \fatsemi (\mathrm{id}_{\mathbf{P}} \otimes \mathrm{as}_{\mathbf{Q},\mathbf{R},\mathbf{S}}))^{\star}(\langle\langle\langle p, q\rangle, r\rangle, s\rangle) \\
&= \uparrow\{p\} \times (\uparrow\{q\} \times (\uparrow\{r\} \times \uparrow\{s\})),
\end{aligned}$$

confirming the pentagon identity. □

Lemma 16.1. Given posets $\mathbf{P}, \mathbf{Q}$, a monotone maps $f : \mathbf{P} \to \mathbf{Q}$, and a family of singleton sets $\{S_i\}_{i\in I}$, with $S_i = \{s_i\}$, $s_i \in \mathbf{P}$, the following equality holds:

$$\uparrow\Big(\bigcup_{p\in\uparrow\bigcup_{i\in I} S_i} \{f(p)\}\Big) = \uparrow\Big(\bigcup_{i\in I} \{f(s_i)\}\Big). \tag{79}$$

Proof. We first want to show that:

$$\underbrace{\uparrow\Big(\bigcup_{p\in\uparrow\bigcup_{i\in I} S_i} \{f(p)\}\Big)}_{\star} \subseteq \underbrace{\uparrow\Big(\bigcup_{i\in I} \{f(s_i)\}\Big)}_{\diamond}. \tag{80}$$

Take a

$$q \in \uparrow\Big(\bigcup_{p\in\uparrow\bigcup_{i\in I} S_i} \{f(p)\}\Big).$$

If we have such a q, it means that there exists a

$$q' \in \bigcup_{p\in\uparrow\bigcup_{i\in I} S_i} \{f(p)\}$$

such that $q' \preceq_{\mathbf{Q}} q$, and hence there is a $p' \in \uparrow\bigcup_{i\in I} S_i$ such that $q' = f(p')$. Consequently, there must exist an $i' \in I$ such that $s_{i'} \preceq_{\mathbf{P}} p'$. The monotonicity of f implies:

$$f(s_{i'}) \preceq_{\mathbf{P}} f(p') = q' \preceq_{\mathbf{Q}} q.$$

We know that $s_{i'} \in \diamond$ and any $q^* \in \mathbf{Q}$ satisfying $f(s_{i'}) \preceq_{\mathbf{Q}} q^*$ belongs to $\uparrow\diamond$. Therefore, $\star \subseteq \uparrow\diamond$, which proves the validity of (80).

We now want to show that:

$$\uparrow\Big(\bigcup_{p\in\uparrow\bigcup_{i\in I}S_i}\{f(p)\}\Big)\supseteq\uparrow\Big(\bigcup_{i\in I}\{f(s_i)\}\Big). \tag{81}$$

By now taking a

$$q\in\uparrow\Big(\bigcup_{i\in I}\{f(s_i)\}\Big),$$

we know that there is an $i'\in I$ such that $f(s_{i'})\preceq_{\mathbf{Q}} q$. Furthermore, we know that $f(s_{i'})\in\diamond$. Therefore, any $q^*\preceq_{\mathbf{Q}} f(s_{i'})$ must be in $\uparrow\diamond$, meaning that $q\in\star$, and proving the validity of (81).

The validity of (80) and (81) implies (79). □

Remark 16.2. Given posets $\mathbf{P},\mathbf{Q}$ and a monotone maps $f:\mathbf{P}\to\mathbf{Q}$, we have:

$$\uparrow\Big(\bigcup_{p'\in\uparrow\{p\}}\{f(p')\}\Big)=\uparrow\{f(p)\}.$$

This follows from Lemma 16.1, by considering a family of singleton sets consisting solely of the set $\{p\}$.

Proof of Lemma 7.14. We first show that the braiding defines an isomorphism. In other words, we want to show

$$(\mathsf{br}_{\mathbf{P},\mathbf{Q}}\fatsemi\mathsf{br}_{\mathbf{Q},\mathbf{P}})^\star=\mathsf{id}_{\mathbf{P}\times\mathbf{Q}}{}^\star.$$

We have

$$\begin{aligned}
&(\mathsf{br}_{\mathbf{P},\mathbf{Q}}\fatsemi\mathsf{br}_{\mathbf{Q},\mathbf{P}})^\star(p,q)\\
&=\bigcup_{\langle p',q'\rangle\in\mathsf{br}_{\mathbf{P},\mathbf{Q}}{}^\star(p,q)}\mathsf{br}_{\mathbf{Q},\mathbf{P}}{}^\star(p',q')\\
&=\bigcup_{\langle q',p'\rangle\in\uparrow\{q\}\times\uparrow\{p\}}\uparrow\{p'\}\times\uparrow\{q'\}\\
&=\uparrow\{p\}\times\uparrow\{q\}\\
&=\mathsf{id}_{\mathbf{P}\times\mathbf{Q}}{}^\star(p,q).
\end{aligned}$$

Note that this comes from the fact that br is an involution. We now show naturality. Consider $f:\mathbf{P}\to\mathbf{Q}$, $g:\mathbf{R}\to\mathbf{S}$. We have

$$\begin{aligned}
&((f\otimes g)\fatsemi\mathsf{br}_{\mathbf{Q},\mathbf{S}})^\star(p,r)\\
&=\big\langle f^\star(p),g^\star(r)\big\rangle\fatsemi\mathsf{br}_{\mathbf{Q},\mathbf{S}}\\
&=\Big\langle\bigcup_{r'\in g^\star(r)}\uparrow r',\bigcup_{p'\in f^\star(p)}\uparrow p'\Big\rangle.
\end{aligned}\tag{82}$$

On the other hand:

$$\begin{aligned}
&(\mathsf{br}_{\mathbf{S},\mathbf{Q}}\fatsemi(f\otimes g))^\star(p,r)\\
&=\langle\uparrow\{r\},\uparrow\{p\}\rangle\fatsemi(f\otimes g)^\star\\
&=\Big\langle\bigcup_{r'\in\uparrow\{r\}}g^\star(r'),\bigcup_{p'\in\uparrow\{p\}}f^\star(p')\Big\rangle.
\end{aligned}\tag{83}$$

Clearly, from Lemma 16.1 and Remark 16.2 we know that (82) and (83) are equivalent, proving naturality. □

Proof of Lemma 7.15. We now just need to show hexagon identities. First, we want to show that

$$(\mathsf{br}_{\mathbf{P},\mathbf{Q}} \otimes \mathsf{id}_{\mathbf{R}}) \fatsemi \mathsf{as}_{\mathbf{Q},\mathbf{P},\mathbf{R}} \fatsemi (\mathsf{id}_{\mathbf{Q}} \otimes \mathsf{br}_{\mathbf{P},\mathbf{R}}) = \mathsf{as}_{\mathbf{P},\mathbf{Q},\mathbf{R}} \fatsemi \mathsf{br}_{\mathbf{P},\mathbf{Q}\otimes\mathbf{R}} \fatsemi \mathsf{as}_{\mathbf{Q},\mathbf{R},\mathbf{P}} \tag{84}$$

To do so, we first look at the left-hand side of (84). We have

$$\begin{aligned}
&((\mathsf{br}_{\mathbf{P},\mathbf{Q}} \otimes \mathsf{id}_{\mathbf{R}}) \fatsemi \mathsf{as}_{\mathbf{Q},\mathbf{P},\mathbf{R}})^\star(\langle p, q\rangle, r) \\
&= \bigcup_{\langle\langle q', p'\rangle, r'\rangle \in (\mathsf{br}_{\mathbf{P},\mathbf{Q}}\otimes\mathsf{id}_{\mathbf{R}})^\star(\langle p, q\rangle, r)} \mathsf{as}^\star_{\mathbf{Q},\mathbf{P},\mathbf{R}}(q', p', r') \\
&= \bigcup_{\langle\langle q', p'\rangle, r'\rangle \in (\uparrow\{q\}\times\uparrow\{p\})\times\uparrow\{r\}} \uparrow\{q'\} \times (\uparrow\{p'\} \times \uparrow\{r'\}) \\
&= \uparrow\{q\} \times (\uparrow\{p\} \times \uparrow\{r\}).
\end{aligned}$$

Furthermore, we have

$$\begin{aligned}
&((\mathsf{br}_{\mathbf{P},\mathbf{Q}} \otimes \mathsf{id}_{\mathbf{R}}) \fatsemi \mathsf{as}_{\mathbf{Q},\mathbf{P},\mathbf{R}} \fatsemi (\mathsf{id}_{\mathbf{Q}} \otimes \mathsf{br}_{\mathbf{P},\mathbf{R}}))^\star(\langle p, q\rangle, r) \\
&= \bigcup_{\langle q', \langle p', r'\rangle\rangle \in \uparrow\{q\}\times(\uparrow\{p\}\times\uparrow\{r\})} (\mathsf{id}_{\mathbf{Q}} \otimes \mathsf{br}_{\mathbf{P},\mathbf{R}})^\star(q', \langle p', r'\rangle) \\
&= \bigcup_{\langle q', \langle p', r'\rangle\rangle \in \uparrow\{q\}\times(\uparrow\{p\}\times\uparrow\{r\})} \uparrow\{q'\} \times (\uparrow\{r'\} \times \uparrow\{p'\}) \\
&= \uparrow\{q\} \times (\uparrow\{r\} \times \uparrow\{p\}).
\end{aligned} \tag{85}$$

We now look at the right-hand side of (84). We have

$$\begin{aligned}
&\mathsf{as}_{\mathbf{P},\mathbf{Q},\mathbf{R}} \fatsemi \mathsf{br}_{\mathbf{P},\mathbf{Q}\otimes\mathbf{R}}{}^\star(\langle p, q\rangle, r) \\
&= \bigcup_{\langle p', \langle q', r'\rangle\rangle \in \mathsf{as}^\star_{\mathbf{P},\mathbf{Q},\mathbf{R}}(\langle p, q\rangle, r)} \mathsf{br}^\star_{\mathbf{P},\mathbf{Q}\otimes\mathbf{R}}(p', \langle q', r'\rangle) \\
&= \bigcup_{\langle p', \langle q', r'\rangle\rangle \in \uparrow\{p\}\times(\uparrow\{q\}\times\uparrow\{r\})} (\uparrow\{q'\} \times \uparrow\{r'\}) \times \uparrow\{p'\} \\
&= (\uparrow\{q\} \times \uparrow\{r\}) \times \uparrow\{p\}.
\end{aligned}$$

Furthermore, we have

$$\begin{aligned}
&(\mathsf{as}_{\mathbf{P},\mathbf{Q},\mathbf{R}} \fatsemi \mathsf{br}_{\mathbf{P},\mathbf{Q}\otimes\mathbf{R}} \fatsemi \mathsf{as}_{\mathbf{Q},\mathbf{R},\mathbf{P}})^\star(\langle p, q\rangle, r) \\
&= \bigcup_{\langle\langle q', r'\rangle, p'\rangle \in (\uparrow\{q\}\times\uparrow\{r\})\times\uparrow\{p\}} \mathsf{as}^\star_{\mathbf{Q},\mathbf{R},\mathbf{P}}(\langle q', r'\rangle, p') \\
&= \bigcup_{\langle\langle q', r'\rangle, p'\rangle \in (\uparrow\{q\}\times\uparrow\{r\})\times\uparrow\{p\}} \uparrow\{q'\} \times (\uparrow\{r'\} \times \uparrow\{p'\}) \\
&= \uparrow\{q\} \times (\uparrow\{r\} \times \uparrow\{p\}).
\end{aligned} \tag{86}$$

Clearly, since (85) and (86) are equal, the first hexagon identity is checked. The second hexagon identity can be checked analogously. □

Proof of Lemma 7.17. We have already checked that the constituents form a symmetric monoidal category. First, we check that the trace indeed returns a valid morphism in $\mathbf{Pos}_U$. Given any posets $\mathbf{P}, \mathbf{Q}, \mathbf{R}$

and morphisms $f : \mathbf{P} \times \mathbf{R} \to \mathbf{Q} \times \mathbf{R}$, and any $p \preceq p' \in \mathbf{P}$, we need to prove that

$$\frac{\mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{Q}}(f)(p) \preceq_{\mathbf{Pos}_U} \mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{Q}}(f)(p')}{\mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{Q}}(f)^{\star}(p) \supseteq \mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{Q}}(f)^{\star}(p')}$$

We know that $f^{\star}$ is a monotone map, meaning that

$$\frac{\langle q, r\rangle \in f^{\star}(p', r)}{\langle q, r\rangle \in f^{\star}(p, r)}$$

Therefore:

$$\frac{q \in \mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{Q}}(f)^{\star}(p')}{q \in \mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{Q}}{}^{\star}(f)(p)}$$

proving that $\mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{Q}}(f)^{\star}$ is a monotone function. Furthermore, due to the monotonicity of $f^{\star}$, for any $q \preceq q' \in \mathbf{Q}$, $p \in \mathbf{P}$, $r \in \mathbf{R}$, we have:

$$\frac{\langle q, r\rangle \in f^{\star}(p, r)}{\langle q', r\rangle \in f^{\star}(p, r)}$$

proving that $\mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{Q}}(f)^{\star}(p)$ is an upper set for all $p \in \mathbf{P}$. We now check the trace axioms one by one.

Naturality in P Given $f : \mathbf{P} \times \mathbf{R} \to \mathbf{Q} \times \mathbf{R}$ and $g : \mathbf{T} \to \mathbf{P}$, we want to prove:

$$\mathrm{Tr}^{\mathbf{R}}_{\mathbf{T},\mathbf{Q}}((g \otimes \mathrm{id}_{\mathbf{R}}) \fatsemi f) = g \fatsemi \mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{Q}}(f).$$

One has:

$$\begin{aligned}
&\mathrm{Tr}^{\mathbf{R}}_{\mathbf{T},\mathbf{Q}}((g \otimes \mathrm{id}_{\mathbf{R}}) \fatsemi f)^{\star}(t) \\
&= \left\{ q \in \mathbf{Q} \mid \bigvee_{r \in \mathbf{R}} \langle q, r\rangle \in ((g \otimes \mathrm{id}_{\mathbf{R}}) \fatsemi f)^{\star}(t, r) \right\} \\
&= \left\{ q \in \mathbf{Q} \mid \bigvee_{r \in \mathbf{R}} \langle q, r\rangle \in \bigcup_{\langle p, r'\rangle \in g^{\star}(t) \times \uparrow\{r\}} f^{\star}(p, r') \right\} \\
&= \left\{ q \in \mathbf{Q} \mid \bigvee_{r \in \mathbf{R}} \langle q, r\rangle \in \bigcup_{p \in g^{\star}(t)} f^{\star}(p, r) \right\}.
\end{aligned} \tag{87}$$

On the other hand, we have

$$\begin{aligned}(g \fatsemi \mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{Q}}(f))^{\star}(t) &= \bigcup_{p \in g^{\star}(t)} \mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{Q}}(f)^{\star}(p) \\ &= \bigcup_{p \in g^{\star}(t)} \left\{ q \in \mathbf{Q} \mid \bigvee_{r \in \mathbf{R}} \langle q, r \rangle \in f^{\star}(p, r) \right\} \\ &= \left\{ q \in \mathbf{Q} \mid \bigvee_{r \in \mathbf{R}} \langle q, r \rangle \in \bigcup_{p \in g^{\star}(t)} f^{\star}(p, r) \right\}. \end{aligned} \tag{88}$$

Clearly (87) and (88) are equivalent, proving the first naturality condition.

Naturality in $\mathbf{Q}$ Given $f : \mathbf{P} \times \mathbf{R} \to \mathbf{Q} \times \mathbf{R}$ and $g : \mathbf{Q} \to \mathbf{T}$, we want to prove:

$$\mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{T}}(f \fatsemi (g \otimes \mathrm{id}_{\mathbf{R}})) = \mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{Q}}(f) \fatsemi g.$$

One has:

$$\begin{aligned}&\mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{T}}(f \fatsemi (g \otimes \mathrm{id}_{\mathbf{R}}))^{\star}(p) \\ &= \left\{ t \in \mathbf{T} \mid \bigvee_{r \in \mathbf{R}} \langle t, r \rangle \in (f \fatsemi (g \otimes \mathrm{id}_{\mathbf{R}}))^{\star}(p, r) \right\} \\ &= \left\{ t \in \mathbf{T} \mid \bigvee_{r \in \mathbf{R}} \langle t, r \rangle \in \bigcup_{\langle q, r \rangle \in f^{\star}(p,r)} g^{\star}(q) \times \uparrow\{r\} \right\}\end{aligned}$$

On the other hand

$$(\mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{Q}}(f) \fatsemi g)^{\star}(p) = \bigcup_{q' \in \{q \in \mathbf{Q} \mid \bigvee_{r \in \mathbf{R}} \langle q, r \rangle \in f^{\star}(p,r)\}} g^{\star}(q'),$$

showing the equivalence.

Vanishing I Given $f : \mathbf{P} \times \mathbf{1} \to \mathbf{Q} \times \mathbf{1}$, we want to prove:

$$\mathrm{Tr}^{\mathbf{1}}_{\mathbf{P},\mathbf{Q}}(f) = \mathsf{ru}^{-1}_{\mathbf{P}} \fatsemi f \fatsemi \mathsf{ru}_{\mathbf{Q}}.$$

$$\begin{aligned}\mathrm{Tr}^{\mathbf{1}}_{\mathbf{P},\mathbf{Q}}(f)^{\star}(p) &= \{ q \in \mathbf{Q} \mid \langle q, \bullet \rangle \in (f \otimes \mathrm{id}_{\mathbf{1}})^{\star}(p, \bullet) \} \\ &= f^{\star}(p),\end{aligned}$$

which trivially proves the statement.

Vanishing II Given $f : (\mathbf{P} \times \mathbf{R}) \times \mathbf{S} \to (\mathbf{Q} \times \mathbf{R}) \times \mathbf{S}$, we want to prove:

$$\mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{Q}}(\mathrm{Tr}^{\mathbf{S}}_{\mathbf{P}\times\mathbf{R},\mathbf{Q}\times\mathbf{R}}(f)) = \mathrm{Tr}^{\mathbf{R}\times\mathbf{S}}_{\mathbf{P},\mathbf{Q}}(\mathsf{as}_{\mathbf{P},\mathbf{R},\mathbf{S}} \fatsemi f \fatsemi \mathsf{as}^{-1}_{\mathbf{Q},\mathbf{R},\mathbf{S}}).$$

We have:

$$\begin{aligned}&\mathrm{Tr}^{\mathbf{R}\times\mathbf{S}}_{\mathbf{P},\mathbf{Q}}(\mathsf{as}_{\mathbf{P},\mathbf{R},\mathbf{S}} \fatsemi f \fatsemi \mathsf{as}^{-1}_{\mathbf{Q},\mathbf{R},\mathbf{S}})^{\star}(p)\\ &= \left\{ q \in \mathbf{Q} \mid \bigvee_{\langle r,s\rangle \in \mathbf{R}\times\mathbf{S}} \langle\langle\langle q, r\rangle, s\rangle\rangle \in f^{\star}(\langle\langle p, r\rangle, s\rangle) \right\}\end{aligned} \tag{89}$$

Furthermore, we have:

$$\mathrm{Tr}^{\mathbf{S}}_{\mathbf{P}\times\mathbf{R},\mathbf{Q}\times\mathbf{R}}(f)^{\star}(p,r) = \left\{ \langle q, r\rangle \in \mathbf{Q}\times\mathbf{R} \mid \bigvee_{s\in\mathbf{S}} \langle\langle\langle q, r\rangle, s\rangle\rangle \in f^{\star}(\langle\langle p, r\rangle, s\rangle) \right\}.$$

Therefore, we can write:

$$\begin{aligned}&\left(\mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{Q}}\left(\mathrm{Tr}^{\mathbf{S}}_{\mathbf{P}\times\mathbf{R},\mathbf{Q}\times\mathbf{R}}(f)\right)\right)^{\star}(p)\\ &= \left\{ q \in \mathbf{Q} \mid \bigvee_{r\in\mathbf{R}} \langle q, r\rangle \in \mathrm{Tr}^{\mathbf{S}}_{\mathbf{P}\times\mathbf{R},\mathbf{Q}\times\mathbf{R}}(f)^{\star}(p,r) \right\}\\ &= \left\{ q \in \mathbf{Q} \mid \bigvee_{r\in\mathbf{R}} \langle q, r\rangle \in \left\{ \langle q', r'\rangle \in \mathbf{Q}\times\mathbf{R} \mid \bigvee_{s\in\mathbf{S}} \langle\langle q', r'\rangle, s\rangle \in f^{\star}(\langle\langle p, r'\rangle, s\rangle) \right\} \right\}\\ &= \left\{ q \in \mathbf{Q} \mid \bigvee r \in \mathbf{R} \left(\bigvee_{s\in\mathbf{S}} \langle\langle q, r\rangle, s\rangle \in f^{\star}(\langle\langle p, r\rangle, s\rangle) \right) \right\}\\ &= \left\{ q \in \mathbf{Q} \mid \bigvee_{\langle r,s\rangle\in\mathbf{R}\times\mathbf{S}} \langle\langle q, r\rangle, s\rangle \in f^{\star}(\langle\langle p, r\rangle, s\rangle) \right\}.\end{aligned} \tag{90}$$

Clearly, (89) and (90) are equivalent, proving the second vanishing axiom.

Superposing Given $f : \mathbf{P}\times\mathbf{R} \to \mathbf{Q}\times\mathbf{R}$, we want to prove:

$$\mathrm{Tr}^{\mathbf{R}}_{\mathbf{S}\times\mathbf{P},\mathbf{S}\times\mathbf{Q}}(\mathsf{as}_{\mathbf{S},\mathbf{P},\mathbf{R}} \fatsemi \mathrm{id}_{\mathbf{S}} \otimes f \fatsemi \mathsf{as}^{-1}_{\mathbf{S},\mathbf{Q},\mathbf{R}}) = \mathrm{id}_{\mathbf{S}} \otimes \mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{Q}}(f).$$

$$
\begin{aligned}
&\mathrm{Tr}^{\mathbf{R}}_{\mathbf{S}\times\mathbf{P},\mathbf{S}\times\mathbf{Q}}(\mathsf{as}_{\mathbf{S},\mathbf{P},\mathbf{R}} \fatsemi \mathrm{id}_{\mathbf{S}} \otimes f \fatsemi \mathsf{as}^{-1}_{\mathbf{S},\mathbf{Q},\mathbf{R}})^{\star}(s,p) \\
&= \left\{ \langle s,q\rangle \in \mathbf{S}\times\mathbf{Q} \mid \bigvee_{r\in\mathbf{R}} \langle s,q,r\rangle \in (\mathrm{id}_{\mathbf{S}} \otimes f)^{\star}(s,p,r) \right\} \\
&= \left\{ \langle s,q\rangle \in \mathbf{S}\times\mathbf{Q} \mid \bigvee_{r\in\mathbf{R}} (s \in \mathrm{id}_{\mathbf{S}}{}^{\star}(s)) \wedge (\langle q,r\rangle \in f^{\star}(p,r)) \right\} \\
&= \left\{ \langle s,q\rangle \in \mathbf{S}\times\mathbf{Q} \mid \bigvee_{r\in\mathbf{R}} (s \in \uparrow\{s\}) \wedge (\langle q,r\rangle \in f^{\star}(p,r)) \right\} \\
&= \left\{ \langle s,q\rangle \in \uparrow\{s\}\times\mathbf{Q} \mid \bigvee_{r\in\mathbf{R}} \langle q,r\rangle \in f^{\star}(p,r) \right\} \\
&= \uparrow\{s\} \times \left\{ q \in \mathbf{Q} \mid \bigvee_{r\in\mathbf{R}} \langle q,r\rangle \in f^{\star}(p,r) \right\}
\end{aligned}
\tag{91}
$$

On the other hand, we have:

$$
(\mathrm{id}_{\mathbf{S}} \otimes \mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{Q}}(f))^{\star}(s,p) = \uparrow\{s\} \times \left\{ q \in \mathbf{Q} \mid \bigvee_{r\in\mathbf{R}} \langle q,r\rangle \in f^{\star}(p,r) \right\}. \tag{92}
$$

Clearly, (91) and (92) are equivalent, proving the superposing axiom.

Yanking We want to prove:

$$
\mathrm{Tr}^{\mathbf{P}}_{\mathbf{P},\mathbf{P}}(\mathsf{br}_{\mathbf{P},\mathbf{P}}) = \mathrm{id}_{\mathbf{P}}.
$$

We have

$$
\begin{aligned}
&\mathrm{Tr}^{\mathbf{P}}_{\mathbf{P},\mathbf{P}}(\mathsf{br}_{\mathbf{P},\mathbf{P}})^{\star}(p) \\
&= \{ p' \in \mathbf{P} \mid \bigvee_{p''\in\mathbf{P}} \langle p',p''\rangle \in \mathsf{br}^{\star}_{\mathbf{P},\mathbf{P}}(p,p'') \} \\
&= \{ p' \in \mathbf{P} \mid \bigvee_{p''\in\mathbf{P}} \langle p',p''\rangle \in \uparrow\{p''\}\times\uparrow\{p\} \} \\
&= \{ p' \in \mathbf{P} \mid \bigvee_{p''\in\mathbf{P}} (p' \in \uparrow\{p''\}) \wedge (p'' \in \uparrow\{p\}) \} \\
&= \{ p' \in \mathbf{P} \mid p' \in \uparrow\{p\} \} \\
&= \uparrow\{p\} \\
&= \mathrm{id}^{\star}_{\mathbf{P}}(p),
\end{aligned}
$$

proving the yanking axiom. □

Proof of Lemma 7.21. First, we need to prove that $\mathrm{Hom}_{\mathbf{Pos}_U}(\mathbf{P};\mathbf{Q})$ forms a poset. To prove this, we check the following, using the order defined previously:

- *Reflexivity:* Given $f \in \mathrm{Hom}_{\mathbf{Pos}_U}(\mathbf{P};\mathbf{Q})$, we can write

$$f^\star(p) \supseteq f^\star(p), \quad \forall p \in \mathbf{P},$$

  which implies $f \preceq_{\mathbf{Pos}_U} f$.
- *Antisymmetry:* Consider

$$f, g \in \mathrm{Hom}_{\mathbf{Pos}_U}(\mathbf{P};\mathbf{Q})$$

  with $f \preceq_{\mathbf{Pos}_U} g$ and $g \preceq_{\mathbf{Pos}_U} f$. We know

$$(f \preceq_{\mathbf{Pos}_U} g) \Rightarrow f^\star(p) \supseteq g^\star(p), \quad \forall p \in \mathbf{P},$$

  but also

$$(g \preceq_{\mathbf{Pos}_U} f) \Rightarrow g^\star(p) \supseteq f^\star(p), \quad \forall p \in \mathbf{P},$$

  implying $f = g$.
- *Transitivity:* Consider

$$f, g, h \in \mathrm{Hom}_{\mathbf{Pos}_U}(\mathbf{P};\mathbf{Q})$$

  with $f \preceq_{\mathbf{Pos}_U} g$ and $g \preceq_{\mathbf{Pos}_U} h$. We have, for all $p \in \mathbf{P}$,

$$\begin{aligned}(f^\star(p) \supseteq g^\star(p)) \wedge (g^\star(p) \supseteq h^\star(p)) &\Rightarrow f^\star(p) \supseteq h^\star(p) \\ &\Rightarrow f \preceq_{\mathbf{Pos}_U} h.\end{aligned}$$

Consider now $f, g \in \mathrm{Hom}_{\mathbf{Pos}_U}(\mathbf{P};\mathbf{Q})$. Their least upper bound (join) is $f \wedge g$, since it is the least morphism such that $f \preceq_{\mathbf{Pos}_U} (f \wedge g)$ and $g \preceq_{\mathbf{Pos}_U} (f \wedge g)$. Their greatest lower bound (meet) is $f \vee g$, since it is the greatest morphism such that $(f \vee g) \preceq_{\mathbf{Pos}_U} f$ and $(f \vee g) \preceq_{\mathbf{Pos}_U} g$. Furthermore, for any $f \in \mathrm{Hom}_{\mathbf{Pos}_U}(\mathbf{P};\mathbf{Q})$, one will have, for all $p \in \mathbf{P}$

$$f^\star(p) \supseteq \emptyset = \top_{\mathrm{Hom}_{\mathbf{Pos}_U}(\mathbf{P};\mathbf{Q})}{}^\star(p),$$

implying that for all $f \in \mathrm{Hom}_{\mathbf{Pos}_U}(\mathbf{P};\mathbf{Q})$ we have $f \preceq_{\mathbf{Pos}_U} \top_{\mathrm{Hom}_{\mathbf{Pos}_U}(\mathbf{P};\mathbf{Q})}$. Finally, for any $f \in \mathrm{Hom}_{\mathbf{Pos}_U}(\mathbf{P};\mathbf{Q})$, one will have, for all $p \in \mathbf{P}$

$$\bot_{\mathrm{Hom}_{\mathbf{Pos}_U}(\mathbf{P};\mathbf{Q})}{}^\star(p) = \mathbf{Q} \supseteq f^\star(p)$$

implying that for all $f \in \mathrm{Hom}_{\mathbf{Pos}_U}(\mathbf{P};\mathbf{Q})$ we have $\bot_{\mathrm{Hom}_{\mathbf{Pos}_U}(\mathbf{P};\mathbf{Q})} \preceq_{\mathbf{Pos}_U} f$. □

Proof of Lemma 7.23. We prove the two conditions.

Preservation of identities We have

$$\begin{aligned}\mathsf{FixFunMinRes}(\mathrm{id}_{\mathbf{P}}^{\mathbf{DP}})^{\star}(p) &= \{q \in \mathbf{Q} \mid \mathrm{id}_{\mathbf{P}}^{\mathbf{DP}}(p^*, q)\}\\ &= \{q \in \mathbf{Q} \mid p \preceq q\}\\ &= \uparrow\{p\}\\ &= \mathrm{id}_{\mathbf{P}}^{\mathbf{Pos}_U\,\star}(p).\end{aligned}$$

Preservation of composition On one hand, we have

$$\begin{aligned}\mathsf{FixFunMinRes}(\mathbf{d} \fatsemi_{\mathbf{DP}} \mathbf{e})^{\star}(p) &= \{r \in \mathbf{R} \mid (\mathbf{d} \fatsemi \mathbf{e})(p^*, r)\}\\ &= \{r \in \mathbf{R} \mid \bigvee_{q \in \mathbf{Q}} \mathbf{d}(p^*, q) \wedge \mathbf{e}(q^*, r)\}.\end{aligned} \tag{93}$$

On the other hand:

$$\begin{aligned}&(\mathsf{FixFunMinRes}(\mathbf{d}) \fatsemi_{\mathbf{Pos}_U} \mathsf{FixFunMinRes}(\mathbf{e}))^{\star}(p)\\ &= \bigcup\nolimits_{q \in \mathsf{FixFunMinRes}(\mathbf{d})^{\star}(p)} \mathsf{FixFunMinRes}(\mathbf{e})^{\star}(q)\\ &= \bigcup\nolimits_{q \in \{q \in \mathbf{Q} \mid \mathbf{d}(p^*, q)\}} \{r \in \mathbf{R} \mid \mathbf{e}(q^*, r)\}\\ &= \{r \in \mathbf{R} \mid (q \in \mathbf{Q}) \wedge \mathbf{d}(p^*, q) \wedge \mathbf{e}(q^*, r)\}\\ &= \{r \in \mathbf{R} \mid \bigvee_{q \in \mathbf{Q}} \mathbf{d}(p^*, q) \wedge \mathbf{e}(q^*, r)\}.\end{aligned} \tag{94}$$

Clearly, (93) and (94) coincide. □

Proof of Lemma 7.25. We prove the two conditions.

Preservation of identities We have

$$\begin{aligned}\mathsf{FixFunMinResBack}(\mathrm{id}_{\mathbf{P}}^{\mathbf{Pos}_U})(p^*, q) &= q \in \mathrm{id}_{\mathbf{Q}}^{\mathbf{Pos}_U\,\star}\\ &= q \in \uparrow\{p\}\\ &= \mathrm{id}_{\mathbf{P}}^{\mathbf{DP}}(p^*, q).\end{aligned}$$

Preservation of composition On one hand, we have

$$\begin{aligned}\mathsf{FixFunMinResBack}(f \fatsemi_{\mathbf{Pos}_U} g)(p^*, r) &= r \in (f \fatsemi_{\mathbf{Pos}_U} g)^{\star}(p)\\ &= r \in \bigcup\nolimits_{q \in f^{\star}(p)} g^{\star}(q).\end{aligned} \tag{95}$$

On the other hand:

$$\begin{aligned}&(\mathsf{FixFunMinResBack}(f) \fatsemi_{\mathbf{DP}} \mathsf{FixFunMinResBack}(g))(p^*, r)\\ &= \bigvee_{q \in \mathbf{Q}} (q \in f^\star(p)) \wedge (r \in g^\star(q))\\ &= r \in \bigcup_{q \in f^\star(p)} g^\star(q).\end{aligned} \tag{96}$$

Clearly, (95) and (96) coincide. □

Proof of Lemma 7.27. First, consider any morphism in $\mathrm{Hom}_{\mathbf{DP}}(\mathbf{P}; \mathbf{Q})$. We have

$$\begin{aligned}&(\mathsf{FixFunMinRes} \fatsemi \mathsf{FixFunMinResBack})(\mathbf{d})(p^*, q)\\ &= q \in \mathsf{FixFunMinRes}(\mathbf{d})^\star(p)\\ &= q \in \{q' \in \mathbf{Q} \mid \mathbf{d}(p^*, q)\}\\ &= \mathbf{d}(p^*, q)\\ &= \mathrm{id}_{\mathbf{DP}}(\mathbf{d})(p^*, q).\end{aligned}$$

Now consider any morphism $H_{\mathbf{d}} \in \mathrm{Hom}_{\mathbf{Pos}_U}(\mathbf{P}; \mathbf{Q})$. We have

$$\begin{aligned}&(\mathsf{FixFunMinResBack} \fatsemi \mathsf{FixFunMinRes})(H_{\mathbf{d}})(p)\\ &= \{q \in \mathbf{Q} \mid q \in {H_{\mathbf{d}}}^\star(p)\}\\ &= \{q \in \mathbf{Q} \mid \mathbf{d}(p^*, q)\}\\ &= {\mathrm{id}_{\mathbf{Pos}_U}}^\star(H_{\mathbf{d}})(p).\end{aligned}$$

These are clearly natural, proving the statement. □

Proof of Lemma 7.28. Given $\mathbf{P}, \mathbf{Q} \in \mathrm{Ob}_{\mathbf{DP}}$ and $\mathbf{d}, \mathbf{e} \in \mathrm{Hom}_{\mathbf{DP}}(\mathbf{P}; \mathbf{Q})$, we want to check the following properties.

Order reversing We want to check

$$\frac{\mathbf{d} \preceq_{\mathbf{DP}} \mathbf{e}}{\mathsf{FixFunMinRes}(\mathbf{d}) \succeq_{\mathbf{Pos}_U} \mathsf{FixFunMinRes}(\mathbf{e})}.$$

We have:

$$\begin{aligned}\mathsf{FixFunMinRes}(\mathbf{d})^\star(p) &= \{q \in \mathbf{Q} \mid \mathbf{d}(p^*, q)\}\\ &\subseteq \{q \in \mathbf{Q} \mid \mathbf{e}(p^*, q)\}\\ &= \mathsf{FixFunMinRes}(\mathbf{e})^\star(p),\end{aligned}$$

implying $\mathsf{FixFunMinRes}(\mathbf{d}) \succeq_{\mathbf{Pos}_U} \mathsf{FixFunMinRes}(\mathbf{e})$.

Meet and join preservation We want to check

$$\mathsf{FixFunMinRes}(\mathbf{d} \wedge \mathbf{e}) = \mathsf{FixFunMinRes}(\mathbf{d}) \wedge_{\mathbf{Pos}_U} \mathsf{FixFunMinRes}(\mathbf{e}),$$

and

$$\mathsf{FixFunMinRes}(\mathbf{d} \vee \mathbf{e}) = \mathsf{FixFunMinRes}(\mathbf{d}) \vee_{\mathbf{Pos}_U} \mathsf{FixFunMinRes}(\mathbf{e}).$$

We have

$$\begin{aligned}
&\mathsf{FixFunMinRes}(\mathbf{d} \wedge \mathbf{e})^\star(p) \\
&= \{q \in \mathbf{Q} \mid (\mathbf{d} \wedge \mathbf{e})(p^*, q)\} \\
&= \{q \in \mathbf{Q} \mid (\mathbf{d}(p^*, q) \wedge_{\mathbf{DP}} \mathbf{e}(p^*, q))\} \\
&= \{q \in \mathbf{Q} \mid \mathbf{d}(p^*, q)\} \cap \{q \in \mathbf{Q} \mid \mathbf{e}(p^*, q)\} \\
&= \mathsf{FixFunMinRes}(\mathbf{d})^\star(p) \wedge_{\mathbf{Pos}_U} \mathsf{FixFunMinRes}(\mathbf{e})^\star(p).
\end{aligned}$$

Similarly:

$$\begin{aligned}
&\mathsf{FixFunMinRes}(\mathbf{d} \vee \mathbf{DP}\mathbf{e})^\star(p) \\
&= \{q \in \mathbf{Q} \mid (\mathbf{d} \vee \mathbf{e})(p^*, q)\} \\
&= \{q \in \mathbf{Q} \mid (\mathbf{d}(p^*, q) \vee_{\mathbf{DP}} \mathbf{e}(p^*, q))\} \\
&= \{q \in \mathbf{Q} \mid \mathbf{d}(p^*, q)\} \cup \{q \in \mathbf{Q} \mid \mathbf{e}(p^*, q)\} \\
&= \mathsf{FixFunMinRes}(\mathbf{d})^\star(p) \vee_{\mathbf{Pos}_U} \mathsf{FixFunMinRes}(\mathbf{e})^\star(p).
\end{aligned}$$

Top and bottom preservation We want to check

$$\mathsf{FixFunMinRes}(\bot_{\mathrm{Hom}_{\mathbf{DP}}(\mathbf{P};\mathbf{Q})}) = \bot_{\mathrm{Hom}_{\mathbf{Pos}_U}(\mathbf{P};\mathbf{Q})},$$

and

$$\mathsf{FixFunMinRes}(\top_{\mathrm{Hom}_{\mathbf{DP}}(\mathbf{P};\mathbf{Q})}) = \top_{\mathrm{Hom}_{\mathbf{Pos}_U}(\mathbf{P};\mathbf{Q})}.$$

We have

$$\begin{aligned}
\mathsf{FixFunMinRes}(\bot_{\mathrm{Hom}_{\mathbf{DP}}(\mathbf{P};\mathbf{Q})})^\star(p) &= \emptyset \\
&= \bot_{\mathrm{Hom}_{\mathbf{Pos}_U}(\mathbf{P};\mathbf{Q})}{}^\star(p)
\end{aligned}$$

Similarly

$$\begin{aligned}
\mathsf{FixFunMinRes}(\top_{\mathrm{Hom}_{\mathbf{DP}}(\mathbf{P};\mathbf{Q})})^\star(p) &= \mathbf{Q} \\
&= \top_{\mathrm{Hom}_{\mathbf{Pos}_U}(\mathbf{P};\mathbf{Q})}{}^\star(p).
\end{aligned}$$

□

Proof of Lemma 7.29. We want to show that

$$\mathsf{FixFunMinRes}(\mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{Q}}(\mathbf{d})) = \mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{Q}}(\mathsf{FixFunMinRes}(\mathbf{d})),$$

for all $\mathbf{d} \in \mathrm{Hom}_{\mathbf{DP}}(\mathbf{P} \times \mathbf{R}; \mathbf{Q} \times \mathbf{R})$, and $\mathbf{P}, \mathbf{Q}, \mathbf{R} \in \mathrm{Ob}_{\mathbf{DP}}$. On one hand, we have

$$\begin{aligned}\mathsf{FixFunMinRes}(\mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{Q}}(\mathbf{d}))^{\star}(p) &= \{q \in \mathbf{Q} \mid \mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{Q}}(\mathbf{d})(p^*, q)\} \\ &= \{q \in \mathbf{Q} \mid \bigvee_{r \in \mathbf{R}} \mathbf{d}(\langle p, r\rangle^*, \langle q, r\rangle)\}\end{aligned}$$

On the other hand, we have

$$\begin{aligned}&\mathrm{Tr}^{\mathbf{R}}_{\mathbf{P},\mathbf{Q}}(\mathsf{FixFunMinRes}(\mathbf{d}))^{\star}(p) \\ &= \{q \in \mathbf{Q} \mid \bigvee_{r \in \mathbf{R}} \langle q, r\rangle \in \mathsf{FixFunMinRes}(\mathbf{d})^{\star}(p, r)\} \\ &= \{q \in \mathbf{Q} \mid \bigvee_{r \in \mathbf{R}} \langle q, r\rangle \in \{\langle q, r\rangle \in \mathbf{Q} \times \mathbf{R} \mid \mathbf{d}(\langle p, r\rangle^*, \langle q, r\rangle)\}\} \\ &= \{q \in \mathbf{Q} \mid \bigvee_{r \in \mathbf{R}} \mathbf{d}(\langle p, r\rangle^*, \langle q, r\rangle)\}.\end{aligned}$$

□

Proof of Lemma 7.31. On the objects, we have:

$$\begin{aligned}\mathsf{FixFunMinRes}(\mathbf{P}) \otimes_{\mathbf{Pos}_U} \mathsf{FixFunMinRes}(\mathbf{Q}) &= \mathbf{P} \otimes_{\mathbf{Pos}_U} \mathbf{Q} \\ &= \mathbf{P} \times \mathbf{Q} \\ &= \mathbf{P} \otimes_{\mathbf{DP}} \mathbf{Q} \\ &= \mathsf{FixFunMinRes}(\mathbf{P} \otimes_{\mathbf{DP}} \mathbf{Q})\end{aligned}$$

Hence, showing associativity and unitality is equivalent to showing such properties for a category with posets as objects, as **DP** is monoidal. □

Proof of Lemma 7.35. Given the finite design problems $\mathbf{d} : \mathbf{P} \nrightarrow \mathbf{Q}$, $\mathbf{e} : \mathbf{Q} \nrightarrow \mathbf{R}$, recall from Def. 7.5 that the representations compose as follows:

$$(H_{\mathbf{d}} \fatsemi H_{\mathbf{e}})(p) = \bigcup_{q \in H_{\mathbf{d}}(p)} H_{\mathbf{e}}(q).$$

From the above expression, $q \in \overline{U}_{\mathrm{f}}\mathbf{Q}$ by finiteness of $\mathbf{d}$, and $H_{\mathbf{e}}(q) \in \overline{U}_{\mathrm{f}}\mathbf{R}$ by finiteness of $\mathbf{e}$. □

Proof of Lemma 7.36. Given the finite design problems $\mathbf{d} : \mathbf{P} \nrightarrow \mathbf{Q}$, $\mathbf{e} : \mathbf{R} \nrightarrow \mathbf{S}$, recall from Section 7.2 that the representations compose as follows:

$$(H_{\mathbf{d}} \otimes H_{\mathbf{e}})(p, r) = H_{\mathbf{d}}(p) \times H_{\mathbf{e}}(r),$$

which clearly belongs to $\overline{U}_{\mathrm{f}}\mathbf{Q} \times \overline{U}_{\mathrm{f}}\mathbf{S}$. □

Proof of Lemma 7.44. Consider a map $f : \mathbf{P} \to_{\mathbf{Pos}} \mathbf{Q}$ that is Scott continuous. Take two elements $x, y \in \mathbf{P}$ such that $x \preceq y$. The set $\mathbf{S} = \{x, y\}$ is directed. From (54), we know that

$$f(\mathsf{Sup}\,\mathbf{S}) = f(y) = \mathsf{Sup}\,\{f(x), f(y)\},$$

which implies that $f(x) \preceq f(y)$. Therefore, f is monotone. □

Proof of Theorem 7.50. This proof is adapted from [122]. The map $h_{\mathsf{loop}(\mathbf{d})}$ can be described as:

$$h_{\mathsf{loop}(\mathbf{d})} : \mathbf{F}_1 \to \mathsf{Anti}_{\mathsf{f}}\,\mathbf{R}, \tag{97}$$

$$f_1 \mapsto \begin{cases} \text{using} & r, f_2 \in \mathbf{R}, \\ \mathsf{Min}_{\preceq_{\mathbf{R}}} & r, \\ \text{s.t.} & r \in h_{\mathbf{d}}(f_1, f_2), \\ & r \preceq_{\mathbf{R}} f_2. \end{cases} \tag{98}$$

Denote by h_{f_1} the map $h_{\mathbf{d}}$ with the first element fixed:

$$h_{f_1} : f_2 \mapsto h_{\mathbf{d}}(f_1, f_2).$$

Rewrite $r \in h_{\mathbf{d}}(f_1, f_2)$ in (97) as

$$r \in h_{f_1}(f_2). \tag{99}$$

Let r be a feasible solution, but not necessarily minimal. Lemma 7.51 implies that the constraint (99) can be rewritten as

$$\{r\} = h_{f_1}(f_2) \cap \uparrow r.$$

Because $f_2 \succeq r$, and h_{f_1} is Scott continuous, it follows that $h_{f_1}(f_2) \succeq_{\mathsf{Anti}_{\mathsf{f}}\,\mathbf{R}} h_{f_1}(r)$. Therefore, by Lemma 7.52, we have

$$\{r\} \succeq_{\mathsf{Anti}_{\mathsf{f}}\,\mathbf{R}} h_{f_1}(r) \cap \uparrow r. \tag{100}$$

This is a recursive condition that all feasible r must satisfy.

Let $\alpha \in \mathsf{Anti}_{\mathsf{f}}\,\mathbf{R}$ be an antichain of feasible resources, and let r be a generic element of $\mathbf{R}$. Tautologically, rewrite α as the minimal elements of the union of the singletons containing its elements:

$$\alpha = \operatorname*{Min}_{\preceq_{\mathbf{R}}} \bigcup_{r \in \alpha} \{r\}. \tag{101}$$

Substituting (100) in (101) we obtain (cf Lemma 7.53)

$$\alpha \succeq_{\mathsf{Anti}_{\mathsf{f}}\,\mathbf{R}} \operatorname*{Min}_{\preceq_{\mathbf{R}}} \bigcup_{r \in \alpha} h_{f_1}(r) \cap \uparrow r. \tag{102}$$

Converse: It is also true that if an antichain α satisfies (102) then all $r \in \alpha$ are feasible. The con-

straint (102) means that for any $r_0 \in \alpha$ on the left side, we can find a r_1 on the right side so that $r_0 \succeq_{\mathbf{R}} r_1$. The point r_1 needs to belong to one of the sets of which we take the union; say that it comes from $r_2 \in \alpha$, so that $r_1 \in h_{f_1}(r_2) \cap \uparrow r_2$. Summarizing:

$$\forall r_0 \in \alpha : \ \exists r_1 : \ (r_0 \succeq_{\mathbf{R}} r_1) \wedge (\exists r_2 \in \alpha : \ r_1 \in h_{f_1}(r_2) \cap \uparrow r_2). \tag{103}$$

Because $r_1 \in h_{f_1}(r_2) \cap \uparrow r_2$, we can conclude that $r_1 \in \uparrow r_2$, and therefore $r_1 \succeq_{\mathbf{R}} r_2$, which together with $r_0 \succeq_{\mathbf{R}} r_1$, implies $r_0 \succeq_{\mathbf{R}} r_2$. We have concluded that there exist two points r_0, r_2 in the antichain α such that $r_0 \succeq_{\mathbf{R}} r_2$; therefore, they are the same point: $r_0 = r_2$. Because $r_0 \succeq_{\mathbf{R}} r_1 \succeq_{\mathbf{R}} r_2$, we also conclude that r_1 is the same point as well. We can rewrite (103) by using r_0 in place of r_1 and r_2 to obtain $\forall r_0 \in \alpha : \ r_0 \in h_{f_1}(r_0)$, which means that r_0 is a feasible resource.

We have concluded that all antichains of feasible resources α satisfy (102), and conversely, if an antichain α satisfies (102), then it is an antichain of feasible resources.

Equation (102) is a recursive constraint for α, of the kind

$$\Phi_{f_1}(\alpha) \preceq_{\mathsf{Anti_f}\,\mathbf{R}} \alpha,$$

with the map Φ_{f_1} defined by

$$\begin{aligned} \Phi_{f_1} : \mathsf{Anti_f}\,\mathbf{R} &\to \mathsf{Anti_f}\,\mathbf{R}, \\ \alpha &\mapsto \operatorname*{Min}_{\preceq_{\mathbf{R}}} \bigcup_{r \in \alpha} h_{f_1}(r) \cap \uparrow r. \end{aligned} \tag{104}$$

If we want the *minimal* resources, we are looking for the *least* antichain:

$$\min_{\preceq_{\mathsf{Anti_f}\,\mathbf{R}}} \{\alpha \in \mathsf{Anti_f}\,\mathbf{R} : \ \Phi_{f_1}(\alpha) \preceq_{\mathsf{Anti_f}\,\mathbf{R}} \alpha\},$$

which is equal to the *least fixed point* of Φ_{f_1}. Therefore, the map $h_{\mathsf{loop}(\mathbf{d})}$ can be written as

$$h_{\mathsf{loop}(\mathbf{d})} : \ f_1 \mapsto \mathsf{lfp}(\Phi_{f_1}).$$

Lemma 7.54 shows that $\mathsf{lfp}(\Phi_{f_1})$ is Scott continuous in f_1. □

Proof of Prop. 7.55. This proof is extracted from [122]. The memory utilization is bounded by $\mathsf{width}(\mathbf{R}_0)$, because the state is an antichain, and $\mathsf{width}(\mathbf{R}_0)$ is the size of the largest antichain. The iteration happens in the space $\mathsf{Anti}\,\mathbf{R}_0$, and we are constructing an ascending chain, so it can take at most $\mathsf{height}(\mathsf{Anti}\,\mathbf{R}_0)$ steps to converge. Finally, in the worst case the map h_0 needs to be evaluated once for each element of the antichain for each step. □

16.2 Proofs related to Part B

Proof sketch of Lemma 10.1. One can prove the statement by induction. The monotonicity holds in the initialization. To prove the induction step, one can first write the prediction update, and leverage properties of the differential Lyapunov equation involved. Finally, one uses the results of the induction step in the measurement update, to prove the result. □

Proof sketch of Lemma 10.2. The statement can be proven by following the *substitution principle* [134]. If the estimator is given a set of observations, it can simulate having less (i.e., having a higher dropping probability) by artificially ignoring selected samples. This can also be proven analytically, by comparing measurement updates in the EKF in the two cases. □

Proof sketch of Lemma 10.3. First, one can derive the error dynamics

$$\begin{aligned}\dot{e}_p(t) &= v_{\mathrm{f}} \sin(\theta_e(t) - \delta(t)) + \rho_e(t),\\ \dot{\theta}_e(t) &= -v_{\mathrm{f}} \sin(\delta(t))/L + \boldsymbol{w}_\theta,\end{aligned}$$

where ρ_e and $\boldsymbol{w}_\theta$ are Brownian processes as per given models. By leveraging properties of the system and measurement noises, one can then show that at any time instant, by larger $\mathbf{W}$ or $\mathbf{P}$, one cannot obtain a smaller expected total lateral error (to parity of initial condition). □

Proof sketch of Lemma 10.4. The expected lateral positional and orientation errors converging more rapidly to zero imply a commanded steering angle converging more rapidly to zero in expectation. This can be formally proven by taking the expectation of the steering angle formulation as presented in the Stanley part. □

Proof sketch of Lemma 10.6. This can be shown by following the procedure in Lemma 10.5 and looking at the behavior of $\delta(t)$. □

Proof sketch of Lemma 10.5. First, one can derive the lateral error dynamics

$$\dot{e}_p = -v_{\mathrm{r}} e_{\mathrm{along}}(t) \sin(\delta(t))/L + \rho_e(t),$$

where ρ_e is a Brownian process and

$$e_{\mathrm{along}}(t) = \begin{bmatrix} x_{\mathrm{t}} - x(t) & y_{\mathrm{t}} - y(t) \end{bmatrix} \cdot \begin{bmatrix} \cos(\theta(t)) \\ \sin(\theta(t)) \end{bmatrix}.$$

Then, one can leverage properties of the system and measurement noises to prove the statement. □

Proof sketch of Lemma 10.7 and Lemma 10.8. One can derive the expected error dynamics for the original, nonlinear system, and leveraging properties of the noise perturbations, one can derive both monotonicity results. □

Proof sketch of Lemma 10.9 and Lemma 10.10. First, one can derive the error dynamics, which are equivalent to Section 16.2. One can prove that at each step, the map describing the dependency of the optimal (initial) control input on the initial lateral error is s-shaped (for positive definite $\mathbf{Q}$). Furthermore, in the presence of heading error, the s-shaped curve is translated proportionally along the input axis. From these facts, one can prove the statements. □

Proof sketch of Lemma 10.11. One can easily write the expression for the expected value of the speed tracking error between any two control steps. Given the properties of the perturbations in the system model and in the measurement model, one can show that this expression is monotonic in system noise and state estimation uncertainty. □

Proof sketch of Lemma 10.12. By explicitly looking at the expression for the expected value of the acceleration resulting from the control, one proceed as for Lemma 10.11 and show the monotonicity. □

Proof of Lemma 11.2. Consider $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3 \in \mathbf{G}$ with $\mathcal{G}_i = \langle \mathbf{V}_i, \mathbf{A}_i, c_i \rangle$, and $c_i : \mathbf{A}_i \to \mathbf{R}$. Clearly $\mathcal{G}_1 \preceq_{\mathbf{G}} \mathcal{G}_1$, since $\mathbf{V}_1 \subseteq \mathbf{V}_1$, $\mathbf{A}_1 \subseteq \mathbf{A}_1$, and $c_1 \succeq_{\mathbf{R}^{\mathbf{A}_1}} c_1$. Furthermore, given $\mathcal{G}_1 \preceq_{\mathbf{G}} \mathcal{G}_2$ and $\mathcal{G}_2 \preceq_{\mathbf{G}} \mathcal{G}_3$ (i.e., $\mathbf{V}_1 \subseteq \mathbf{V}_2 \subseteq \mathbf{V}_3$, $\mathbf{A}_1 \subseteq \mathbf{A}_2$, $\mathbf{A}_2 \subseteq \mathbf{A}_3$, $c_1 \succeq_{\mathbf{R}^{\mathbf{A}_1}} c_2|_{\mathbf{A}_1}$, and $c_2 \succeq_{\mathbf{R}^{\mathbf{A}_2}} c_3|_{\mathbf{A}_2}$), one has $\mathbf{V}_1 \subseteq \mathbf{V}_3$, $\mathbf{A}_1 \subseteq \mathbf{A}_3$, and $c_1 \succeq_{\mathbf{R}^{\mathbf{A}_1}} c_3|_{\mathbf{A}_1}$, implying $\mathcal{G}_1 \preceq_{\mathbf{G}} \mathcal{G}_3$. Finally, it is easy to see that $\mathcal{G}_1 \preceq_{\mathbf{G}} \mathcal{G}_2$ and $\mathcal{G}_2 \preceq_{\mathbf{G}} \mathcal{G}_1$ implies $\mathcal{G}_1 = \mathcal{G}_2$. □

Proof of Lemma 11.5. Consider $q_1, q_2, q_3 \in \mathbf{Q}$. Clearly $q_1 \preceq_{\mathbf{Q}} q_1$. Let $q_1 \preceq_{\mathbf{Q}} q_2$ and $q_2 \preceq_{\mathbf{Q}} q_3$, and let $\langle o^1, d^1, \alpha^1 \rangle \in q_1$. Since $q_1 \preceq_{\mathbf{Q}} q_2$, there is $\langle o^2, d^2, \alpha^2 \rangle \in q_2$ such that $o^1 = o^2$, $d^1 = d^2$, and $\alpha^2 \geq \alpha^1$. Since $q_2 \preceq_{\mathbf{Q}} q_3$, there is $\langle o^3, d^3, \alpha^3 \rangle \in q_3$ such that $o^2 = o^3$, $d^2 = d^3$, and $\alpha^3 \geq \alpha^2$. So, $o^1 = o^3$, $d^1 = d^3$, $\alpha^3 \geq \alpha^1$, proving that $q_1 \preceq_{\mathbf{Q}} q_3$. Finally, $q_1 \preceq_{\mathbf{Q}} q_2$ and $q_2 \preceq_{\mathbf{Q}} q_1$ implies $q_1 = q_2$ (given that origin-destination pairs are not repeated). □

Proof of Lemma 11.8. We need to prove that for $\upsilon_1, \upsilon_2 \in \mathbb{R}_{\geq 0}$ one has: $\upsilon_1 \leq \upsilon_2 \implies \mathrm{red}_{\mathrm{R,V}}(\upsilon_1) \preceq_{\mathbf{G}} \mathrm{red}_{\mathrm{R,V}}(\upsilon_2)$. Following the definition, $\mathrm{red}_{\mathrm{R,V}}(\upsilon_1)$ and $\mathrm{red}_{\mathrm{R,V}}(\upsilon_2)$ will share the same set of vertices (satisfying the vertex condition). Furthermore, $\upsilon_1 \leq \upsilon_2$ implies that the arcs $\mathbf{A}_1$ of $\mathrm{red}_{\mathrm{R,V}}(\upsilon_1)$ will be a subset of the set of arcs $\mathbf{A}_2$ of $\mathrm{red}_{\mathrm{R,V}}(\upsilon_2)$. Finally, the edge colors remain unchanged, except for speed-related one. Let c_1, c_2 the colors associated to $\mathrm{red}_{\mathrm{R,V}}(\upsilon_1)$ and $\mathrm{red}_{\mathrm{R,V}}(\upsilon_2)$, respectively. Clearly $\min\{\upsilon_1, x\} \leq \min\{\upsilon_2, x\}$ for any $x \in \mathbb{R}_{\geq 0}$. This, together with (66), gives $c_1 \succeq_{\mathbf{S}^{\mathbf{A}_1}} c_2$, proving monotonicity. □

Proof of Lemma 11.11. We need to prove that given $\upsilon_1, \upsilon_2 \in \mathbb{R}_{\geq 0}$, one has: $\upsilon_1 \leq \upsilon_2 \implies \mathrm{red}_{\mathrm{R,M}}(\upsilon_1) \preceq_{\mathbf{G}} \mathrm{red}_{\mathrm{R,M}}(\upsilon_2)$. First, notice that sets of vertices and arcs are preserved by $\mathrm{red}_{\mathrm{R,M}}$. Second, the argument for the edge attributes is analogous to the one in the proof of Lemma 11.8. The two facts together prove monotonicity. □

Proof of Lemma 11.13. We need to prove that given $n_1, n_2 \in \mathbb{R}_{\geq 0}$, one has: $n_1 \leq n_2 \implies \mathrm{red}_{\mathrm{P}}(n_1) \preceq_{\mathbf{G}} \mathrm{red}_{\mathrm{P}}(n_2)$. Again, we notice that the set of vertices and arcs are preserved by $\mathrm{red}_{\mathrm{P}}$. Furthermore, $n_1 \leq n_2$

implies that $t_{\mathrm{WS}} + \frac{n_{\mathrm{S,base}}}{2n_1\varphi_{j,\mathrm{base}}} \geq t_{\mathrm{WS}} + \frac{n_{\mathrm{S,base}}}{2n_2\varphi_{j,\mathrm{base}}}$, proving monotonicity. □

Proof of Lemma 11.15. Let $r \preceq_{\mathbf{G}^2\times\bar{\mathbb{N}}\times\overline{\mathbb{R}}_{\geq 0}^3} r'$. Since all feasible solutions of (70) with r remain feasible with r', $\bar{d}_{\mathrm{IAMOD}}(q^*, r) \subseteq \bar{d}_{\mathrm{IAMOD}}(q^*, r')$ for all $q \in \mathbf{Q}$. Similarly, let $q' \preceq_{\mathbf{Q}} q$. Since all feasible solutions of (70) remain feasible (possibly by replacing demand with empty vehicles and by artificially adding loops to the graph), $\bar{d}_{\mathrm{IAMOD}}(q', r) \subseteq \bar{d}_{\mathrm{IAMOD}}(q, r)$ for all $r \in \mathbf{G}^2 \times \bar{\mathbb{N}} \times \overline{\mathbb{R}}_{\geq 0}^3$. This proves monotonicity. □

List of Figures

List of Tables

Bibliography

[1] G. Zardini, D. I. Spivak, A. Censi, and E. Frazzoli, "A compositional sheaf-theoretic framework for event-based systems," in *Proceedings of the 3rd Annual International Applied Category Theory Conference 2020, Cambridge, USA, 6-10th July 2020*, ser. Electronic Proceedings in Theoretical Computer Science, vol. 333, Open Publishing Association, 2020, pp. 139–153. DOI: 10.4204/EPTCS.333.10.

[2] E. Chollet, B. Clarke, M. Johnson, M. Songa, V. Wang, and G. Zardini, "Limits and colimits in a category of lenses," in *Proceedings of the Fourth International Conference on Applied Category Theory, Cambridge, United Kingdom, 12-16th July 2021*, K. Kishida, Ed., ser. Electronic Proceedings in Theoretical Computer Science, vol. 372, Open Publishing Association, 2022, pp. 164–177. DOI: 10.4204/EPTCS.372.12.

[3] D. o. E. a. S. A. United Nations, "68% of the world population projected to live in urban areas by 2050, says un," UN, Tech. Rep., 2021. [Online]. Available: https://www.un.org/development.

[4] M. Czepkiewicz, J. Heinonen, and J. Ottelin, "Why do urbanites travel more than do others? A review of associations between urban form and long-distance leisure travel," *Environmental Research Letters*, vol. 13, no. 7, p. 073 001, 2018. DOI: 10.1088/1748-9326/aac9d2.

[5] C. Calastri, S. Borghesi, and G. Fagiolo, "How do people choose their commuting mode? An evolutionary approach to travel choices," *Economia politica*, vol. 36, no. 3, pp. 887–912, 2019. DOI: 10.1007/s40888-018-0099-1.

[6] S. Ranchordás, "Smart mobility, transport poverty and the legal framework of inclusive mobility," in *Smart Urban Mobility: Law, Regulation, and Policy*. Springer Berlin Heidelberg, 2020, pp. 61–80. DOI: 10.1007/978-3-662-61920-9_4.

[7] T. city of New York, "Onenyc 2050 report," NYC, Tech. Rep., 2021. [Online]. Available: http://onenyc.cityofnewyork.us/strategies.

[8] T. Berger, C. Chen, and C. B. Frey, "Drivers of disruption? Estimating the uber effect," *European Economic Review*, vol. 110, pp. 197–210, 2018. DOI: 10.1016/j.euroecorev.2018.05.006.

[9] B. Rogers, "The social costs of Uber," *U. Chi. L. Rev. Dialogue*, vol. 82, p. 85, 2015.

[10] T. Yigitcanlar, M. Wilson, and M. Kamruzzaman, "Disruptive impacts of automated driving systems on the built environment and land use: An urban planner's perspective," *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 5, no. 2, p. 24, 2019. DOI: 10.3390/joitmc5020024.

[11] U. Nations, "Cities and pollution," UN, Tech. Rep., 2021. [Online]. Available: https://www.un.org/en/climatechange.

[12] E. C. for Mobility and Trasport, "Sustaiable and smart mobility strategy," EU, Tech. Rep., 2021. [Online]. Available: https://ec.europa.eu/transport/sites/transport/files/2021-mobility-strategy-and-action-plan.pdf.

[13] H. A. Simon, *The Sciences of the Artificial (3rd Ed.)* Cambridge, MA, USA: MIT Press, 1996. DOI: 10.7551/mitpress/12107.001.0001.

[14] S. A. Seshia, S. Hu, W. Li, and Q. Zhu, "Design automation of cyber-physical systems: Challenges, advances, and opportunities," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 9, pp. 1421–1434, 2017. DOI: 10.1109/TCAD.2016.2633961.

[15] E. A. Lee, "Cyber physical systems: Design challenges," in *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, 2008, pp. 363–369. DOI: 10.1109/ISORC.2008.25.

[16] A. Q. Nilles, D. A. Shell, and J. M. O'Kane, "Robot Design: Formalisms, Representations, and the Role of the Designer," 2018. [Online]. Available: http://arxiv.org/abs/1806.05157.

[17] A. MacCormack, C. Baldwin, and J. Rusnak, "Exploring the duality between product and organizational architectures: A test of the "mirroring" hypothesis," *Research Policy*, vol. 41, no. 8, pp. 1309–1324, Oct. 2012. DOI: 10.1016/j.respol.2012.04.011. [Online]. Available: http://dx.doi.org/10.1016/j.respol.2012.04.011.

[18] O. L. de Weck, D. Roos, and C. L. Magee, *Engineering Systems: Meeting Human Needs in a Complex Technological World*. The MIT Press, Oct. 2011. DOI: 10.7551/mitpress/8799.001.0001. [Online]. Available: https://doi.org/10.7551/mitpress/8799.001.0001.

[19] E. Antonsson and J. Cagan, *Formal Engineering Design Synthesis*. 2001. DOI: 10.1017/cbo9780511529627.

[20] K. K. Fu, M. C. Yang, and K. L. Wood, "Design principles: Literature review, analysis, and future directions," *Journal of Mechanical Design, Transactions of the ASME*, vol. 138, no. 10, pp. 1–13, 2016. DOI: 10.1115/1.4034105.

[21] O. Sigmund and K. Maute, "Topology optimization approaches: A comparative review," *Structural and multidisciplinary optimization*, vol. 48, no. 6, pp. 1031–1055, 2013. DOI: 10.1007/s00158-013-0978-6.

[22] J. L. Jewett and J. V. Carstensen, "Topology-optimized design, construction and experimental evaluation of concrete beams," *Automation in Construction*, vol. 102, pp. 59–67, 2019. DOI: 10.1016/j.autcon.2019.02.001.

[23] M. I. Campbell, J. Cagan, and K. Kotovsky, "A-Design: An agent-based approach to conceptual design in a dynamic environment," *Research in Engineering Design - Theory, Applications, and Concurrent Engineering*, vol. 11, no. 3, pp. 172–192, 1999. DOI: 10.1007/s001630050013.

[24] M. I. Campbell, J. Cagan, and K. Kotovsky, "The A-design approach to managing automated design synthesis," *Research in Engineering Design*, vol. 14, no. 1, pp. 12–24, 2003. DOI: 10.1007/s00163-002-0025-x.

[25] H. Yoshikawa, "General design theory and a cad system," *Man-machine Communication in CAD/CAM*, vol. 35, 1981.

[26] Y. Reich, "A Critical Review of General Design Theory," vol. 7, pp. 1–18, 1995. DOI: 10.1007/BF01681909.

[27] V. Hubka and W. E. Eder, *Theory of technical systems: a total concept theory for engineering design*. Springer Science & Business Media, 2012. DOI: 10.1007/978-3-642-52121-8.

[28] N. P. Suh, "Axiomatic Design Theory for Systems," *Research in Engineering Design*, vol. 10, pp. 189–209, 1998. DOI: 10.1007/s001639870001.

[29] N. P. Suh, *Axiomatic design: advances and applications*. Oxford university press New York, 2009. DOI: 10.1016/S0142-694X(02)00058-3.

[30] J. S. Arora, "Multi-objective optimum design concepts and methods," *Introduction to optimum design*, pp. 657–679, 2012. DOI: 10.1016/B978-012064155-0/50017-3.

[31] I. Y. Kim and O. L. de Weck, "Adaptive weighted sum method for multiobjective optimization: A new method for pareto front generation," *Structural and multidisciplinary optimization*, vol. 31, no. 2, pp. 105–116, 2006. DOI: 10.1007/s00158-005-0557-6.

[32] O. L. de Weck, "Multiobjective optimization: History and promise," 2004.

[33] A. Prorok, M. Malencia, L. Carlone, G. S. Sukhatme, B. M. Sadler, and V. Kumar, "Beyond robustness: A taxonomy of approaches towards resilient multi-robot systems," *arXiv preprint arXiv:2109.12343*, 2021.

[34] H. Kress-Gazit, M. Lahijanian, and V. Raman, "Synthesis for Robots: Guarantees and Feedback for Robot Behavior," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 211–236, 2018. DOI: 10.1146/annurev-control-060117-104838.

[35] G. S. Hornby, H. Lipson, and J. B. Pollack, "Generative representations for the automated design of modular physical robots," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 4, pp. 703–719, 2003. DOI: 10.1109/TRA.2003.814502.

[36] H. Lipson and J. B. Pollack, "Automatic design and manufacture of robotic lifeforms," *Letters to Nature*, vol. 406, no. August, 2000. DOI: 10.1038/35023115.

[37] I. Incer, "The Algebra of Contracts," Ph.D. dissertation, 2022.

[38] E. Kolberg, Y. Reich, and I. Levin, "Designing winning robots by careful design of their development process," *Research in Engineering Design*, vol. 25, no. 2, pp. 157–183, 2014. DOI: 10.1007/s00163-014-0171-y.

[39] Q. Zhu and A. Sangiovanni-Vincentelli, "Codesign methodologies and tools for cyber–physical systems," *Proceedings of the IEEE*, vol. 106, no. 9, pp. 1484–1500, 2018. DOI: 10.1109/JPROC.2018.2864271.

[40] J.-P. Merlet, "Optimal design of robots," in *Robotics: Science and systems*, 2005. DOI: 10.15607/RSS.2005.I.041.

[41] M. Lahijanian, M. Svorenova, A. A. Morye, *et al.*, *Resource-performance tradeoff analysis for mobile robots*, 2018. DOI: 10.1109/LRA.2018.2803814.

[42] S. Seok, A. Wang, M. Y. Chuah, *et al.*, "Design principles for energy-efficient legged locomotion and implementation on the mit cheetah robot," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 3, pp. 1117–1129, 2014. DOI: 10.1109/TMECH.2014.2339013.

[43] J. M. O'Kane and S. M. LaValle, "Comparing the power of robots," *The International Journal of Robotics Research*, vol. 27, no. 1, pp. 5–23, Jan. 2008. DOI: 10.1177/02783649070820.

[44] S. Karaman and E. Frazzoli, "High-speed motion with limited sensing range in a poisson forest," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, 2012, pp. 3735–3740. DOI: 10.1109/CDC.2012.6426047.

[45] G. Bravo-Palacios, A. Del Prete, and P. M. Wensing, "One robot for many tasks: Versatile co-design through stochastic programming," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1680–1687, 2020. DOI: 10.1109/LRA.2020.2969948.

[46] F. Ramos, A. S. Vázquez, R. Fernández, and A. Olivares-Alarcos, "Ontology based design, control and programming of modular robots," *Integrated Computer-Aided Engineering*, vol. 25, no. 2, pp. 173–192, 2018. DOI: 10.3233/ICA-180569.

[47] A. A. C. Collin, "A systems architecture framework towards hardware selection for autonomous navigation," Ph.D. dissertation, Massachusetts Institute of Technology, 2019.

[48] A. Censi, E. Mueller, E. Frazzoli, and S. Soatto, "A Power-Performance Approach to Comparing Sensor Families, with application to comparing neuromorphic to traditional vision sensors," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015-June, May 2015, pp. 3319–3326. DOI: 10.1109/ICRA.2015.7139657.

[49] J. Ichnowski, J. Prins, and R. Alterovitz, "Cloud-based motion plan computation for power-constrained robots," in *Algorithmic Foundations of Robotics XII*, Springer, 2020, pp. 96–111. DOI: 10.1007/978-3-030-43089-4_7.

[50] A. Schulz, J. Xu, B. Zhu, C. Zheng, E. Grinspun, and W. Matusik, "Interactive design space exploration and optimization for CAD models," *ACM Transactions on Graphics*, vol. 36, no. 4, 2017. DOI: 10.1145/3072959.3073688.

[51] A. Schulz, H. Wang, E. Grinspun, J. Solomon, and W. Matusik, "Interactive exploration of design trade-offs," *ACM Transactions on Graphics*, vol. 37, no. 4, 2018. DOI: 10.1145/3197517.3201385.

[52] A. Schulz, "Computational Design for the Next Manufacturing Revolution," Ph.D. dissertation, 2018.

[53] G. Jing, T. Tosun, M. Yim, and H. Kress-Gazit, "Accomplishing high-level tasks with modular robots," *Autonomous Robots*, vol. 42, pp. 1337–1354, 2018. DOI: 10.1007/s10514-018-9738-1.

[54] T. Tosun, G. Jing, H. Kress-Gazit, and M. Yim, "Computer-aided compositional design and verification for modular robots," *Springer Proceedings in Advanced Robotics*, vol. 2, pp. 237–252, 2018. DOI: 10.1007/978-3-319-51532-8-15.

[55] B. R. Donald, "On information invariants in robotics," *Artificial Intelligence*, vol. 72, no. 1-2, pp. 217–304, 1995. DOI: 10.1016/0004-3702(94)00024-U.

[56] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 451–462, 2008. DOI: 10.1109/TSP.2008.2007095.

[57] V. Gupta, T. H. Chung, B. Hassibi, and R. M. Murray, "On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage," *Automatica*, vol. 42, no. 2, pp. 251–260, 2006. DOI: 10.1016/j.automatica.2005.09.016.

[58] C. Giraud and B. Jouvencel, "Sensor selection: A geometrical approach," in *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, vol. 2, 1995, 555–560 vol.2. DOI: 10.1109/IROS.1995.526271.

[59] M. Shamaiah, S. Banerjee, and H. Vikalo, "Greedy sensor selection: Leveraging submodularity," in *49th IEEE Conference on Decision and Control (CDC)*, 2010, pp. 2572–2577. DOI: 10.1109/CDC.2010.5717225.

[60] V. Tzoumas, L. Carlone, G. J. Pappas, and A. Jadbabaie, "Lqg control and sensing co-design," *IEEE Transactions on Automatic Control*, vol. 66, no. 4, pp. 1468–1483, 2021. DOI: 10.1109/TAC.2020.2997661.

[61] A. Collin, A. Siddiqi, Y. Imanishi, Y. Matta, T. Tanimichi, and O. de Weck, "A multiobjective systems architecture model for sensor selection in autonomous vehicle navigation," in *International Conference on Complex Systems Design & Management*, Springer, 2020, pp. 141–152. DOI: 10.1007/978-3-030-34843-4_12.

[62] A. Collin, A. Siddiqi, Y. Imanishi, E. Rebentisch, T. Tanimichi, and O. L. de Weck, "Autonomous driving systems hardware and software architecture exploration: Optimizing latency and cost under safety constraints," *Systems Engineering*, vol. 23, no. 3, pp. 327–337, 2020. DOI: 10.1002/sys.21528.

[63] B. Guo, O. Karaca, T. Summers, and M. Kamgarpour, "Actuator placement for optimizing network performance under controllability constraints," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, IEEE, 2019, pp. 7140–7147. DOI: 10.1109/CDC40024.2019.9030204.

[64] B. Guo, O. Karaca, T. Summers, and M. Kamgarpour, "Actuator placement under structural controllability using forward and reverse greedy algorithms," *IEEE Transactions on Automatic Control*, vol. 66, no. 12, pp. 5845–5860, 2020. DOI: 10.1109/TAC.2020.3044284.

[65] D. Golovin, M. Faulkner, and A. Krause, "Online distributed sensor selection," in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2010, pp. 220–231. DOI: 10.1145/1791212.1791239.

[66] T. Tanaka and H. Sandberg, "Sdp-based joint sensor and controller design for information-regularized optimal lqg control," in *2015 54th IEEE Conference on Decision and Control (CDC)*, IEEE, 2015, pp. 4486–4491. DOI: 10.1109/CDC.2015.7402920.

[67] S. Tatikonda and S. Mitter, "Control under communication constraints," *IEEE Transactions on automatic control*, vol. 49, no. 7, pp. 1056–1068, 2004. DOI: 10.1109/TAC.2004.831187.

[68] D. Soudbakhsh, L. T. Phan, O. Sokolsky, I. Lee, and A. Annaswamy, "Co-design of control and platform with dropped signals," in *Proceedings of the ACM/IEEE 4th international conference on cyber-physical systems*, 2013, pp. 129–140. DOI: 10.1145/2502524.2502542.

[69] U. Rosolia, A. Singletary, and A. D. Ames, "Unified multirate control: From low-level actuation to high-level planning," *IEEE Transactions on Automatic Control*, vol. 67, no. 12, pp. 6627–6640, 2022. DOI: 10.1109/TAC.2022.3184664.

[70] D. A. Shell, J. M. O'Kane, and F. Z. Saberifar, "On the design of minimal robots that can solve planning problems," *IEEE Transactions on Automation Science and Engineering*, pp. 1–12, 2021. DOI: 10.1109/TASE.2021.3050033.

[71] S. Ghasemlou, J. M. Okane, and D. A. Shell, "Delineating boundaries of feasibility between robot designs," *IEEE International Conference on Intelligent Robots and Systems*, pp. 422–429, 2018. DOI: 10.1109/IROS.2018.8593811.

[72] S. Ghasemlou and J. M. O'Kane, "Accelerating the Construction of Boundaries of Feasibility in Three Classes of Robot Design Problems," *IEEE International Conference on Intelligent Robots and Systems*, pp. 2532–2538, 2019. DOI: 10.1109/IROS40897.2019.8968258.

[73] S. Ghasemlou, "Algorithmic Robot Design: Label Maps, Procrustean Graphs, and the Boundary of Non-destructiveness," Ph.D. dissertation, 2020, p. 14. [Online]. Available: https://all3dp.com/2/fused-deposition-modeling-fdm-3d-printing-simply-explained/.

[74] F. Z. Saberifar, J. M. O'Kane, and D. A. Shell, "The Hardness of Minimizing Design Cost Subject to Planning Problems," *Springer Proceedings in Advanced Robotics*, vol. 14, pp. 868–883, 2020. DOI: 10.1007/978-3-030-44051-0-50.

[75] M. Erdmann, "Understanding Action and Sensing by Designing Action-Based Sensors," *International Journal of Robotics Research*, pp. 483–509, 1995. DOI: 10.1177/027836499501400506.

[76] F. Z. Saberifar, S. Ghasemlou, D. A. Shell, and J. M. O'Kane, "Toward a language-theoretic foundation for planning and filtering," *International Journal of Robotics Research*, vol. 38, no. 2-3, pp. 236–259, 2019. DOI: 10.1177/0278364918801503.

[77] M. Maasoumy, Q. Zhu, C. Li, F. Meggers, and A. Sangiovanni-Vincentelli, "Co-design of control algorithm and embedded platform for building hvac systems," in *Proceedings of the ACM/IEEE 4th International Conference on Cyber-Physical Systems*, 2013, pp. 61–70. DOI: 10.1145/2502524.2502533.

[78] B. Zheng, P. Deng, R. Anguluri, Q. Zhu, and F. Pasqualetti, "Cross-layer codesign for secure cyber-physical systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 5, pp. 699–711, 2016. DOI: 10.1109/TCAD.2016.2523937.

[79] A. Spielberg, A. Amini, L. Chin, W. Matusik, and D. Rus, "Co-learning of task and sensor placement for soft robotics," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1208–1215, 2021. DOI: 10.1109/LRA.2021.3056369.

[80] L. Ballotta, L. Schenato, and L. Carlone, "Computation-communication trade-offs and sensor selection in real-time estimation for processing networks," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2952–2965, 2020. DOI: 10.1109/TNSE.2020.3008337.

[81] Y. Hu, J. Liu, A. Spielberg, *et al.*, "ChainQueen: A real-time differentiable physical simulator for soft robotics," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2019-May, pp. 6265–6271, 2019. DOI: 10.1109/ICRA.2019.8794333.

[82] P. Ma, T. Du, J. Z. Zhang, *et al.*, "DiffAqua: A Differentiable Computational Design Pipeline for Soft Underwater Swimmers with Shape Interpolation," *ACM Transactions on Graphics*, vol. 40, no. 4, 2021. DOI: 10.1145/3450626.3459832.

[83] M. Dubied, M. Y. Michelis, A. Spielberg, and R. K. Katzschmann, "Sim-to-Real for Soft Robots Using Differentiable FEM: Recipes for Meshing, Damping, and Actuation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5015–5022, 2022. DOI: 10.1109/LRA.2022.3154050.

[84] P. Karkus, B. Ivanovic, S. Mannor, and M. Pavone, "Diffstack: A differentiable and modular control stack for autonomous vehicles," in *Proceedings of The 6th Conference on Robot Learning*, K. Liu, D. Kulic, and J. Ichnowski, Eds., ser. Proceedings of Machine Learning Research, vol. 205, PMLR, 2023, pp. 2170–2180.

[85] J. Xu, A. Spielberg, A. Zhao, D. Rus, and W. Matusik, "Multi-Objective Graph Heuristic Search for Terrestrial Robot Design," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2021-May, no. Icra, pp. 12 759–12 765, 2021. DOI: 10.1109/ICRA48506.2021.9561818.

[86] A. Zhao, J. Xu, M. Konaković-Luković, *et al.*, "RoboGrammar: Graph grammar for terrain-optimized robot design," *ACM Transactions on Graphics*, vol. 39, no. 6, 2020. DOI: 10.1145/3414685.3417831.

[87] A. Spielberg, B. Araki, C. Sung, R. Tedrake, and D. Rus, "Functional co-optimization of articulated robots," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 5035–5042. DOI: 10.1109/ICRA.2017.7989587.

[88] A. Spielberg, A. Zhao, T. Du, Y. Hu, D. Rus, and W. Matusik, "Learning-in-the-loop optimization: End-to-end control and co-design of soft robots through learned deep latent representations," *Advances in Neural Information Processing Systems*, vol. 32, no. NeurIPS, 2019.

[89] A. Schulz, C. Sung, A. Spielberg, *et al.*, "Interactive robogami: An end-to-end system for design of robots with ground locomotion," *International Journal of Robotics Research*, vol. 36, no. 10, pp. 1131–1147, 2017. DOI: 10.1177/0278364917723465.

[90] C. R. Sung, "Computational design of foldable robots via composition," no. 2011, p. 192, 2016. [Online]. Available: https://dspace.mit.edu/handle/1721.1/113734.

[91] A. M. Mehta and D. Rus, "An end-to-end system for designing mechanical structures for print-and-fold robots," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1460–1465, 2014. DOI: 10.1109/ICRA.2014.6907044.

[92] A. M. Mehta, J. DelPreto, K. W. Wong, S. Hamill, H. Kress-Gazit, and D. Rus, "Robot Creation from Functional Specifications," *Springer Proceedings in Advanced Robotics*, vol. 3, pp. 631–648, 2018. DOI: 10.1007/978-3-319-60916-4-36.

[93] A. M. Mehta, J. Delpreto, B. Shaya, and D. Rus, "Cogeneration of mechanical, electrical, and software designs for printable robots from structural specifications," *IEEE International Conference on Intelligent Robots and Systems*, no. Iros, pp. 2892–2897, 2014. DOI: 10.1109/IROS.2014.6942960.

[94] T. Wang, Y. Zhou, S. Fidler, and J. Ba, "Neural graph evolution: Towards efficient automatic robot design," *7th International Conference on Learning Representations, ICLR 2019*, pp. 1–17, 2019.

[95] M. Römmerman, D. Kühn, and F. Kirchner, "Robot design for space missions using evolutionary computation," *2009 IEEE Congress on Evolutionary Computation, CEC 2009*, pp. 2098–2105, 2009. DOI: 10.1109/CEC.2009.4983200.

[96] I. Tanev, T. Ray, and A. Buller, "Automated evolutionary design, robustness, and adaptation of sidewinding locomotion of a simulated snake-like robot," *IEEE Transactions on Robotics*, vol. 21, no. 4, pp. 632–645, 2005. DOI: 10.1109/TRO.2005.851028.

[97] C. Leger and J. Bares, "Automated synthesis and optimization of robot configurations," *Proceedings of the ASME Design Engineering Technical Conference*, vol. 1A-1998, 1998. DOI: 10.1115/DETC98/MECH-5945.

[98] S. Ha, S. Coros, A. Alspach, *et al.*, "Computational Design of Robotic Devices From High-Level Motion Specifications," *IEEE Transactions on Robotics*, vol. 34, no. 5, pp. 1240–1251, 2018. DOI: 10.1109/TRO.2018.2830419.

[99] B. Thomaszewski, S. Coros, D. Gauge, V. Megaro, E. Grinspun, and M. Gross, "Computational design of linkage-based characters," *ACM Transactions on Graphics*, vol. 33, no. 4, pp. 1–9, 2014. DOI: 10.1145/2601097.2601143.

[100] M. Geilinger, R. Poranne, R. Desai, B. Thomaszewski, and S. Coros, "Skaterbots: Optimization-based design and motion synthesis for robotic creatures with legs and wheels," vol. 37, no. 4, 2018. DOI: 10.1145/3197517.3201368.

[101] V. Megaro, B. Thomaszewski, M. Nitti, O. Hilliges, M. Gross, and S. Coros, "Interactive design of 3d-printable robotic creatures," *ACM Transactions on Graphics*, vol. 34, no. 6, pp. 1–9, 2015. DOI: 10.1145/2816795.2818137.

[102] S. Ha, S. Coros, A. Alspach, J. Kim, and K. Yamane, "Joint optimization of robot design and motion parameters using the implicit function theorem," *Robotics: Science and Systems*, vol. 13, 2017. DOI: 10.15607/rss.2017.xiii.003.

[103] G. Bharaj, S. Coros, B. Thomaszewski, J. Tompkin, B. Bickel, and H. Pfister, "Computational design of walking automata," *Proceedings - SCA 2015: 14th ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, pp. 93–100, 2015. DOI: 10.1145/2786784.2786803.

[104] J. H. Park and H. Asada, "Concurrent design optimization of mechanical structure and control for high speed robots," *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, vol. 116, no. 3, pp. 344–356, 1994. DOI: 10.1115/1.2899229.

[105] N. Cheney, J. Bongard, V. SunSpiral, and H. Lipson, "Scalable co-optimization of morphology and control in embodied machines," *Journal of the Royal Society Interface*, vol. 15, no. 143, 2018. DOI: 10.1098/rsif.2017.0937.

[106] N. Cheney, J. Bongard, V. Sunspiral, and H. Lipson, "On the Difficulty of Co-Optimizing Morphology and Control in Evolved Virtual Creatures," 2013.

[107] O. Magnussen, M. Ottestad, and G. Hovland, "Multicopter design optimization and validation," *Modeling, Identification and Control*, vol. 36, no. 2, pp. 67–79, 2015. DOI: 10.4173/mic.2015.2.1.

[108] J. Ziglar, R. K. Williams, and A. Wicks, "Context-aware system synthesis, task assignment, and routing," *Autonomous Robots*, vol. 47, no. 2, pp. 193–210, 2023. DOI: 10.1007/s10514-022-10076-3.

[109] A. Benveniste, B. Caillaud, D. Nickovic, *et al.*, "Contracts for system design," *Foundations and Trends® in Electronic Design Automation*, vol. 12, no. 2-3, pp. 124–400, 2018. DOI: 10.1561/1000000053.

[110] A. Ferrari and A. Sangiovanni-Vincentelli, "System design: Traditional concepts and new paradigms," in *Proceedings 1999 IEEE International Conference on Computer Design: VLSI in Computers and Processors (Cat. No.99CB37040)*, 1999, pp. 2–12. DOI: 10.1109/ICCD.1999.808256.

[111] K. Keutzer, A. Newton, J. Rabaey, and A. Sangiovanni-Vincentelli, "System-level design: Orthogonalization of concerns and platform-based design," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, no. 12, pp. 1523–1543, 2000. DOI: 10.1109/43.898830.

[112] A. Sangiovanni-Vincentelli, L. Carloni, F. De Bernardinis, and M. Sgroi, "Benefits and challenges for platform-based design," in *Proceedings of the 41st Annual Design Automation Conference*, New York, NY, USA: Association for Computing Machinery, 2004, pp. 409–414. DOI: 10.1145/996566.996684.

[113] A. Sangiovanni-Vincentelli, "Quo vadis, sld? reasoning about the trends and challenges of system level design," *Proceedings of the IEEE*, vol. 95, no. 3, pp. 467–506, 2007. DOI: 10.1109/JPROC.2006.890107.

[114] P. Nuzzo, A. L. Sangiovanni-Vincentelli, D. Bresolin, L. Geretti, and T. Villa, "A platform-based design methodology with contracts and related tools for the design of cyber-physical systems," *Proceedings of the IEEE*, vol. 103, no. 11, pp. 2104–2132, 2015. DOI: 10.1109/JPROC.2015.2453253.

[115] P. Nuzzo, J. Li, A. L. Sangiovanni-Vincentelli, Y. Xi, and D. Li, "Stochastic assume-guarantee contracts for cyber-physical system design," *ACM Transactions on Embedded Computing Systems*, vol. 18, no. 1, 2019. DOI: 10.1145/3243216.

[116] A. Sangiovanni-Vincentelli, W. Damm, and R. Passerone, "Taming dr. frankenstein: Contract-based design for cyber-physical systems*," *European Journal of Control*, vol. 18, no. 3, pp. 217–238, 2012. DOI: 10.3166/ejc.18.217-238.

[117] I. Incer, A. Badithela, J. Graebener, *et al.*, "Pacti: Scaling Assume-Guarantee Reasoning for System Analysis and Design," pp. 1–22, 2023. [Online]. Available: http://arxiv.org/abs/2303.17751.

[118] S. Liu, A. Saoud, P. Jagtap, D. V. Dimarogonas, and M. Zamani, "Compositional Synthesis of Signal Temporal Logic Tasks via Assume-Guarantee Contracts," *Proceedings of the IEEE Conference on Decision and Control*, vol. 2022-Decem, no. Cdc, pp. 2184–2189, 2022. DOI: 10.1109/CDC51059.2022.9992715.

[119] K. Ghasemi, S. Sadraddini, and C. Belta, "Compositional synthesis via a convex parameterization of assume-guarantee contracts," *HSCC 2020 - Proceedings of the 23rd International Conference on Hybrid Systems: Computation and Control ,part of CPS-IoT Week*, no. 1, 2020. DOI: 10.1145/3365365.3382212.

[120] A. Saoud, P. Jagtap, M. Zamani, and A. Girard, "Compositional abstraction-based synthesis for interconnected systems: An approximate composition approach," *IEEE Transactions on Control of Network Systems*, vol. 8, no. 2, pp. 702–712, 2021. DOI: 10.1109/TCNS.2021.3050123.

[121] M. Sharf, B. Besselink, A. Molin, Q. Zhao, and K. H. Johansson, "Assume/Guarantee Contracts for Dynamical Systems: Theory and Computational Tools," *IFAC-PapersOnLine*, vol. 54, no. 5, pp. 25–30, 2021. DOI: 10.1016/j.ifacol.2021.08.469. [Online]. Available: https://doi.org/10.1016/j.ifacol.2021.08.469.

[122] A. Censi, "A mathematical theory of co-design," *arXiv preprint arXiv:1512.08055v7*, 2015.

[123] A. Censi, J. Lorand, and G. Zardini, *Applied Category Theory for Engineering*. 2023, Work in progress book. [Online]. Available: https://bit.ly/3H6pwMo.

[124] E. Rechtin and M. W. Maier, *The Art of Systems Architecting*. USA: CRC Press, Inc., 1997. DOI: 10.1201/9781420079142.

[125] B. Davey and H. Priestley, *Introduction to Lattices and Order*, 2nd ed. Cambridge University Press, 2002. DOI: 10.1017/CBO9780511809088.

[126] S. Roman, *Lattices and Ordered Sets*. Springer, 2008. DOI: 10.1007/978-0-387-78901-9.

[127] S. M. LaValle *et al.*, "Sensing and filtering: A fresh perspective based on preimages and information spaces," *Foundations and Trends® in Robotics*, vol. 1, no. 4, pp. 253–372, 2012.

[128] G. Zardini, D. Milojevic, A. Censi, and E. Frazzoli, "Co-design of embodied intelligence: A structured approach," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 7536–7543. DOI: 10.1109/IROS51168.2021.9636513.

[129] P. Duhr, D. Buccheri, C. Balerna, A. Cerofolini, and C. H. Onder, "Minimum-race-time energy allocation strategies for the hybrid-electric formula 1 power unit," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 6, pp. 7035–7050, 2023. DOI: 10.1109/TVT.2023.3237388.

[130] A Lodi, S Martello, and M Monaci, "Two-dimensional packing problems: A survey," *European Journal of Operational Research*, vol. 141, no. 2, pp. 241–252, 2002. DOI: https://doi.org/10.1016/S0377-2217(02)00123-6.

[131] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao blackwellized particle filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, Feb. 2007. DOI: 10.1109/TRO.2006.889486.

[132] A. Locher, M. Perdoch, and L. Van Gool, "Progressive prioritized multi-view stereo," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[133] M. Z. Zia, L. Nardi, A. Jack, *et al.*, "Comparative design space exploration of dense and semi-dense slam," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1292–1299. DOI: 10.1109/ICRA.2016.7487261.

[134] G. Zardini, A. Censi, and E. Frazzoli, "Co-design of autonomous systems: From hardware selection to control synthesis," in *2021 European Control Conference (ECC)*, 2021, pp. 682–689. DOI: 10.23919/ECC54610.2021.9654960.

[135] M. H. Davis, "Linear estimation and stochastic control," 1977.

[136] H. Kwakernaak and R. Sivan, *Linear optimal control systems*. Wiley-interscience New York, 1972, vol. 1.

[137] P. De Leenheer and E. D. Sontag, "A note on the monotonicity of matrix riccati equations," Tech. Rep., 2004.

[138] E. Hendricks, O. Jannerup, and P. H. Sørensen, *Linear systems control: deterministic and stochastic methods*. Springer.

[139] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. Jordan, and S. Sastry, "Kalman filtering with intermittent observations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, 2004. DOI: 10.1109/TAC.2004.834121.

[140] A. Censi, "Kalman filtering with intermittent observations: Convergence for semi-markov chains and an intrinsic performance measure," *IEEE Transactions on Automatic Control*, 2011. DOI: 10.1109/TAC.2010.2097350.

[141] S. M. Melzer and B. C. Kuo, "Sampling period sensitivity of the optimal sampled data linear regulator," *Automatica*, vol. 7, no. 3, pp. 367–370, 1971. DOI: 10.1016/0005-1098(71)90129-4.

[142] E. Bini and G. M. Buttazzo, "The optimal sampling pattern for linear control systems," *IEEE Transactions on Automatic Control*, vol. 59, no. 1, pp. 78–90, 2013. DOI: 10.1109/TAC.2013.2279913.

[143] A. Benveniste, D. Nickovic, B. Caillaud, *et al.*, *Contracts for system design*. 2018, vol. 12, pp. 124–400. DOI: 10.1561/1000000053.

[144] E. S. Kim, M. Arcak, and S. A. Seshia, "A small gain theorem for parametric assume-guarantee contracts," *HSCC 2017 - Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control (part of CPS Week)*, pp. 207–216, 2017. DOI: 10.1145/3049797.3049805.

[145] L. Sandel, G. Zardini, S. Mitrova, *et al.*, "Enhancing efficiency and reliability of electric vehicles via adaptive e-gear control," in *2023 IEEE International Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2023.

[146] A. Rezaeizadeh, G. Zardini, E. Frazzoli, and S. Mastellone, "Reliability-aware control of power converters in mobility applications," *2024 European Control Conference*, 2024.

[147] L. Nardi, B. Bodin, M. Z. Zia, *et al.*, "Introducing slambench, a performance and accuracy benchmarking methodology for slam," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 5783–5790. DOI: 10.1109/ICRA.2015.7140009.

[148] B. Fong and D. I. Spivak, *An invitation to applied category theory: seven sketches in compositionality*. Cambridge University Press, 2019.

[149] E. Manes and M. Arbib, *Algebraic approaches to program semantics*. Springer-Verlag, 1986. DOI: 10.1007/978-1-4612-4962-7.

[150] E. A. Lee and S. A. Seshia, *Introduction to Embedded Systems - A Cyber-Physical Systems Approach*, 1st ed. 2010, Available for download on the authors' website http://leeseshia.org/.

[151] G. Gierz, K. H. Hofmann, K. Keimel, J. D. Lawson, M. Mislove, and D. S. Scott, *Continuous Lattices and Domains*. Cambridge University Press, 2003. DOI: 10.1017/cbo9780511542725.

[152] A. Censi, "Uncertainty in monotone codesign problems," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1556–1563, 2017. DOI: 10.1109/LRA.2017.2674970.

[153] NVIDIA. "Nvidia products." available online: https://www.nvidia.com. (2020).

[154] Basler. "Basler cameras." available online: https://www.baslerweb.com. (2020).

[155] Flir. "Flir cameras." available online: https://www.flir.com. (2020).

[156] W. B. Arthur, "The structure of invention," *Research policy*, vol. 36, no. 2, pp. 274–287, 2007.

[157] C. L. Magee and O. L. de Weck, "3.1. 3 complex system classification," in *INCOSE International Symposium*, Wiley Online Library, vol. 14, 2004, pp. 471–488.

[158] R. J. Van Wyk, "Management of technology: New frameworks," *Technovation*, vol. 7, no. 4, pp. 341–351, 1988.

[159] M. Kloetzer and C. Belta, "A fully automated framework for control of linear systems from temporal logic specifications," *IEEE Transactions on Automatic Control*, vol. 53, no. 1, pp. 287–297, 2008. DOI: 10.1109/TAC.2007.914952.

[160] G. Fainekos, H. Kress-Gazit, and G. Pappas, "Temporal logic motion planning for mobile robots," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, pp. 2020–2025. DOI: 10.1109/ROBOT.2005.1570410.

[161] A. Bhatia, L. E. Kavraki, and M. Y. Vardi, "Sampling-based motion planning with temporal goals," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 2689–2696. DOI: 10.1109/ROBOT.2010.5509503.

[162] E. M. Wolff, U. Topcu, and R. M. Murray, "Optimization-based trajectory generation with linear temporal logic specifications," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 5319–5325. DOI: 10.1109/ICRA.2014.6907641.

[163] G. Pahl, W. Beitz, J. Feldhusen, and K.-H. Grote, "Engineering design: A systematic approach," *MRS Bulletin*, vol. 71, 1996. DOI: 10.1557/S0883769400035776.

[164] P. Shankar, B. Morkos, D. Yadav, and J. D. Summers, "Towards the formalization of non-functional requirements in conceptual design," *Research in Engineering Design*, vol. 31, no. 4, pp. 449–469, Oct. 2020. DOI: 10.1007/s00163-020-00345-6.

[165] Z. Yu, G. Zardini, A. Censi, and S. Fuller, "Visual confined-space navigation using an efficient learned bilinear optic flow approximation for insect-scale robots," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 4250–4256. DOI: 10.1109/IROS47612.2022.9981585.

[166] Y. M. Chukewad, J. James, A. Singh, and S. Fuller, "Robofly: An insect-sized robot with simplified fabrication that is capable of flight, ground, and water surface locomotion," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 2025–2040, 2021. DOI: 10.1109/TRO.2021.3075374.

[167] A. Reschka, J. R. Böhmer, F. Saust, B. Lichte, and M. Maurer, "Safe, dynamic and comfortable longitudinal control for an autonomous vehicle," in *2012 IEEE Intelligent Vehicles Symposium*, IEEE, 2012, pp. 346–351. DOI: 10.1109/IVS.2012.6232159.

[168] A. A. Association *et al.*, *Aaa's your driving costs*, 2018.

[169] Velodyne. "Velodyne lidars." available online: https://velodynelidar.com. (2020).

[170] Ouster. "Ouster lidars." available online: https://ouster.com. (2020).

[171] M. Braun, S. Krebs, F. Flohr, and D. M. Gavrila, "Eurocity persons: A novel benchmark for person detection in traffic scenes," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1844–1861, 2019. DOI: 10.1109/TPAMI.2019.2897684.

[172] B Borgmann, M Hebel, M Arens, and U Stilla, "Pedestrian detection and tracking in sparse mls point clouds using a neural network and voting-based approach," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 2, pp. 187–194, 2020. DOI: 10.5194/isprs-annals-V-2-2020-187-2020.

[173] K. Liu, W. Wang, and J. Wang, "Pedestrian detection with lidar point clouds based on single template matching," *Electronics*, vol. 8, no. 7, 2019. DOI: 10.3390/electronics8070780. [Online]. Available: https://www.mdpi.com/2079-9292/8/7/780.

[174] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[175] D. Gallup, J.-M. Frahm, P. Mordohai, and M. Pollefeys, "Variable baseline/resolution stereo," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8. DOI: 10.1109/CVPR.2008.4587671.

[176] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1851–1858.

[177] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, vol. 32, pp. 1231–1237, 11 2013. DOI: 10.1177/0278364913491297.

[178] G. Zardini, Z. Suter, A. Censi, and E. Frazzoli, "Task-driven modular co-design of vehicle control systems," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, 2022, pp. 2196–2203. DOI: 10.1109/CDC51059.2022.9993107.

[179] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, 2016. DOI: 10.1109/TIV.2016.2578706.

[180] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," in *2015 IEEE intelligent vehicles symposium (IV)*, IEEE, 2015, pp. 1094–1099. DOI: 10.1109/IVS.2015.7225830.

[181] M. Rokonuzzaman, N. Mohajer, S. Nahavandi, and S. Mohamed, "Review and performance evaluation of path tracking controllers of autonomous vehicles," *IET Intelligent Transport Systems*, vol. 15, no. 5, pp. 646–670, 2021. DOI: 10.1049/itr2.12051.

[182] R. C. Coulter, "Implementation of the pure pursuit path tracking algorithm," Carnegie-Mellon UNIV Pittsburgh PA Robotics INST, Tech. Rep., 1992.

[183] M. Althoff, M. Koschi, and S. Manzinger, "Commonroad: Composable benchmarks for motion planning on roads," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 719–726. DOI: 10.1109/IVS.2017.7995802.

[184] G. Zardini, N. Lanzetti, M. Salazar, A. Censi, E. Frazzoli, and M. Pavone, "On the co-design of av-enabled mobility systems," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–8. DOI: 10.1109/ITSC45102.2020.9294499.

[185] G. Zardini, N. Lanzetti, M. Salazar, A. Censi, E. Frazzoli, and M. Pavone, "Towards a co-design framework for future mobility systems," in *Annual Meeting of the Transportation Research Board*, Washington D.C., United States, Jan. 2020.

[186] G. Zardini, N. Lanzetti, A. Censi, E. Frazzoli, and M. Pavone, "Co-design to enable user-friendly tools to assess the impact of future mobility solutions," *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 2, pp. 827–844, 2023. DOI: 10.1109/TNSE.2022.3223912.

[187] G. Zardini, N. Lanzetti, M. Pavone, and E. Frazzoli, "Analysis and control of autonomous mobility-on-demand systems," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, no. 1, 2022. DOI: 10.1146/annurev-control-042920-012811.

[188] T. Yigitcanlar, M. Wilson, and M. Kamruzzaman, "Disruptive impacts of automated driving systems on the built environment and land use: An urban planner's perspective," *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 5, no. 2, p. 24, 2019. DOI: 10.3390/joitmc5020024.

[189] M. Salazar, N. Lanzetti, F. Rossi, M. Schiffer, and M. Pavone, "Intermodal autonomous mobility-on-demand," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3946–3960, 2020. DOI: 10.1109/TITS.2019.2950720.

[190] R. Z. Farahani, E. Miandoabchi, W. Y. Szeto, and H. Rashidi, "A review of urban transportation network design problems," *European Journal of Operational Research*, vol. 229, pp. 281–302, 2013. DOI: 10.1016/j.ejor.2013.01.001.

[191] V. Guihaire and J.-K. Hao, "Transit network design and scheduling: A global review," *Transportation Research Part B: Methodological*, vol. 42, pp. 1251–1273, 2008. DOI: 10.1016/j.tra.2008.03.011.

[192] A. Loder, M. C. Bliemer, and K. W. Axhausen, "Optimal pricing and investment in a multi-modal city — introducing a macroscopic network design problem based on the mfd," *Transportation Research Part A: Policy and Practice*, vol. 156, pp. 113–132, 2022. DOI: https://doi.org/10.1016/j.tra.2021.11.026.

[193] Z. Cong, B. De Schutter, and R. Babuska, "Co-design of traffic network topology and control measures," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 56–73, 2015. DOI: 10.1016/j.trc.2015.01.031.

[194] Q. Luo, S. Li, and R. C. Hampshire, "Optimal design of intermodal mobility networks under uncertainty: Connecting micromobility with mobility-on-demand transit," *EURO Journal on Transportation and Logistics*, vol. 10, p. 100 045, 2021. DOI: 10.1016/j.ejtl.2021.100045. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2192437621000170.

[195] R. O. Arbex and C. B. da Cunha, "Efficient transit network design and frequencies setting multi-objective optimization by alternating objective genetic algorithm," *Transportation Research Part B: Methodological*, vol. 81, pp. 355–376, 2015. DOI: 10.1016/j.trb.2015.06.014.

[196] L. Sun, J. G. Jin, D.-H. Lee, K. W. Axhausen, and A. Erath, "Demand-driven timetable design for metro services," *Transportation Research Part C: Emerging Technologies*, vol. 46, pp. 284–299, 2014. DOI: 10.1016/j.trc.2014.06.003.

[197] S. Su, X. Li, T. Tang, and Z. Gao, "A subway train timetable optimization approach based on energy-efficient operation strategy," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 883–893, 2013. DOI: 10.1109/TITS.2013.2244885.

[198] J. A. Barrios and J. D. Godier, "Fleet sizing for flexible carsharing systems: Simulation-based approach," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2416, pp. 1–9, 2014. DOI: 10.3141/2416-01.

[199] D. J. Fagnant and K. M. Kockelman, "Dynamic ride-sharing and fleet sizing for a system of shared autonomous vehicles in austin, texas," *Transportation*, vol. 45, no. 1, pp. 143–158, 2018. DOI: 10.1007/s11116-016-9729-z.

[200] M. M. Vazifeh, P. Santi, G. Resta, S. H. Strogatz, and C. Ratti, "Addressing the minimum fleet problem in on-demand urban mobility," *Nature*, vol. 557, no. 7706, p. 534, 2018. DOI: 10.1038/s41586-018-0095-1.

[201] P. M. Boesch, F. Ciari, and K. W. Axhausen, "Autonomous vehicle fleet sizes required to serve different levels of demand," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2542, no. 1, pp. 111–119, 2016. DOI: 10.3141/2542-13.

[202] M. Meghjani, S. D. Pendleton, K. A. Marczuk, *et al.*, "Multi-class fleet sizing and mobility on demand service," in *International Conference on Complex Systems Design & Management*, Springer, vol. 878, 2018, pp. 37–49. DOI: 10.1007/978-3-030-02886-2_4.

[203] J. Narayan, O. Cats, N. van Oort, and S. P. Hoogendoorn, "Fleet size determination for a mixed private and pooled on-demand system with elastic demand," *Transportmetrica A: Transport Science*, vol. 17, no. 4, pp. 897–920, 2021. DOI: 10.1080/23249935.2020.1819910.

[204] K. Spieser, K. Treleaven, R. Zhang, E. Frazzoli, D. Morton, and M. Pavone, "Toward a systematic approach to the design and evaluation of automated mobility-on-demand systems: A case study in singapore," in *Road vehicle automation*, Springer, 2014, pp. 229–245. DOI: 10.1007/978-3-319-05990-7_20.

[205] H. Zhang, C. J. Sheppard, T. E. Lipman, and S. J. Moura, "Joint fleet sizing and charging system planning for autonomous electric vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4725–4738, 2020. DOI: 10.1109/TITS.2019.2946152.

[206] G. J. Beaujon and M. A. Turnquist, "A model for fleet sizing and vehicle allocation," *Transportation Science*, vol. 25, no. 1, pp. 19–45, 1991. DOI: 10.1287/trsc.25.1.19.

[207] A. Wallar, W. Schwarting, J. Alonso-Mora, and D. Rus, "Optimizing multi-class fleet compositions for shared mobility-as-a-service," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 2998–3005. DOI: 10.1109/ITSC.2019.8916904.

[208] H. K. Pinto, M. F. Hyland, H. S. Mahmassani, and I. Ö. Verbas, "Joint design of multimodal transit networks and shared autonomous mobility fleets," *Transportation Research Part C: Emerging Technologies*, vol. 113, pp. 2–20, 2020. DOI: 10.1016/j.trc.2019.06.010.

[209] T. Seo and Y. Asakura, "Multi-objective linear optimization problem for strategic planning of shared autonomous vehicle operation and infrastructure design," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 4, pp. 3816–3828, 2022. DOI: 10.1109/TITS.2021.3071512.

[210] S. Shaheen and A. Cohen, "Shared micromoblity policy toolkit: Docked and dockless bike and scooter sharing," 2019.

[211] M. Ghamami and M. Shojaei, "Introducing a design framework for a multi-modal public transportation system, focusing on mixed-fleet bike-sharing systems," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2672, no. 36, pp. 103–115, 2018. DOI: 10.1177/0361198118799.

[212] C.-C. Lu, "Robust multi-period fleet allocation models for bike-sharing systems," *Networks and Spatial Economics*, vol. 16, no. 1, pp. 61–82, 2016. DOI: 10.1007/s11067-013-9203-9.

[213] S. Yan, C.-C. Lu, and M.-H. Wang, "Stochastic fleet deployment models for public bicycle rental systems," *International Journal of Sustainable Transportation*, vol. 12, no. 1, pp. 39–52, 2018. DOI: 10.1080/15568318.2017.1324586.

[214] Q. Luo, S. Li, and R. C. Hampshire, "Optimal design of intermodal mobility networks under uncertainty: Connecting micromobility with mobility-on-demand transit," *EURO Journal on Transportation and Logistics*, p. 100 045, 2021.

[215] J. Y. Chow and H. R. Sayarshad, "Symbiotic network design strategies in the presence of coexisting transportation networks," *Transportation Research Part B: Methodological*, vol. 62, pp. 13–34, 2014. DOI: 10.1016/j.trb.2014.01.008.

[216] D. Kondor, X. Zhang, M. Meghjani, P. Santi, J. Zhao, and C. Ratti, "Estimating the potential for shared autonomous scooters," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2021. DOI: 10.1109/TITS.2020.3047141.

[217] A. Henao and W. E. Marshall, "The impact of ride-hailing on vehicle miles traveled," *Transportation*, vol. 46, no. 6, pp. 2173–2194, 2019. DOI: 10.1007/s11116-018-9923-2.

[218] P. Blackwell, K. Carter-Cram, E. Pape, and S. Islam, "E-scooter impact on traffic congestion," 2019.

[219] C. F. Daganzo and N. Geroliminis, "An analytical approximation for the macroscopic fundamental diagram of urban traffic," *Transportation Research Part B: Methodological*, vol. 42, no. 9, pp. 771–781, 2008. DOI: 10.1016/j.trb.2008.06.008.

[220] D. Dahl. "If you're annoyed at drivers going under the speed limit, the problem isn't them, it's you." available online, The Bellingham Herald. (2018).

[221] A. Mas-Colell, M. D. Whinston, and J. R. Green, *Microeconomic Theory*. Oxford Univ. Press, 1995.

[222] A. Horni, K. Nagel, and K. W. Axhausen, Eds., *The Multi-Agent Transport Simulation MATSim*. Ubiquity Press, 2016.

[223] M. Haklay and P. Weber, "OpenStreetMap: User-generated street maps," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, 2008. DOI: 10.1109/MPRV.2008.80.

[224] GTFS. "GTFS: Making public transit data universally accessible." available online at https://gtfs.org/. (2019).

[225] ODDC. "Taxicab trips in 2016." available online at https://opendata.dc.gov/search?q=taxicabs, Open Data DC. (2017).

[226] PIM. "Metrorail ridership by origin and destination." available online at https://planitmetro.com/2012/10/31/data-download-metrorail-ridership-by-origin-and-destination/, Plan It Metro. (2012).

[227] F. Siddiqui. "As ride hailing booms in d.c., it's not just eating in the taxi market – it's increasing vehicle trips." available online, The Washington Post. (2018).

[228] DoA, Ed., *Military Police Traffic Operations*. Department of the Army, 1977.

[229] S. Dixon, H. Irshad, and V. White, "Deloitte city moblity index – washington d.c.," Deloitte, Tech. Rep., 2018.

[230] N. Pavlenko, P. Slowik, and N. Lutsey, "When does electrifying shared mobility make economic sense?" The International Council on Clean Transportation, Tech. Rep., 2019.

[231] P. M. Boesch, F. Becker, H. Becker, and K. W. Axhausen, "Cost-based analysis of autonomous mobility services," *Transport Policy*, vol. 64, pp. 76–91, 2018. DOI: 10.1016/j.tranpol.2017.09.005.

[232] D. J. Fagnant and K. Kockelman, "Preparing a nation for autonomous vehicles: Opportunities, barriers and policy recommendations," *Transportation Research Part A: Policy and Practice*, vol. 77, pp. 167–181, 2015. DOI: 10.1016/j.tra.2015.04.003.

[233] G. S. Bauer, J. B. Greenblatt, and B. F. Gerke, "Cost, energy, and environmental impact of automated electric taxi fleets in manhattan," *Environmental Science & Technology*, vol. 52, no. 8, pp. 4920–4928, 2018. DOI: 10.1021/acs.est.7b04732.

[234] T. Litman, "Autonomous vehicle implementation predictions – implications for transport planning," Victoria Transport Policy Institute, Tech. Rep., 2019.

[235] Z. Wadud, "Fully automated vehicles: A cost of ownership analysis to inform early adoption," *Transportation Research Part A: Policy and Practice*, vol. 101, pp. 163–176, 2017. DOI: 10.1016/j.tra.2017.05.005.

[236] W. Time. "Carbon footprint data." Available at https://api.watttime.org, Wired. (Mar. 2018).

[237] D. Schellong, P. Sadek, C. Schaetzberger, and T. Barrack, "The promise and pitfalls of e-scooter sharing," Boston Consulting Group, Tech. Rep., 2019.

[238] D. C.-H. Chao, P. J. van Duijsen, J. J. Hwang, and C.-W. Liao, "Modeling of a taiwan fuel cell powered scooter," in *2009 International Conference on Power Electronics and Drive Systems (PEDS)*, 2009, pp. 913–919. DOI: 10.1109/PEDS.2009.5385788.

[239] G. of the District of Columbia, "District of columbia, capitel bikeshare development plan," District of Columbia, Tech. Rep., 2015.

[240] S. Korus. "Electric scooters: The unit economics may spell trouble." available online, ARK Invest. (2019).

[241] J. Hollingsworth, B. Copeland, and J. X. Johnson, "Are e-scooters polluters? the environmental impacts of shared dockless electric scooters," *Environmental Research Letters*, vol. 14, no. 8, p. 084 031, 2019. DOI: 10.1088/1748-9326/ab2da8.

[242] Z. Kou, X. Wang, S. F. A. Chiu, and H. Cai, "Quantifying greenhouse gas emissions reduction from bike share systems: A model considering real-world trips and transportation mode choice patterns," *Resources, Conservation and Recycling*, vol. 153, p. 104 534, 2020. DOI: 10.1016/j.resconrec.2019.104534.

[243] P. Van Zyl, P. van Mensch, N. Ligterink, R. Droege, and G. Kadijk, "Update emission model for two wheeled mopeds," *TNO report, TNO*, R11088, 2014.

[244] WMATA, "Fy2018 proposed budget," Washington Metropolitan Area Transit Authority, Tech. Rep., 2017.

[245] L. Aratani. "Metro to debut first of its 7000-series cars on blue line on april 14." available online, The Washington Post. (2015).

[246] WMATA, "Sustainability report 2018," Washington Metropolitan Area Transit Authority, Tech. Rep., 2018.

[247] H. Becker, F. Becker, R. Abe, *et al.*, "Impact of vehicle automation and electric propulsion on production costs for mobility services worldwide," *Transportation Research Part A: Policy and Practice*, vol. 138, pp. 105–126, 2020. DOI: 10.1016/j.tra.2020.04.021.

[248] J. H. Gawron, G. A. Keoleian, R. D. De Kleine, T. J. Wallington, and K. Hyung Chul, "Life cycle assessment of connected and automated vehicles: Sensing and computing subsystem and vehicle level effects," *Environmental Science & Technology*, vol. 52, pp. 3249–3256, 2018. DOI: 10.1021/acs.est.7b04576.

[249] WCP, "The automotive lidar market," Woodside Capital Partners, Tech. Rep., 2018.

[250] P. Lienert. "Cost of driverless vehicles to drop dramatically: Delphi ceo." available online, Insurance Journal. (2019).

[251] K. Korosec. "Uber spent usd 457 million on self-driving and flying car r&d last year." available online, TechCrunch. (2019).

[252] P. Howard and D. Sylvan, "Expert consensus on the economics of climate change," Institute for Policy Integrity – New York University School of Law, Tech. Rep., 2015.

[253] T Etezadi and J. Beasley, "Vehicle fleet composition," *Journal of the Operational Research Society*, vol. 34, pp. 87–91, 1983. DOI: 10.1057/jors.1983.11.

[254] A. Zanardi, G. Zardini, and S. Srinivasan, S. Bolognani, A. Censi, F. Dörfler, E. Frazzoli, "Posetal games: Efficiency, existence, and refinement of equilibria in games with prioritized metrics," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1292–1299, 2022. DOI: 10.1109/LRA.2021.3135030.

[255] N. Lanzetti, G. Zardini, M. Schiffer, M. Ostrovsky, and M. Pavone, "Do self-driving cars swallow public transport? a game-theoretical perspective on transportation systems," en, in *INFORMS Annual Meeting 2019*, Informs, 2019-10-22.

[256] G. Zardini, N. Lanzetti, L. Guerrini, E. Frazzoli, and F. Dörfler, "Game theory to study interactions between mobility stakeholders," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, Best Paper Award (1st place), 2021, pp. 2054–2061. DOI: 10.1109/ITSC48978.2021.9564501.

[257] G. Zardini, N. Lanzetti, G. Belgioioso, C. Hartnik, S. Bolognani, F. Dörfler, E. Frazzoli, "Strategic interactions in multi-modal mobility systems: A game-theoretic perspective," in *2023 IEEE International Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2023.

[258] A. Censi, E. Frazzoli, J. Lorand, and G. Zardini, "Categorification of negative information using enrichment," in *Proceedings Fifth International Conference on Applied Category Theory, Glasgow, United Kingdom, 18-22 July 2022*, J. Master and M. Lewis, Eds., ser. Electronic Proceedings in Theoretical Computer Science, vol. 380, Open Publishing Association, 2023, pp. 22–40. DOI: 10.4204/EPTCS.380.2.

[259] G. Freiling and A. Hochhaus, "Properties of the solutions of rational matrix difference equations," *Computers & Mathematics with Applications*, vol. 45, no. 6-9, pp. 1137–1154, 2003.