Diss. ETH No. 27854

Towards Guarantees for Adaptive Embedded Systems in Uncertain Environments

A thesis submitted to attain the degree of

Doctor of Sciences of ETH Zurich
(Dr. sc. ETH Zurich)

presented by
STEFAN DRAŠKOVIĆ
MSc ETH EEIT, ETH Zurich

born on 29.08.1990
citizen of Serbia

accepted on the recommendation of
Prof. Dr. Lothar Thiele, examiner
Prof. Dr.-Ing. Rolf Ernst, co-examiner

2021

Institut für Technische Informatik und Kommunikationsnetze
Computer Engineering and Networks Laboratory

TIK-SCHRIFTENREIHE NR. 192

STEFAN DRAŠKOVIĆ

Towards Guarantees for Adaptive Embedded Systems in Uncertain Environments

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

A dissertation submitted to
ETH Zurich
for the degree of Doctor of Sciences

DISS. ETH NO. 27854

Prof. Dr. Lothar Thiele, examiner
Prof. Dr.-Ing. Rolf Ernst, co-examiner

Examination date: 26 August 2021

DOI 10.3929/ethz-b-000522305

Мојој породици
и свима који су веровали у мене

Abstract

Contemporary embedded systems are making their way into many application domains where strict requirements on their correct and timely operation apply. Examples range from traditional safety-critical systems, which must adhere to hard real-time constraints, to distributed sensor networks needing to provide a reliable minimal service level. Yet, the very trends popularizing embedded systems are causing detrimental unpredictability in their operation.

A prime example is energy harvesting, which promises to power embedded devices indefinitely. Relying on energy from the environment introduces substantial uncertainness to a system's operation due to the natural volatility and intermittence of the sources. To this end, embedded systems are often designed with adaptive operation in mind. Energy controllers can thus choose an appropriate mode of operation, either reacting to the system's state or proactively estimating the environment's future.

Another trend is the increase in complexity. Hardware advances often optimize the average performance while sacrificing timing predictability. Thus, in combination with modern software, task execution times become more and more variable, and possible but unlikely very long execution times occur. Oversizing computational resources to ensure correct operation for every rare scenario is often impossible, and adaptive operation again offers itself as a sensible alternative. An example is mixed criticality scheduling, where the performance of safety-critical tasks is always guaranteed, while it is acceptable to occasionally enter an operating mode where less important tasks are dropped or executed in a low resource mode.

Unfortunately, with the introduction of adaptability, it becomes a challenge to guarantee all requirements. Regardless of the constraints, a system's operation is influenced not only by the environment's non-determinism but also by the adaption strategy of the system. This dissertation investigates the influence of the environment on an adaptive embedded system and what kind of guarantees can be given. The two examples of an uncertain environment given above are studied, first when the available energy depends on the environment and then when the computational resources depend on the execution times of current and past tasks. Our main contributions are:

- For energy harvesting embedded systems, we discuss how to optimally use a backup battery, such that the system's lifetime and long-term utility are maximized while a minimal service level is guaranteed. In the solar energy harvesting scenario, we present novel estimators of future energy availability, which are used to further reduce the variability in operation by enabling a system to adapt to its environment proactively.
- We formalize a novel Markov analysis of energy harvesting systems, which also considers the uncertainty of energy consumption created by variability in execution times of tasks. With this analysis, we provide useful probabilistic performance metrics: the probability of failure due to a lack of energy or the probability that the system operates in a particular mode.
- Trace-based and real-world experiments illustrate the aforementioned performance guarantees in diverse solar energy harvesting scenarios, both for indoor and outdoor locations. Finally, the accuracy of our analysis is verified, as it can consider many implementation artifacts.
- In the mixed criticality domain, we use a Markov analysis of pending executions to present guarantees on the probability of deadline miss per hour, globally and in each mode of operation, and the probability that a task is executed in degraded mode. Our proposed scheduling scheme is shown to clearly outperform state-of-the-art solutions.

Zusammenfassung

Gegenwärtig halten eingebettete Systeme Einzug in viele Anwendungsbereiche, in denen strenge Anforderungen an ihren korrekten und zeitgerechten Betrieb gelten. Die Beispiele reichen von traditionellen sicherheitskritischen Systemen, die sich an harte Echtzeitbedingungen halten müssen, bis hin zu verteilten Sensornetzen, die ein zuverlässiges Mindestmaß an Diensten bieten müssen. Doch genau die Trends, die eingebettete Systeme populär machen, führen zu nachteiliger Unvorhersehbarkeit in deren Betrieb.

Ein Musterbeispiel ist das Energy Harvesting, das eine unbegrenzte Energieversorgung für eingebettete Geräte verspricht. Die Nutzung von Umweltenergie schafft aufgrund der natürlichen Schwankungen und Unterbrechungen der Energiequellen erhebliche Unsicherheiten für den Betrieb eines Systems. Aus diesem Grund werden eingebettete Systeme häufig für einen adaptiven Betrieb konzipiert. Energieregler können so einen passenden Betriebsmodus wählen, indem sie entweder auf den Zustand des Systems reagieren oder proaktiv die zukünftige Umgebung abschätzen.

Ein weiterer Trend ist die Zunahme der Komplexität. Weiterentwicklungen der Hardware optimieren oft die durchschnittliche Rechenleistung auf Kosten der Vorhersehbarkeit bei den Zeitabläufen. In Kombination mit moderner Software ergeben sich daher immer größere Schwankungen bei den Ausführungszeiten, und es entstehen mögliche, aber seltene, sehr lange Ausführungszeiten. Eine Überdimensionierung der Rechenressourcen, um einen korrekten Betrieb für jedes ausgefallene Szenario zu gewährleisten, ist oft nicht möglich, und auch hier bietet sich ein adaptiver Betrieb als sinnvolle Alternative an. Ein Beispiel ist Mixed-Criticality-

Scheduling, wo die Ausführungsqualität sicherheitskritischer Aufgaben immer gewährleistet ist, während man in Kauf nimmt, gelegentlich in einen Betriebsmodus einzutreten, bei dem weniger wichtige Aufgaben fallengelassen oder in einem ressourcenreduzierten Modus ausgeführt werden.

Leider steigt mit der Einbeziehung der Adaptivität die Herausforderung, alle notwendigen Vorgaben einzuhalten. Davon abgesehen wird der Betrieb eines Systems nicht nur durch den Nichtdeterminismus der Umgebung, sondern auch durch seine Anpassungsstrategie beeinflusst. Diese Dissertation beleuchtet den Einfluss der Umwelt auf ein adaptives eingebettetes System und untersucht, welche Art von Garantien gegeben werden können. Die beiden oben genannten Beispiele einer unsicheren Umgebung werden betrachtet: Zunächst, wenn die verfügbare Energie von der Umgebung abhängt, und anschließend, wenn die Rechenressourcen von den Ausführungszeiten der aktuellen und vergangenen Aufgaben abhängen. Dies sind unsere wichtigsten Beiträge:

- Für eingebettete Systeme mit eigenständiger Energiegewinnung erörtern wir, wie eine Backup-Batterie optimal genutzt werden kann, so dass die Lebensdauer des Systems und der langfristige Nutzen maximiert werden, aber gleichzeitig ein minimales Serviceniveau stets gewährleistet ist. Im Kontext der Solarenergienutzung stellen wir neuartige Abschätzungsmethoden für die zu erwartende Energieverfügbarkeit vor, die dazu dienen, die Betriebsschwankungen weiter zu verringern, indem ein System in die Lage versetzt wird, sich proaktiv an seine Umgebung anzupassen.
- Wir formalisieren eine neuartige Markov-Analyse von Energy-Harvesting-Systemen, die auch jene Unsicherheit des Energieverbrauchs berücksichtigt, die durch die Variabilität der Ausführungszeiten entsteht. Mit dieser Analyse stellen wir nützliche probabilistische Leistungsmetriken zur Verfügung: Die Wahrscheinlichkeit eines Ausfalls aufgrund von Energiemangel oder auch die Wahrscheinlichkeit, dass das System in einem bestimmten Modus arbeitet.

- Aufzeichnungsbasierte und praktische Experimente veranschaulichen die oben genannten Leistungsgarantien in verschiedenen Szenarien der Solarenergiegewinnung, sowohl für Innen- als auch für Außenstandorte. Schließlich wird die Genauigkeit unserer Analyse anhand vieler Implementierungsartefakte verifiziert, welche zu berücksichtigen sie in der Lage ist.
- Im Mixed-Criticality-Bereich verwenden wir eine Markov-Analyse der anstehenden Ausführungen, um Garantien für die Wahrscheinlichkeit einer Terminüberschreitung pro Stunde, sowohl global als auch in jedem Betriebsmodus, vozustellen, sowie die Wahrscheinlichkeit, dass eine Aufgabe im reduzierten Modus ausgeführt wird. Das von uns vorgeschlagene Scheduling-Konzept, wird gezeigt, übertrifft bisherige Lösungen deutlich.

Résumé

Les systèmes modernes embarqués s'intègrent de mieux en mieux dans de nombreux domaines d'application caractérisés par des exigences strictes en matière de bon fonctionnement et de précision temporelle. Les exemples vont des systèmes traditionnels de sécurité soumis à des contraintes exigeantes en temps réel, aux réseaux de capteurs distribués devant fournir un certain minimum de fiabilité. Pourtant, ces mêmes tendances rendant les systèmes embarqués populaires d'utilisation causent une imprévisibilité préjudiciable à leur fonctionnement.

Une illustration type est la récupération d'énergie, qui promet d'alimenter infiniment les systèmes embarqués. Toutefois, des incertitudes notables sont introduites dans l'opération d'un système dépendant de l'énergie de l'environnement en raison de la volatilité naturelle et de l'intermittence des sources. À cette fin, les systèmes embarqués sont souvent conçus pour permettre un fonctionnement adaptatif. En réagissant à l'état du système, ou en anticipant l'environnement futur, les contrôleurs d'énergie peuvent ainsi choisir un mode d'opération approprié.

L'augmentation de la complexité est aussi une autre tendance. En effet, l'amélioration du hardware optimise souvent la performance moyenne au détriment de la prévisibilité temporelle. Combiné à du software moderne, le temps d'exécution des tâches devient de plus en plus variable et dans de rares cas, bien que possible, peut donner lieu à de longs temps d'exécutions. Garantir un bon fonctionnement pour chaque cas rare en surdimensionnant les ressources de calcul est souvent impossible et une opération adaptative s'avère être une alternative judicieuse. Un exemple est le

mixed-criticality-scheduling, où la performance des tâches critiques pour la sécurité est toujours assurée, tout en acceptant d'entrer occasionnellement dans un mode d'opération dans lequel des tâches moins importantes peuvent être exécutées à faible ressources voire abandonnées.

Malheureusement, avec l'introduction de l'adaptabilité, il devient plus délicat de satisfaire toutes les exigences. Indépendamment des contraintes, le fonctionnement de tels systèmes est non seulement influencé par le non-déterminisme de l'environnement mais aussi par la stratégie d'adaptation choisie par le système. Cette thèse étudie l'influence de l'environnement sur un système embarqué adaptatif et le type de garanties qui peuvent s'ensuivre. Les deux exemples d'environnement incertain cités auparavant sont examinés, tout d'abord lorsque l'énergie disponible dépend de l'environnement et ensuite lorsque les ressources de calculs dépendent des temps d'exécution des tâches présentes et passées. Nos principales contributions sont les suivantes :

- Concernant les systèmes embarqués de collecte d'énergie, il est discuté de la manière optimale d'utiliser une batterie de secours de sorte à maximiser la durée de vie et l'utilisation à long terme du système, tout en garantissant un niveau de service minimum. Dans le contexte de la récolte d'énergie solaire, de nouveaux estimateurs de la disponibilité future d'énergie sont présentés, réduisant d'avantage la variabilité pendant le fonctionnement en autorisant le système à s'adapter proactivement à son environnement.
- Nous formalisons une nouvelle analyse de Markov des systèmes de récolte d'énergie, considérant aussi l'incertitude de la consommation d'énergie crée par la variabilité des temps d'exécution des tâches. Grâce à cette nouvelle analyse, des indicateurs probabilistes de performance sont fournis : la probabilité d'une panne due à un manque d'énergie ou la probabilité qu'un système opère dans un mode particulier.

- Des expériences du monde réel et basées sur des enregistrements illustrent les garanties de performances évoquées ci-dessus dans divers scenarios de récolte d'énergie solaire, à l'intérieur comme à l'extérieur. Enfin, la précision de notre analyse est vérifiée puisqu'elle peut tenir compte de beaucoup d'artéfact d'implémentation.
- Dans le domaine du mixed-criticality, nous utilisons une analyse de Markov des exécutions en attente afin de présenter des garanties sur la probabilité de dépassement d'échéance par heure, globalement et dans chaque mode de fonctionnement, et la probabilité qu'une tâche soit exécutée dans un mode réduit. Notre schéma d'ordonnancement s'avère surpasser les solutions dernier cri.

Сажетак

Савремени ембедед системи проналазе примену у многим областима где важе строги захтеви за исправан и благовремен рад. Примери се могу наћи од традиционалних система где је сигурност кључна, и који се морају придржавати временских ограничења, до дистрибуираних сензорских мрежа које поузданo пружају минимални ниво услуге. Ипак, исти трендови који популаризују ембедед системе изазивају штетну непредвидивост у њиховом раду.

Најбољи пример је прикупљање енергије из околине, које обећава вечно напајање ембедед уређаја. Ослањање на енергију присутну у околини уноси значајну нестабилност у рад система, због природне несталности и испрекидане доступности енергетског извора. Због тога, ембедед системи су често направљени са могућношћу прилагодљивог рада. Контролер енергије тако може изабрати одговарајући режим рада, било да реагује на стање система или да проактивно процењује будућност околине.

Још један тренд је повећање комплексности. Напредак хардвера често доводи до побољшања просечних перформанси, док се жртвује предвидивост временског одзива. Тако, у комбинацији са савременим софтвером, време извршавања задатака постаје све променљивије, и могуће је иако мало вероватно да дође до веома дуготрајних времена извршавања. Предимензионисање рачунарских ресурса како би се обезбедио исправан рад у сваком ретком случају је често немогуће, па је прилагодљив рад и овде разумна алтернатива. Пример је распоређивање задатака мешовите критичности, где је

извршавање сигурносно-критичних задатака увек загарантовано, док је прихватљиво повремено ући у режим рада где се мање важни задаци не извршавају или се извршавају у режиму са смањеним ресурсима.

Нажалост, код прилагодљивих система, постаје изазов гарантовати исправан рад. Без обзира на задата ограничења, на рад система не утиче само променљиво окружење, већ и стратегија прилагођавања. Ова дисертација истражује утицај окружења на прилагодљиве ембедед системе, и какве се гаранције на рад могу давати. Проучавају се два поменута примера несталног окружења, прво када расположива енергија зависи од околине, а затим када рачунарски ресурси зависе од времена извршавања тренутних и прошлих задатака. Наши главни доприноси су следећи:

- За ембедед системе који прикупљају енергију из околине, дискутујемо како оптимално користити резервну батерију, тако да су век трајања система и мера корисности максимални, док се истовремено минималан ниво услуге гарантује. У случају када се прикупља соларна енергија, представљамо нове методе прогнозирања доступности енергије које систем користи да се проактивно прилагоди свом окружењу и тако смањи варијабилност у режиму рада.
- Формализујемо нову Марковљеву анализу система који прикупљају енергију из околине, која такође узима у обзир несигурност потрошње енергије због варијабилности времена извршавања задатака. Овом анализом пружамо корисне стохастичке мере о перформансама: вероватноћу квара због недостатка енергије или вероватноћу да ће систем радити у одређеном режиму.
- Практични експерименти и експерименти засновани на измереним записима илуструју горе наведене гаранције перформанси у разноврсним примерима прикупљања соларне енергије, како на отвореном тако и у затвореном простору. Коначно, верификована је и тачност наше анализе, па она може узети у обзир многе нуспродукте имплементације.

- У домену мешовите критичности, користимо Марковљеву анализу предстојећих извршења задатака да представимо гаранцију која се односи на вероватноћу пропуштања задатог рока по сату, глобално за цео систем као и за сваки режим рада посебно, и вероватноћу да ће задатак бити извршен у режиму са деградацијом. Показало се да концепт распоређивања који смо предложили јасно надмашује најсавременија решења из праксе.

Acknowledgements

Looking back towards my many years at ETH, I can not help but remember all the people that made this doctoral thesis possible. This work is much more than an answer to a research assignment, it is the result of an incredible journey that made me the person I am today.

First and foremost, I would like to thank my mentor Prof. Lothar Thiele. He set the goal of excellent research, which defined the Computer Engineering group (TEC) I had the honor to be a part of, and kept us all moving forward. With his tolerant and careful guidance, I have managed to overcome all obstacles that emerged. Whether I was struggling to formalize an idea or present my work clearly, I could always rely on helpful discussions with him. Likewise, I would like to thank Prof. Rolf Ernst for his interest in my thesis and his insightful review. I am also grateful to Prof. Laurent Vanbever for serving on the examination committee.

I would like to express my heartfelt appreciation to the many friends and colleagues from the Computer Engineering group. I have to start with Pengcheng Huang, who welcomed me into the world of Mixed Criticality and aided me in writing my first paper. Likewise, I thank Andrés Gómez for introducing me to energy harvesting systems and for teaching me many tricks of the doctorand trade. I am also grateful for the extended cooperation with Rehan Ahmed. Without our many thorough discussions on theoretical subjects, this thesis would not have been what it is today.

I worked in most likely the nicest office of Gloriastrasse 35, G 81, with wonderful office mates Andreas Tretter, Matthias Meyer, Lukas Sigrist, and Naomi Stricker. Andreas was a senior doctorand

when I arrived, and he supported me daily as I was getting used to academia. In this regard, I also thank the 'old' TEC members Georgia Giannopoulou, Roman Lim, and Felix Sutton. Matthias, Lukas, and I did most of our studies concurrently, as did Roman Trüb, Romain Jacob, Balz Maag, and Philipp Miedl. We had many fun times together, from our first steps until graduation. Finally, I want to send my best wishes to the newer 'TECies'. Naomi and Andreas Biri were always there to lend a helping hand as my doctoral exam drew nearer. Thanks to their dedication, and to Xiaoxi He, Yun Cheng, and Zhongnan Qu, we successfully fulfilled all teaching duties even in the face of the coronavirus pandemic.

Reto Da Forno and Tonio Gsell were instrumental in setting up experiments and practical set-ups, for which I am much obliged. Last but not least, I would like to state my appreciation to the senior members of TEC, Jan Beutel, Olga Saukh, Zimu Zhou, and Rehan. I found their devotion to research and teaching inspiring.

Throughout my doctorate, I had the pleasure to supervise numerous students. One way or another, they have all contributed to my research: Kevin Keller, Cong Lin, Luca Stalder, Szebedy Bence, Joel Büsser, David Geiter, Gregory Wüest, Colin Berner, Marvin Häberle, Francisco Duvauchelle, Weikang Kong, Markus Kiser, Alexandra Schneider, and Jannik William.

I am deeply grateful to Beat Futterknecht, whom I could depend on for every administrative or practical problem I had. He ensured that I always felt at home at ETH and in Zürich.

A special and most personal thank you goes out to my father Mijomir, mother Zlatica, sister Marina and grandmother Smilja for their sincere and unconditional support in all of my endeavors.

Contents

List of Figures

List of Tables

List of Algorithms

Acronyms

AMC adaptive mixed-criticality.
Astro astronomical prediction.

Delta-T delta – transmittance.
DMPO deadline monotonic priority ordering.

ENO energy-neutral operation.
EnoMax optimal reactive scheme under ENO constraints.
EWMA exponentially weighted moving average.
EWMA-T exponentially weighted moving average – transmittance.

FHC finite horizon control.

MAE mean absolute error.
MAPE mean absolute percentage error.
MC mixed-criticality.

pDMPO probabilistic deadline monotonic priority ordering.
pMC probabilistic mixed-criticality.
ProEnergy profile energy prediction model.
ProEnergy-T profile energy prediction model – transmittance.
pWCET probabilistic worst-case execution time.

UB-HL upper bound on fixed priority preemptive MC schemes.

WCET worst-case execution time.
WCMA weather-conditioned moving average.
WCMA-T weather-conditioned moving average – transmittance.

1

Introduction

Embedded systems have become pervasive and are now commonly found in many applications where their correct and timely operation needs to be guaranteed. Traditionally, this is the case for hard real-time systems, mainly in the automotive or avionics domains, where strict industry standards regarding safety apply [ISO18, RE12]. However, recent examples of such requirements have emerged in the field of wireless sensor networks, where networks deployed in remote environments provide reliable and precise long-term measurements [WBDF+19] for applications such as environmental monitoring [WFM+18] as well as natural hazard warnings [MFCP+19]. Dependable operation can likewise be found with cyber-physical systems, where a study demonstrates how automatic control loops can be established over low-power wireless links [MBJ+19].

Nonetheless, the same trends that are popularizing embedded systems are making their operation harder to understand and predict. A prime example is energy harvesting. It has emerged as a way to power embedded devices indefinitely, which is particularly important in remote and hard-to-reach places. Coupled with the associated scalability and low maintenance, the benefits for large-scale distributed systems are evident.

But one of the biggest challenges with energy harvesting embedded systems is ensuring reliability and determinism in their operation [BASM16]. Indeed, the smaller the internal energy storage capacity of a node, the higher is the sensitivity of its operation to short-term non-deterministic changes of the harvested power. At the same time, energy harvested from the environment is usually volatile and intermittent, especially solar energy.

To deal with this inherent uncertainty, an energy management subsystem is designed to control the energy flow between the energy source, the rechargeable energy storage, and the consumer. It also adapts the consumer's operation, such that the system's lifetime is prolonged or an overall utility is maximized. This adaptive operation can be realized in numerous ways, such as with the simple adjustment of sensing, processing, or communication rates. Further adaptation examples include changing the precision of a numeric calculation, which is commonly implemented in machine learning or iterative algorithms, or relaxing timing constraints. Otherwise, applications can even have "energy scarcity" modes of operation, specifically created for periods with unfavorable harvesting conditions.

Though general approaches to quantify performance of embedded systems adapting to uncertain environments are lacking, many solutions for particular cases exist. A large rechargeable storage can virtually isolate the consumer from its environment, though this solution is commonly not suitable due to size, safety, environmental, and cost reasons. On the other end of the spectrum, the harvesting source can be directly coupled to the consumer [BWM+14], and computing concepts that deal with frequently interrupted processing have been developed [GSM+16, RSF11]. Yet unless there is a reliable correlation between harvested energy on the one hand and actions to be performed on the other, the usefulness of such energy provisioning in critical applications is limited.

At the same time, another trend sees the increase of embedded systems' complexity across the board. Typical hardware architectures now include heterogeneous multi-core processors [SZDF+15], intricate memory hierarchies, or programable low-power accelerators [RCM+15]. These, in turn, support more advanced software.

To take an example from avionics, real-time applications traditionally isolated from one another can now be consolidated into one hardware platform, even though they were designed with different standards in mind [BD17]. Another, perhaps more prominent case, is the deployment of complex algorithms on resource constrained devices. Examples include a convolutional neural network for fast natural hazard detection on a wireless sensor network [MFCP+19] or highly parallelized ultrasound image processing for a mobile medical device [KTH+16].

Hardware advances commonly optimize a system's average performance. However, in combination with complicated software, possible but unlikely worst-case behaviors may become significantly worse. Guarantees in this regard are obtained by finding worst-case execution times of tasks, for which correct operation is verified. Unfortunately, obtaining tight upper bounds on execution times is becoming increasingly difficult. It has become intractable to measure this value by exhaustively testing combinations of inputs and internal states, while analytical methods struggle to identify worst-case paths through a program. In any case, safe values can only be provided with major pessimism.

Consequently, over-provisioning of resources to guarantee correct operation in every scenario is becoming infeasible due to size, weight, or cost constraints. A promising alternative is the use of richer system models in combination with adaptivity to cope with worst-case behavior, as is done with mixed-criticality systems. Mixed-criticality systems feature tasks with various criticality levels, which quantify a task's importance with regard to safety. In the simplest case, each task has an 'optimistic' and 'pessimistic' worst-case execution time estimate [Ves07]. During normal operation, when the optimistic estimates hold, all tasks are guaranteed to finish their execution in time. However, when tasks take longer to execute, only tasks with higher criticality levels have such guarantees.

Going even further, one may assume a stochastic model of execution times. Ideally, this means that, for each task, the execution time is modeled with an independent random variable. Though in practice a task's execution depends on many factors, existing work suggests that statistical independence from hardware states or other tasks can be achieved with certain pessimistic approximations. Such

pessimistic random variables are termed probabilistic worst-case execution times [DBG17]. Their introduction allows for improved schedulability due to the so-called multiplexer gain, meaning that the likelihood of high execution times of many tasks coinciding is very small. Despite industry's proclivity for statistical guarantees, a statistical analysis based on probabilistic worst-case execution times quantifying the potential benefit of mixed-criticality scheduling is largely lacking.

This doctoral dissertation investigates the impact of environmental factors on adaptive embedded systems and the type of guarantees one can give. Two aforementioned examples of an uncertain environment are investigated: the available energy, dependant on the harvesting environment, and the execution time of processes, dependant on the current and past inputs. We provide worst-case and statistical guarantees for systems in such scenarios.

1.1 The Uncertain Environment

Let us now take a closer look at the two aforementioned uncertain environments examples. Besides noting what makes them uncertain, we especially examine use cases where guaranteeing operation is key. These are systems that have real-time constraints timing-wise or energy-neutral constraints in the energy domain.

Types of uncertainty. In general, randomness when modeling any process comes either from the *aleatoric* variability or the *epistemic* uncertainty. Aleatoric variability means the inherent randomness, which is in our setting primarily present in the environment. Additionally, certain hardware and software elements feature built-in randomness, as is the case with random cache replacement or with the Aloha communication protocol. Epistemic uncertainty stems from our lack of knowledge. In practice, it may come from our inability to fully observe, or our limited capacity to model, a system's state or its environment. Aleatoric variability and epistemic uncertainty are, in practice, often tightly intertwined, as we will see in our two examples.

1.1.1 Energy Harvesting

A plethora of existing works demonstrates diverse ways of extracting energy from a system's surroundings. For example, in geological [BBF+11] or ecological [CB10] monitoring, energy can be harvested from the ambient environment: light, temperature differences, or wind or water flows. In an anthropogenic surrounding, energy can also be extracted from movement or electromagnetic waves. For instance, energy created by movement commonly powers wearable devices in the health monitoring domain [NAE+17], either in the form of vibration or acceleration. Still, regardless of the source, energy harvested from the environment is typically unpredictable. Consider that when estimating the future availability of harvestable solar energy, we encounter both our inability to model the atmosphere accurately and the inherent randomness of weather. To make such a supply useful, one either takes a transient or an energy-neutral approach. We explain both and highlight why guarantees are only relevant for the latter.

Transient operation. In the case of transiently powered embedded systems, a strong correlation is required to exist between the availability of harvestable energy and the target computation. Typical use-cases are an AC power meter powered by the current it is measuring [DCD13], or a low-resolution camera powered by ambient light [GSS+17]. Since they are designed to work only when sufficient energy is available, they power off or enter a deep low-power mode otherwise. As far as software is concerned, these systems utilize intermittent programming models, and many exist [RSF11, BDW+16]. On the hardware side, transient systems are designed with little or no energy buffers, using special energy management units [GSM+16] and non-volatile memory [BWM+14, JRR14].

We do not consider transient systems in this thesis. They do not allow much room for adaptability in their operation. Rather they are directly controlled by their environment and largely follow its non-deterministic behavior. Since they are required to work only when harvestable energy is available, proper dimensioning of the energy harvester is crucial for correct operation.

Energy-neutral operation. For many applications powered by harvested energy, a specified non-zero minimal service is required, despite the power source being volatile and intermittent. Here the energy-neutral paradigm is defined, which says that an embedded system needs to realize this minimal service without additional energy. To decrease the variability of harvested energy, intermediate energy buffers are used, such as super capacitors or rechargeable batteries. Yet still, the system can adapt to the energy availability by dynamically scaling its consumption. To take the common environmental monitoring example, solar energy is available during the day, especially during summer. Even though operation is expected year-round, many processing or communication tasks may be postponed to periods when more energy is available, see [BSBT14a, BSBT14b] for a mountain, [CVS+07] for a field, or [TJC08] for a forest scenario.

When designing energy-neutral systems, ideally, the variability in consumption is minimized, as this generally leads to a larger long-term utilization. Depending on how the system's consumption is decided, energy-neutral energy management schemes may be reactive or proactive. Reactive schemes dynamically adapt the system's consumption solely based on the system's internal state, primarily meaning the charge of the rechargeable storage. Notably, Vigorito et al. [VGB07] propose a theoretical optimal reactive scheme. While reactive schemes are a low-complexity solution, they generally suffer from a high consumption variance.

Proactive energy-neutral schemes include estimations of future availability of energy in their control loop. Specifically for solar energy, many estimators exist. All of these leverage the observed history of harvested energy [KHZS07, RPBASR09, CPS12], while some additionally utilize external weather forecasts [SGIS10, SSIS11]. Proactive schemes typically perform better than reactive ones, though this highly depends on their future energy estimate accuracy [ST20].

However, solely focusing on maximizing the long-term utility easily makes a system prone to failure. Consider a proactive scheme, where an optimistic prediction error can lead to excess energy use, ultimately depleting the rechargeable storage and making the minimal energy consumption unsustainable. An adequately designed energy-neutral system thus uses adaptivity to also increase

its flexibility to the uncertain environment. Providing guarantees on behavior and functionality for such adaptive operation is a major focus of the first two chapters.

1.1.2 Timing Behavior

An embedded system consists of multiple tasks which together operate on a common hardware platform. Naturally, a system's inputs can not be completely predicted, and they determine the task's program flow and impact the timing behavior. But more importantly, a high degree of epistemic uncertainty is present in modern architectures. Most dynamic or virtualization features aimed to improve the average timing performance are considered sources of unpredictable and unbounded timing delays. The many hardware and software examples include direct memory access, caches, pipelining, dynamic storage allocation, probabilistic arbitration or communication protocols, operating system overheads, virtual storage, garbage collection, and so on [Hal04]. Though many applications exist where only the average performance is relevant, real-time systems need to fulfill both timing as well as functional requirements.

Real-time operation. For hard real-time systems, any missed deadline of a task is considered a system failure. In practice, even in many safety critical applications, a missed deadline can be tolerated if it happens infrequently. Such is the case for avionics systems, where industry standards [RE12] tolerate a failure if its probability of occurrence is less than a certain small value. These constraints are referred to as soft real-time. Finally, non real-time systems allow for a deadline miss to happen frequently, as their operation still retains usefulness even when timing constraints are violated.

Mixed-criticality operation. Mixed-criticality systems are a special class of real-time systems, where each task has a criticality level associated with it. Depending on the criticality level, a failure of a task due to a deadline miss or other reasons can have a more or less severe impact on the overall operation and safety of the

system. For an avionics application, the DO-178C [RE12] standard defines five criticality levels, 'A' to 'E', with 'A' being the highest, and assigns various failure tolerances to each level. Here, a failure of a task of criticality 'B' can have a negative impact on the overall safety of the aircraft, while a failure of a task of criticality 'D' may only slightly increase the aircraft crew's workload. Intuitively, an additional requirement exists when tasks of different criticality levels are present on a common platform: One must ensure that lower criticality tasks do not hinder the correct execution of higher criticality tasks.

Time predictability. In order to verify a real-time system's timing requirements, a characterization of the timing behavior of each task needs to be done. For this, a timing analysis aims to estimate or upper bound the worst-case execution time of a task. Many analysis techniques exist [DCG19b]. Some are based on measurements, where a task is run either on a simulator or the actual hardware, while static approaches use formal hardware and software models to draw conclusions. When a worst-case execution time value is obtained, time-predictability is defined as a measure of the associated pessimism [TW04].

Specially designed time-predictable hardware and software is necessary in cases when pessimism of worst-case execution time estimates would otherwise be too high. A recent industry trend of using multi-core architectures enhanced interest in such an approach. For instance, a design flow for mapping multiple applications on a heterogeneous embedded platform exists [SBR+12]. Furthermore, it is demonstrated that reasonable bounds on inter-task interference can be made even for highly utilized systems, using temporal partitioning [TGTT17] or careful memory allocation [TGBT17].

In this thesis, we assume systems where appropriate design choices reduce the execution time uncertainties to a reasonable level. We propose novel scheduling techniques that balance performance and guaranteed operation in this context.

Analogy with energy harvesting. Even though our two example environments seem different, they have many parallels. In both

cases, the system's ability to operate is variable, and a failure occurs either due to a lack of available energy, or due to tasks consuming excess resources. Following this analogy, minimal service requirements present in energy-neutral operation correspond to hard real-time requirements of important tasks in mixed-criticality applications, as both need to be guaranteed despite the present non-determinism. Furthermore, additional service is provided when more energy is available, in the same way as additional lower-criticality tasks are executed when more computational resources are available. Similarities between these two domains have inspired us to share insights and results from one use case to the other, which shall be illustrated throughout this thesis.

1.2 Adaptability Principles

Now that we commented on two examples of uncertainties an embedded system encounters, let us expand on the principle of adaptive operation. Adaptive operation is an alternative to over-provisioning resources, which we saw as not always feasible. In essence, an adaptive system will change its functional behavior online as a response to the current or perception of the future state of its environment. The adaptation strategy is decided on precisely and deterministically offline. If a system adapts its operation solely as a response to the state it is in, we say that adaption is done reactively. Otherwise, if adaption is done by also taking into account an estimated future state of the system, this is termed proactive adaption.

The set of all possible states an adaptive system can be in is called the system's *modes of operation*. It can be discrete or continuous, implying a finite or infinite number of modes, respectively. Depending on the mode of operation, individual tasks may execute normally, with some form of degradation, or they could be completely dropped. When a task is executed with degradation, it could mean that multiple implementations of the task with different functionalities are implemented, but also it could be that the task's activation times and deadlines, or even the used hardware and

its properties, are adjusted. Additionally, a task can have certain features intended to be executed only under favorable conditions, which is referred to as surplus functionality.

Let us focus on mixed-criticality systems first. To realize timing constraints for the most important tasks, a system may adapt by dropping tasks of lower criticality when it is necessary to provide additional computational resources to the more important ones. Depending on the system model and scheduling scheme, many mechanisms for this adaption exist, notably [Ves07, BBD11b] for fixed-priority preemptive scheduling, [BBD+11a] for dynamic scheduling schemes, and see also Chapter 4 for a comprehensive overview. Furthermore, degraded operation is sometimes suggested as an alternative to completely dropping tasks in mixed-criticality systems [GSHT13, HGST14]. Generally, the system can choose between as many modes of operation as there are criticality levels, so these are often called criticality modes. Finally, we note that criticality modes are chosen reactively as a response to the observed execution time of released tasks. A proactive approach is not possible in this scenario, as the timing behavior of tasks can not be forecasted.

Because of the domains they are currently found in, mixed-criticality systems are seldomly coupled with energy harvesting. Nevertheless, several works explore the principle that such a system changes its criticality mode in response to energy scarcity situations [AKTM16, SKT17, RMF19].

With energy-neutral operation, having many modes that enable multiple levels of functionality is common, and the system can switch between them both proactively and reactively. Two special cases are noteworthy. On the one hand, a specially constructed minimal 'emergency' mode of operation provides only the most crucial functionality. On the other hand, certain sporadic resource-heavy tasks such as updating statistical models or configuring machine learning parameters are designed to only execute in an energy-surplus mode of operation.

Because of the many adaptivity options available, and the coarse time-scale common in energy-neutral operation, it makes sense to abstract away from individual tasks and combine them into one value, the target energy consumption. This way, the system

can adjust its operation to any target consumption value between the minimum and maximum. Many works dealing with energy harvesting embedded systems model the system's adaptability this way [KHZS07, BKT15, VGB07].

Finally, let us mention the use of backup batteries in energy harvesting systems. In energy harvesting scenarios that have unlimited non-determinism, and may feature prolonged periods without any harvestable energy, they are used to guarantee a minimal service level. Examples of systems with a backup battery operating in especially unpredictable environments can be found in a busy seaport [VFPV16] or a low illumination indoor environment [VPN+16]. In principle, a system with a backup battery will strive to operate under energy-neutral constraints, but it will adapt to use the backup power source when such operation is not possible. Practically speaking, this is a valuable addition that adds another layer of security against system failure. It is thus unsurprising that many hardware solutions for energy management with a backup battery exist [Tex19a, JVT11].

1.3 Contributions and Outline

Throughout this thesis, we analyze an embedded system's adaption to its environment and strive to provide certain performance guarantees. We deal with two types of guarantees, worst-case and stochastic. Regardless of the scenario, we face two challenges that complicate providing such guarantees:

- There is non-determinism present in the environment.
- In response, the system adapts its behavior online.

Worst-case guarantees are arguably more common in the state-of-the-art. In the mixed-criticality domain, we review schedulability analyses that precisely verify whether tasks keep with their hard real-time requirements, even when each of them executes with their worst-case execution time. Such a mathematically constructed worst situation is rarely available for energy harvesting scenarios, though. Thus, our worst-case analysis of a system in this regard

includes first assessing the harvesting source based on available historical data, either from the same or similar environments, and then quantifying the observed worst-case performance. This quantification is done both for abstract and non-ideal system models, and relies on extensive simulations. Additional verification is done by recreating exact harvesting conditions in a special test environment [Sig20].

Stochastic guarantees are harder to obtain but provide more insightful results. We have found that modeling adaptive embedded systems as stochastic Markov processes provides an appropriate mathematical apparatus for calculating probability-of-failure metrics. When doing a probabilistic timing analysis, the amount of pending execution corresponds to states in the Markov chain. In this setting, appropriate random variables exist that describe upper bounds of execution times of tasks. They are known as probabilistic worst-case execution times and can be obtained using an array of existing techniques [DCG19b]. Most importantly, we demonstrate that such a timing analysis provides safe upper-bounds on deadline miss probabilities of tasks, as well as probabilities that tasks are degraded or dropped due to run-time decisions of the scheduler.

For energy harvesting systems, we create appropriate models of the harvesting source using historical data. In our solar energy example, random variables model the amount of energy harvested each week of the year. By assigning a Markov process state to each charge level of the rechargeable storage, we are able to accurately model complex energy management schemes. This enables us to derive, among other metrics, the probability of system failure.

Outline. In Chapter 2, the focus is on energy harvesting systems, which proactively adapt their target energy consumption in response to the locally stored energy level, and the predicted availability of energy. The most interesting case we explore is when relatively precise estimators of harvestable energy exist, but we also cover cases when the environment is especially hard to predict, as well as a theoretical case when the exact future is known in advance. We especially use solar energy harvesting as our use case. Variability of execution times is not considered in this chapter, instead one can assume that tasks consume the most energy with

their worst-case execution times.

For such a system the goal is to prolong the lifetime under energy-neutral operation constraints, while maximizing the long-term utility. When constraints on energy neutrality can not be met, we consider how a backup battery should be used to maximize the system's lifetime. We first formally model this problem, and deal with the unpractical but theoretically interesting case when the future of harvestable energy is known. For this case, an optimal control algorithm is presented. Next, we introduce a practical proactive adaption scheme, which we additionally supplement with our precise and low-complexity future energy estimators. We then use trace-based simulations to find the worst-case performance of this scheme, and a real-world system implementation finally verifies the accuracy of our results.

Chapter 3 continues by providing a stochastic analysis of energy harvesting systems. Apart from the randomness introduced from energy harvesting, in this chapter we also take into account the variability in energy consumption created by uncertain execution times of tasks. This analysis provides such metrics as the probability of failure, or the probability that the system consumes a certain energy level. Additionally, we solve the dimensioning problem, where feasible energy management subsystem design points are found for a specified application and failure tolerance. The same solar energy harvesting example as before was used for extensive simulations.

Finally, Chapter 4 deals with providing real-time constraints for mixed-criticality systems, where the uncertainty solely comes from the run times. The adaptability, in this case, is reactive and happens when it seems necessary to provide additional computational resources to tasks of higher criticality. Using our probabilistic mixed-criticality analysis, we analyze our proposed adaptive scheduling scheme and provide the probability of deadline miss per hour, and the probability that a task is executed in degraded mode.

Table 1.1: Overview of sources of non-determinism, adaptability principles, and analysis methods found in each chapter of this thesis

	Ch. 2	Ch. 3	Ch. 4
★ *Source of Non-determinism*			
Execution times of tasks is			
modeled in a worst-case manner	✓	✓	✓[1]
modeled with random variables		✓	✓
Harvested energy's availability is			
especially hard to predict	✓		
known in advance[2]	✓	✓[1]	
modeled with random variables		✓	
★ *Adaptivity*			
Modes of operation are			
discrete		✓	✓
continuous	✓	✓	
Their operation is adjusted			
reactively	✓[1]	✓[1]	✓
proactively	✓	✓	
The use of backup resources is considered	✓		
★ *Analysis methods and provided guarantees*			
Worst-case analysis and guarantees	✓		✓[1]
Markov analysis with probabilistic guarantees		✓	✓

[1] For comparison only.
[2] Theoretical scenario, unobtainable in practice.

2

Worst-case Analysis of Energy Harvesting Systems

A prime example of embedded systems operating in a variable environment is when an energy harvester powers them. Energy harvesting has been extensively used to allow for long-term and unattended operation of nodes in large-scale distributed systems. Often in such scenarios, a system's mode of operation is adapted on-line to the temporary availability of energy, either proactively or reactively, which enables a reduction of the rechargeable energy storage of the harvesting system. However, the smaller the relative rechargeable energy storage, the higher is the sensitivity of the system's operation to short-term non-deterministic changes of the harvested power.

By doing a formal study, we first analyze the case when uncertainties in energy harvesting are removed. This enables us to design an optimal but practically unrealizable energy consumption function. Optimal in this sense means that the long-term utility of the system is maximized, energy-neutral operation is realized, and the minimal energy consumed in a single time step is the largest among all feasible consumption functions. Furthermore, this minimal energy consumption is found to be always larger than zero,

except in the corner case when absolutely no energy was harvested before. A practical finite horizon control solution is presented next. Utilizing a reasonable estimate of future energy availability, it approximates the optimal consumption well.

Although the optimal minimal energy consumption is guaranteed to be non-zero, depending on the harvesting environment it may become arbitrarily small. When this is not acceptable, because a certain consumption level is required, reliable and predictable operation under any circumstances can be achieved with an additional backup battery. First commercial products that allow for an efficient energy exchange between an energy harvester, backup battery, and energy consumer are available. But, until now, there were no energy management algorithms that optimize the long-term utility while maximizing the lifetime of the system in terms of its backup battery. We present both an optimal and finite horizon control solution with this goal in mind.

Both with and without a backup battery, we implement our energy management strategy on resource-constrained hardware. Multiple simulations, as well as experiments on indoor and outdoor solar data traces, show the accuracy of our model, the run-time overhead, and the efficiency of our approach.

2.1 Introduction

Energy harvesting is a key technology to provide long-term, autonomous and sustainable energy to embedded devices in large-scale distributed systems. A large body of results shows the potential of extracting energy from light, temperature differences, acceleration, vibration, and electromagnetic waves, see for example the review by Shaikh and Zeadally [SZ15]. Motivated by cost, size, safety and environmental reasons, there is a recent trend to substantially reduce the necessary local rechargeable energy storage, often in the context of transient and intermittent computing [LBC+17]. This approach also allows to replace rechargeable batteries with super-capacitors that are accompanied by advantages such as increased safety, a high number of charge-discharge cycles

and high retrieval efficiency.

The high variation inherent in energy harvesting sources, such as solar energy, creates a challenge. The smaller the relative maximal energy storage capacity of the node, the higher is the sensitivity of its operation to short-term non-deterministic changes of the harvested power or the power demand. Concepts to decrease the sensitivity and to deal with frequently interrupted processing have been developed [GSM+16, RSF11], including extreme approaches where the harvesting source is directly coupled to the node [BWM+14]. In an ideal scenario, a system adapts its consumption and achieves perpetual operation, commonly termed energy-neutral operation (ENO) [KHZS07], if the consumed energy never exceeds the harvested one.

Nevertheless, there are many applications where correct operation requires the unconditional availability of some amount of energy within a given time interval. Examples are automatic control, surveillance, early warning, or safety-critical sensing applications. In addition, low-power multi-hop networks typically need periodic refresh operations for synchronization. Unless there is a reliable correlation between harvested energy on the one hand and events to be sensed or actions to be performed on the other, the usefulness of intermittent or transient approaches to energy provisioning in critical applications is limited.

One approach to providing performance guarantees to such systems is to reduce the unpredictability of the energy harvesting source, and adapt the system's operation accordingly. In solar energy harvesting systems, some of the variations in input energy are caused by the diurnal solar cycle and the yearly seasons, and these phenomena are precisely known. Other variations are non-deterministic, they are caused by cloud cover or other weather conditions, and can therefore only be estimated. Several existing works utilize harvestable energy forecasts of various accuracy, tightly coupled with power management schemes, in an attempt to achieve continuous energy-neutral operation [KHZS07, BSBT14a, MBTB07]. However, providing minimum performance guarantees is largely unexplored and often difficult, so in this chapter we investigate *how to control the operation of a harvesting system's consumption, such that long-term utility is optimized, the minimum*

performance is increased, and continuous operation is guaranteed. In this regard, we build upon the seminal work of Buchli et al. [BKT15].

Still, the best way to provide hard guarantees in reliability and predictability of node operation can be obtained with an *additional* backup battery, a primary cell for example. This solution not only has been suggested by academia, e.g., [JAD19, VFPV16, JVT11], but industry is also providing first commercial products that allow for an efficient energy exchange between energy harvester, backup battery and energy consumer [Tex19a]. Such a design not only allows an otherwise unreliable system to operate without interruption, but also offers additional benefits such as guaranteeing minimal service, increasing the efficiency and speed of cold-start from an empty energy storage, retaining state across restarts, or operating without interruption for years or decades.

One of the most important aspects in energy harvesting nodes is the energy management system. It is responsible for deciding on the energy flow between the energy source, the rechargeable energy storage unit and the energy consumer. It also controls the configuration and operation of the consumer such that an overall system utility is maximized. Examples of varying operation of the consumer are adapting the sensing rate, the communication rate, the kind of data processing algorithms used or even going to a low energy "energy scarcity" mode. Surprisingly, energy management and control has been extensively covered in the general sense of energy harvesting sensing systems, see [KHZS07, VGB07, SSIS11, MTBB09], but no results are available in the context of an additional backup battery.

The present paper fills this gap by providing models and methods to answers the following question: *how to control the operation of a harvesting system's backup battery, such that a minimal amount of energy consumption is guaranteed, its lifetime is maximized and its long-term utility is optimized?* As a result, the described concepts and hardware for energy harvesting nodes with a backup battery can now be combined with control algorithms to make optimal use of the available resources such as the backup battery, the harvested energy and the energy storage capacity. In summary, the paper contains the following results:

1. We formalize a model describing a complete harvesting system with an optional backup battery, which adapts to energy availability by choosing online one out of continuous modes of operation. Additionally, there is a the possibility to incorporate all relevant inefficiencies of practical implementations.
2. We explain an impractical optimal control algorithm. For a harvesting system without a backup battery, dubbed the 'original' case, it optimizes the minimum use energy and long-term utility. When a backup battery is present, the 'backup battery' case, our novel extension optimizes the long-term node utility while guaranteeing a given minimal use energy regardless of the harvesting conditions, and maximizing the life-time of the system.
3. We present a practical finite horizon control (FHC) approximation of the optimal control algorithm, based on model predictive control, that copes with non-perfect estimations of future energy. In addition, efficient implementations of FHC are referenced, including one based on Look-up Tables (LUT) witch is deployable to even the most resource-constrained systems.
4. We improve upon state-of-the-art energy prediction schemes, which are necessary for proactive energy management strategies like FHC, by leveraging the extra-terrestrial solar model. Our schemes improve upon state of the art prediction accuracies, which leads to an overall improvement in FHC performance.
5. We present an implementation of the overall hardware and software, and a characterization of its essential properties. Extensive emulation and experimental results on both indoor and outdoor solar data that shows the applicability of our approach, as well as the accuracy of our model.

This chapter is organized as follows: after commenting on related work in the next section, we describe the abstract system model in Section 2.3. Based on it, we provide an optimal control algorithm in Section 2.4 and embed it into a model predictive control scheme in 2.5, both for systems with and without a backup battery. Then, in Section 2.6, we present state-of-the-art solar

energy prediction schemes, as well as our own ones based on the extra-terrestrial solar model. All estimators are compared in terms of prediction accuracy and computational complexity as well. Following this, in Section 2.7 we describe how to incorporate non-ideal hardware properties into the formal concepts described before. Finally, Section 2.8 presents our sample implementation of an energy harvesting system with a backup battery, which is accurately modeled in Section 2.10, while Section 2.9 contains extensive experimental results.

2.2 Related Work

Energy systems. Superficially, there is a close relation to energy systems incorporating renewable sources like solar and wind. Whereas one can draw analogies in terms of energy generation, energy storage (water tanks, car batteries), application control (demand side management), there are essential differences in terms of models and methods due to the incorporation of the energy grid, sufficient energy for implementing complex control strategies, energy conversion and transmission, consumer behavior and cost functions, see [ZS14, AOS+12].

Energy-neutral operation. In order for a system to operate solely on harvested energy, state-of-the-art work focuses primarily on defining dynamic power management schemes, as well as on constructing and dimensioning the components of an energy management system. A first work on dynamic power management was written by Kansal et al. [KHZS07], where an analytical model featuring energy-neutral operation (ENO) constraints was developed with the goal of maximizing performance. The problem of computing the dynamic power management profile efficiently was explored by Moser et al. [MTBB09]. Examples of uninterrupted environmental monitoring nodes operating over multiple years are given by Buchli et al. for a high-mountain environment [BSBT14a, BSBT14b], by Corke et al. for a remote field [CVS+07], and by Taneja et al. [TJC08] for a deep forest scenario. Even though these

works feature detailed design principles and power management procedures, uninterrupted operation is not guaranteed under a non-deterministic harvesting environment. In addition, none of these results considers the availability of a backup battery.

ENO and solar energy estimation. Many systems designed for ENO feature estimations of future harvested energy in their control loop. A plethora of solar energy prediction schemes exist. Included among them are EWMA by Kansal et al. [KHZS07], WCMA by Recas Piorno et al. [RPBASR09], and ProEnergy by Cammarano et al. [CPS12]. In general, all of these schemes leverage the history of harvested energy to predict the energy which will be harvested in the future, see also Section 2.6.1 for details. Additional schemes sometimes use weather forecasts to improve prediction accuracy, see for example Sharma et al. [SGIS10, SSIS11], although these are not applicable in the general case due infrastructure requirements. In the common use-case when the prediction horizon is one day, we improve upon state-of-the-art schemes by deterministically predicting variation in solar energy caused by the diurnal solar cycle and the yearly seasonal variation, which is done using the extraterrestrial solar model. To the best of our knowledge, this approach has been used by two research works. Buchli et al. [BSBT14a] use an extraterrestrial model to create a long-term solar energy predictor, with one week prediction intervals and a one year horizon. Bao et al. [BWL+14] use the extraterrestrial model along with externally acquired cloud cover information to predict energy.

Reactive ENO. Some schemes for ENO do not feature estimates of future states, but are instead purely reactive. Notably, Vigorito et al. [VGB07] propose an optimal reactive scheme under ENO constraints (EnoMax), which dynamically adapts the system's consumption based on the state of charge of the rechargeable storage. While this is a low-complexity solution, EnoMax and similar schemes suffer from high consumption variance.

Transient systems. Transiently powered computing systems are designed with little or no energy buffers, thus working only when

sufficient energy is available, while retaining their state otherwise [MAH17]. Generic energy management units for these systems exist [GSM+16]. Examples of use-cases include a passive camera powered by RFID by Naderiparizi et al. [NPK+15], or an AC power meter powered by the current it is measuring by DeBruin et al. [DCD13]. Due to its low overhead, state retention is often realized in FRAM technology [BWM+14, JRR14]. Furthermore, transient systems feature complex intermittent programming models, i.e., [RSF11, BDW+16]. However, these systems may remain offline for extended periods of time, which is only acceptable for a reduced set of application scenarios.

Existing hardware implementations with backup energy. A plethora of nodes featuring photovoltaic energy harvesting as well as backup batteries exist. For example, Visconti et al. [VFPV16] present a wireless sensor network (WSN) solution for tracking of goods in a commercial seaport. In Jackson et al. [JAD19], a general purpose WSN has a large lithium ion rechargeable battery, while a primary battery is used to ensure uninterrupted operation. A WSN optimized for harvesting energy in low illumination environments, by Vračar et al. [VPN+16], features a similar approach. Hardware components for energy management systems with backup batteries exist, from the bq25505 energy management chip by Texas Instruments [Tex19a] featuring a simple multiplex, to the smart platform by Jessen et al. [JVT11] that enables complex energy management functions. Still, there are no results available that allow to control the operation of the node and the energy management system such that the backup battery is used as little as possible while maximizing the long-term utility of the node.

2.3 System Model

We start by introducing an abstract model of energy harvesting systems, which is the basis for energy control. Later in the chapter, the model will be refined in order to take into account non-ideal behavior of the energy management system. We use this model as a

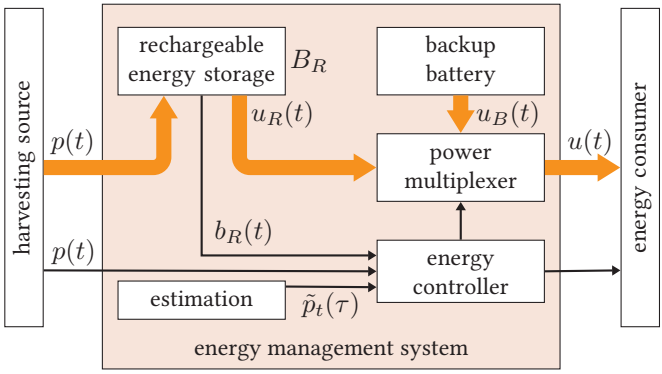


Figure 2.1: Overview of the energy management system model. The thick and thin arrows show the energy and information flow, respectively

generalization for both systems with and without a backup battery.

An abstract overview of the overall setup is shown in Figure 2.1. The energy management system has two types of energy reservoirs: a *rechargeable energy storage* and a *backup battery*. The rechargeable energy storage is replenished by an energy harvester, while the backup battery is a primary battery. For our initial abstract model, we suppose the reservoirs are loss-free, but extensions are described in Section 2.7. The energy management system provides the energy consumer with information about the target energy consumption. As a consequence, the consumer adapts its operation to match this target value, for example by changing its sensing rate, the processing speed, the communication rate, or the like.

Formal model. We use a discrete time system model, where each time instance t is a positive integer $t \in \mathbb{Z}^{0+}$. For now, we assume this model is given, while a comparison between continuous and discrete time system models for energy harvesting systems is given in the following chapter, in Section 3.2. When a value relates to time instance t, we mean it applies for time interval $[t, t+1)$.

The following definitions and relations, or their variations, are commonly found in related work, e.g., [KHZS07]. The *harvested*

energy during time instance t is denoted as $p(t)$. This value depends on the energy harvester and the environment. The consumed energy during time instance t is denoted as $u(t)$, with u being the *use function*. The energy management system choses a mode of operation for the next time instance, and a target energy consumption is associated with it. When different from the actual use energy, this target use function is denoted $\tilde{u}$. The consumed energy $u(t)$ equals the energy taken from the rechargeable energy storage and from the backup battery,

$$u(t) = u_R(t) + u_B(t) \tag{2.1}$$

The energy management system also decides which portion is taken from which energy reservoir. The *base consumption* $u_{\mathsf{BASE}} \geq 0$ defines a lower bound on the consumed energy with $u(t) \geq u_{\mathsf{BASE}}$, which applies for every time instance $\forall t \in \mathbb{Z}^{0+}$.

The rechargeable energy storage has $b_R(t)$ energy stored at the beginning of time instance t, and a maximum capacity B_R. For our initial abstract model, we suppose it is loss-free. For the rechargeable energy storage, the following holds:

$$b_R(t+1) = \min\{b_R(t) + p(t) - u_R(t), B_R\} \tag{2.2}$$

The harvested energy that is discarded because the rechargeable energy storage is full is called “wasted energy" for short. The *use of the backup battery* in time interval $[t_1, t_2)$ is denoted as

$$E(t_1, t_2) = \sum_{t_1 \leq t < t_2} u_B(t) \tag{2.3}$$

A utility function can be any strictly concave function $\mu : \mathbb{R}_{\geq 0} \to \mathbb{R}_{>0}$. This concavity means that it models the diminishing return of investment if the energy consumption of a node grows. One may also define a maximal consumption, over which no additional utility is realized. The total utility in a time interval $[t_1, t_2)$ is the sum of the utility during this time:

$$U(t_1, t_2) = \sum_{t_1 \leq \tau < t_2} \mu(u(\tau)) \tag{2.4}$$

Note, a strictly concave function μ satisfies

$$\mu(\alpha x + (1-\alpha)y) > \alpha\mu(x) + (1-\alpha)\mu(y)$$

for any $0 \leq \alpha \leq 1$ and $x, y \in \mathbb{R}_{\geq 0}$.

Harvested energy estimation. Let us define the harvested energy estimate function $\hat{p}_t(\tau)$. It is a part of the energy management system, and it is a function of two parameters: the time of prediction t, and the time for which we predict τ. The time of prediction is a parameter because, in a realistic setting, the estimate is updated as time passes and new harvested energy values are observed. We assume that the estimate is defined for all $\tau \in [t, t + T_H)$, where $T_H > 0$ is the *prediction horizon*, which specifis for how far in the future a harvested energy estimate exists.

Optimization problems. We deal with two optimization problems, and each is solved in two steps. The two optimization problems refer to systems with and without a backup battery, respectively, and these will be known as the 'original' and 'backup battery' problems. The original problem is solved by Buchli et al. [BKT15], and is given here for reference as our solution to the backup battery problem extends upon it.

For both problems, at first we assume that we know the harvested energy in the future. This forms the basis for the second step, model predictive energy control, which is the case when the future harvested energy is only estimated with a function $\tilde{p}_t(\tau)$.

Optimization Problem 2.1 *Original Problem*: ([BKT15], rephrased). Given no backup battery and no minimal required energy consumption $u_{\mathsf{BASE}} = 0$. Given the initial and the final rechargeable energy storage states $b_R(0)$ and $b_R(T)$, and the harvested energy $p(t)$ for all $t \in [0, T)$. Given the following constraints for the energy management system for all $t \in [0, T)$:

$$\begin{gathered} u_B(t) = 0 \\ u_R(t) \geq 0 \\ u(t) = u_R(t) + u_B(t) \geq u_{\mathsf{BASE}} = 0 \\ u_R(t) \leq b_R(t) + p(t) \\ b_R(t+1) = \min\{b_R(t) + p(t) - u_R(t), B_R\} \geq 0 \end{gathered} \tag{2.5}$$

Under the above constraints, an optimal energy controller determines a rechargeable use function $u_R^*(t)$ such that the total utility $U^*(0, T)$ is maximized.

Optimization Problem 2.2 *Backup Battery Problem*: Given the minimal required energy consumption u_{BASE}, the initial and the final rechargeable energy storage states $b_R(0)$ and $b_R(T)$, and the harvested energy $p(t)$ for all $t \in [0, T)$. Given the following constraints for the energy management system for all $t \in [0, T)$:

$$\begin{aligned} u_B(t) &\geq 0 \\ u_R(t) &\geq 0 \\ u(t) = u_R(t) + u_B(t) &\geq u_{\mathsf{BASE}} \\ u_R(t) &\leq b_R(t) + p(t) \\ b_R(t+1) = \min\{b_R(t) + p(t) - u_R(t), B_R\} &\geq 0 \end{aligned} \tag{2.6}$$

Under the above constraints, an optimal energy controller determines a rechargeable use function $u_R^*(t)$ and backup use function $u_B^*(t)$ that satisfy the following criteria:

O1: The total use of the backup energy $E^*(0, T)$ is the smallest among all possible use functions, and therefore, the lifetime of the node is maximized.

O2: Among the use functions that satisfy the above criterion, the use function $u^*(t) = u_R^*(t) + u_B^*(t)$ has the maximal total utility $U^*(0, T)$.

As a shorthand notation, we denote all use functions that satisfy constraints as *feasible*. Furthermore, if $u(t) < u_{\mathsf{BASE}}$, we say the system is in a *failure state* at time t due to insufficient energy availability. Likewise, if $b_R(t) + p(t) - u_R(t) < 0$, the system is in a *failure state* at time t due to depletion of the rechargeable energy storage.

Solutions to these optimization problems are presented in Section 2.4. Based on finite horizon control, solutions to the problem with unknown future energy are given in Section 2.5.

Before we continue to solve our optimization problems, we present the following lemma which reformulates the constraints, by stating that no solution where wasted energy exists can be optimal.

Lemma 2.1: *A use function $u_R(t)$ that satisfies $b_R(t') + p(t') - u_R(t') > B_R$ for some $t' \in [0, T)$ can not be optimal.*

Proof: If we would replace $u_R(t')$ by $u'_R(t') = b_R(t') + p(t') - B_R$ then the resulting total utility would be larger, as $u'_R(t') > u_R(t')$ and no system state changed, i.e., all stored energies are the same.□

The last equation of both constraints (2.5) and (2.6) can thus be restated as follows:

$$0 \leq b_R(t+1) = b_R(t) + p(t) - u_R(t) \leq B_R \tag{2.7}$$

2.4 Optimal Energy Use

For the theoretical case when the future harvested energy is perfectly estimated, we solve two optimization problems. This provides a first step towards understanding practical use functions.

At every time interval, the energy controller decides which mode of operation to go into. This consumes more or less energy, but also increases or decreases the use function for that time interval. Bad decisions may cause rechargeable storage depletion, wasted energy, or a sub-optimal use function. Taking this complexity into account, we first investigate certain properties of the optimal solutions.

We then present algorithms to compute them, when the harvesting energy $p(t)$ is known. Formally speaking, we restate a unique optimal use function $u^*_R(t)$ from [BKT15], which is the solution to Optimization problem 2.1. Next we present a unique optimal use function $u^*(t)$ which is the solution to Optimization problem 2.2, meaning it has the minimal total backup energy usage (O1) and among all functions fulfilling this condition, it has the largest possible utility (O2).

2.4.1 Properties

Let us start with showing the properties and the uniqueness of the original optimization problem's solution. Later, we prove how solutions to the two problems are connected.

The original problem. The main property of the optimal solution is formulated by the following theorem.

Theorem 2.1: *(Theorem 1 from [BKT15]). Given a use function $u_R^*(t)$ that satisfies constraints (2.5). If the following relations hold:*

$$\begin{aligned} u_R^*(s-1) < u_R^*(s) \;&\Rightarrow\; b_R^*(s) = 0 \\ u_R^*(t-1) > u_R^*(t) \;&\Rightarrow\; b_R^*(t) = B_R \end{aligned} \tag{2.8}$$

Then $u_R^*(t)$ maximizes the total utility $U^*(0,T)$, it is unique, and it maximizes the minimal used energy.

Proof: This is a direct consequence of Lemmas 2.2 and 2.3. □

This theorem provides a necessary and sufficient condition for u_R^*. It is the consequence of two lemmas, the first of which says that the optimal energy is constant as long as the rechargeable storage is neither full nor empty. Otherwise, the optimal use function shrinks and grows, respectively.

Lemma 2.2: *(Lemma 1 from [BKT15]). Any optimal use function $u_R^*(t)$ satisfies:*

$$\begin{aligned} &\forall s,\tau,t \,|\, \tau \in [s,t] :\; 0 < b_R^*(\tau) < B_R \;\Rightarrow \\ &\qquad \tau \in [s-1,t] :\; u_R^*(\tau) = u_R^*(t) \end{aligned} \tag{2.9}$$

$$u_R^*(s-1) < u_R^*(s) \;\Rightarrow\; b_R^*(s) = 0 \tag{2.10}$$

$$u_R^*(t-1) > u_R^*(t) \;\Rightarrow\; b_R^*(t) = B_R \tag{2.11}$$

The next lemma then states that the optimal use function $u_R^*(t)$ is unique.

Lemma 2.3: *(Lemma 2 from [BKT15]). If there exists a use function $u_R^*(t)$ that satisfies the necessary optimality conditions (2.9) to (2.11) of Lemma 2.2, and that does not lead to a failure, then it is unique.*

The backup battery problem. Now we continue and demonstrate how to take a solution $u_R^*(t)$ to Optimization problem 2.1, and obtain from it a solution $u^*(t)$ to Optimization problem 2.2 for a system with a backup battery and a specified minimal use function u_{BASE}. This transformation of one solution to the other is practical, and can be expressed as the following theorem.

Theorem 2.2: *Given an optimal solution $u_R^*(t)$ to Optimization problem 2.1. Then $u^*(t)$:*

$$\begin{aligned} u^*(t) &= u_R^*(t) + u_B^*(t) \\ u_B^*(t) &= \max\{0, u_{\mathsf{BASE}} - u_R^*(t)\} \end{aligned} \tag{2.12}$$

is a solution to Optimization problem 2.2, and is unique.

Proof: This is a direct consequence of Lemmas 2.4 and 2.5, and the fact that $u_R^*(t)$ is unique (Theorem 2.1). □

Note that the uniqueness clause applies to $u^*(t)$, and not necessarily to $u_R^*(t)$ and $u_B^*(t)$. The first lemma shows that the solution in (2.12) minimizes the use of the backup battery and therefore, maximizes the lifetime of the node.

Lemma 2.4: *Given an optimal solution $u_R^*(t)$ to Optimization problem 2.1. Then $u^*(t)$, given by (2.12), has the minimal total backup usage $E^*(0,T)$ among all use functions that satisfy (2.6).*

Proof: $u_B^*(t)$ and $u_R^*(t)$ are solutions to Optimization problem 2.2, with base consumption u_{BASE}, as

$$\begin{aligned} u^*(t) = u_R^*(t) + u_B^*(t) = \\ u_R^*(t) + \max\{0, u_{\mathsf{BASE}} - u_R^*(t)\} = \\ \max\{u_R^*(t), u_{\mathsf{BASE}}\} \geq u_{\mathsf{BASE}} \end{aligned}$$

Therefore, we need to prove that $u_B^*(t)$ has the minimal total backup usage $E^*(0,T)$ (criteria O1).

Given some u_{BASE}, we consider intervals where $u_R^*(t) < u_{\mathsf{BASE}}$ and therefore, the backup battery is used,

$$u_R^*(t) < u_{\mathsf{BASE}} \;\Rightarrow\; u_B^*(t) = u_{\mathsf{BASE}} - u_R^*(t) > 0$$

If no intervals like these exist, then the total backup usage is trivially minimal, $E^*(0,T) = 0$. Otherwise, we will show that for any of these intervals, the total spent rechargeable energy is maximal, and therefore the total used energy from the backup storage is minimal.

Let us take one of these intervals $[t_1, t_2]$. If $t_1 = 0$, then the energy in the rechargeable energy storage at time t_1 is given $b_R(0)$.

For any other value of t_1 we have $u_R^*(t_1 - 1) \geq u_{\mathsf{BASE}}$, and since $u_R^*(t_1) < u_{\mathsf{BASE}}$, properties from Theorem 2.1 give us $b_R^*(t_1) = B_R$. In other words, whatever time $t_1 > 0$ may be, the initial stored energy of the rechargeable energy storage for interval $[t_1, t_2]$ is maximal among all use functions.

For the end of the interval, the situation is similar: whatever time $t_2 < T$ may be, the final stored energy of the rechargeable energy storage for interval $[t_1, t_2]$ is minimal among all use functions. If $t_2 = T$, then the final stored energy $b_R(T)$ is given, and for other values of t_2 it is zero. As before, this is due to $u_R^*(t_2 + 1) \geq u_{\mathsf{BASE}}$, $u_R^*(t_2) < u_{\mathsf{BASE}}$ and properties from Theorem 2.1.

From (2.7) we know that $b_R^*(t + 1) = b_R^*(t) + p(t) - u_R^*(t)$ and therefore

$$\sum_{t_1 \leq t \leq t_2} u_R^*(t) = b_R(t_1) + \sum_{t_1 \leq t \leq t_2} p(t) - b_R(t_2)$$

When $t_1 > 0$ and $t_2 < T$, we can substitute $b_R(t_1) = B_R$ and $b_R(t_2) = 0$:

$$\sum_{t_1 \leq t \leq t_2} u_R^*(t) = B_R + \sum_{t_1 \leq t \leq t_2} p(t)$$

In other words, the used rechargeable energy equals $b_R(t_1) - b_R(t_2)$ (which is, except for corner cases, the capacity B_R), plus the harvested energy in $[t_1, t_2]$. As it is not possible to use more energy in any time interval than what can be stored in the energy buffer plus the harvested energy in this time interval, the term $\sum_{t_1 \leq t \leq t_2} u_R^*(t)$ is maximal. No other use function $u_R(t)$ can have a larger sum in $[t_1, t_2]$. Therefore, the used backup energy in $[t_1, t_2]$ is minimal:

$$E^*(t_1, t_2 + 1) = \sum_{t_1 \leq t \leq t_2} u_B^*(t) = \sum_{t_1 \leq t \leq t_2} (u^*(t) - u_R^*(t)) =$$
$$(t_2 - t_1 + 1) \cdot u_{\mathsf{BASE}} - (b_R(t_1) - b_R(t_2)) - \sum_{t_1 \leq t \leq t_2} p(t) \qquad \square$$

The next lemma deals with maximization of the total utility $U^*(0, T)$.

Lemma 2.5: *Given an optimal solution $u_R^*(t)$ to Optimization problem 2.1. Then $u^*(t)$, given by (2.12), has the maximum total utility $U^*(0,T)$ among all use functions that satisfy (2.6) and have the minimal total backup usage $E^*(0,T)$.*

Proof: We prove this lemma by contradiction. We assume that use function $u(t)$ with utility $U(0,T)$ is different from $u^*(t)$ as defined in the statement, but has the maximal total utility among all use functions optimal according to O1. By the end of this proof, we will show that this use function $u(t)$ can in fact be replaced by a use function $u'(t)$ with a strictly higher utility $U'(0,T)$ and same backup battery usage (O1), thus making the contradiction.

Due to the fact that $u(t)$ is different from $u^*(t)$, and as a consequence of properties of $u^*(t)$ shown in Theorem 2.1, we can claim the following: There exists at least one interval (t_1,t_2) during which the rechargeable energy storage is neither empty nor full, where the function $u(t)$ is not constant. As $u(t)$ is not constant during interval (t_1,t_2), there exists a maximum and a minimum $u(t)$ inside the interval, say $u(\tau_1)$ and $u(\tau_2)$, respectively, with $\tau_1 \neq \tau_2$.

We now define $u'(t)$ to be the same as $u(t)$, except at these two times τ_1 and τ_2. For these two instances, we define

$$u'(\tau_1) = u(\tau_1) - \delta\,, \quad u'(\tau_2) = u(\tau_2) + \delta$$

We can choose δ to be any small non-zero value, so that it does not violate the constrains on stored energy usage. More precisely, we focus just on changing the rechargeable component of $u(t)$ to create $u'(t)$, and here the maximum allowed variation is $\delta = \min_{t_1\leq\tau\leq t_2}\{|b_R(\tau)|, |B_R - b_R(\tau)|\} > 0$. This way, $\sum_{\tau=t_1-1}^{t_2} u(\tau) = \sum_{\tau=t_1-1}^{t_2} u'(\tau)$ and also, the stored energy functions will satisfy $b'_R(\tau) = b_R(\tau)$ for all $0 \leq \tau \leq t_1 - 1$ and $t_2 + 1 \leq \tau \leq T$.

We finally have to show that $u'(t)$ has a strictly higher utility $U'(0,T)$ than $u(t)$, and hence $u(t)$ is not optimal. We have

$$\begin{aligned} &U'(0,T) - U(0,T) = \\ &\quad \mu(u(\tau_1) - \delta) + \mu(u(\tau_2) + \delta) - \mu(u(\tau_1)) - \mu(u(\tau_2)) \end{aligned} \tag{2.13}$$

Using the condition that μ is strictly concave, it can be shown that there exists a constant α that leads to

$$\begin{aligned}\mu(u(\tau_1)-\delta) &= \mu(\alpha u(\tau_1)+(1-\alpha)u(\tau_2)) \\ &> \alpha\mu(u(\tau_1))+(1-\alpha)\mu(u(\tau_2)) \\ \mu(u(\tau_2)+\delta) &= \mu((1-\alpha)u(\tau_1)+\alpha u(\tau_2)) \\ &> (1-\alpha)\mu(u(\tau_1))+\alpha\mu(u(\tau_2))\end{aligned} \tag{2.14}$$

Applying (2.14) into (2.13) proves $U'(0,T)-U(0,T)>0$, which means that $u(t)$ could not have been optimal, thus $u^*(t)$ has the maximal total utility $U^*(0,T)$ (O2). □

This concludes our analysis of the two optimal solutions: their properties, and how one can be transformed into the other.

2.4.2 Computation

The solutions to Optimization problems 2.1 and 2.2 still need to be computed. We thus start by presenting an iterative algorithm which approximates to a desired precision the optimal solution $u_R(t)$ to the original problem, based on the harvested energy trace $p(t)$.

As far as the backup battery extension is concerned, there is only one main problem to be solved. *How can we efficiently implement (2.12) such that switching constraints as present in typical hardware implementations are taken into account?* Following (2.12), the overall use function for the energy consumer is given by $u^*(t) = \max\left(u_R^*(t), u_{\mathsf{BASE}}\right)$ and it is unique, see Theorem 2.2. Nevertheless, there is some flexibility in partitioning $u^*(t)$ into its backup battery $u_B^*(t)$ and rechargeable energy storage $u_R^*(t)$ components, which we demonstrate.

The original problem. An equivalence exists between the Optimization problem 2.1 and the shortest Euclidean path in simple polygons problem in computational geometry. This was demonstrated by Chen et al. [CSSJ11], and proven by defining a *harvesting polygon*, through which each path corresponds to a feasible use function $u_R(t)$ with no wasted energy. Most importantly, the shortest path corresponds to the optimal solution $u_R^*(t)$.

Guibas et al. [GHL+87] describe an algorithm of complexity $\mathcal{O}(n)$ to compte the shortest feasible path, where n is the number

Algorithm 2.1: Computing the optimal use function for the original problem, an iterative approach

```
1  procedure get optimal (p, b_R(0), b_R(T), T, B_R, ϵ)
2      for t in 0 < t ≤ T do
3          σ_l(t) ← Σ_{τ=1}^{t} p(τ − 1)
4          σ_u(t) ← σ_l(t) + B_R
5      f(0) ← B_R − b_R(0)
6      for t in 0 < t < T do
7          f(t) ← (σ_u(t) + σ_l(t))/2
8      f(T) ← σ_u(T) − b_R(T)
9      repeat
10         f′ ← f
11         for t in 0 < t < T do
12             f(t) ← (f(t − 1) + f(t + 1))/2
13             f(t) ← max{min{f(t), σ_u(t)}, σ_l(t)}
14     until max|f′ − f| < ϵ
15     for t in 0 ≤ t < T do
16         u_R(t) ← f(t + 1) − f(t)
17     return u_R
```

of polygon vertices. This implies that the solution $u_R^*(t)$ can be computed as efficiently. However, this complex algorithm is out of scope of this thesis, and therefore we describe an efficient yet simple way to calculate $u_R^*(t)$, Algorithm 2.1, taken from [BKT15] and based on Li and Klette [LK07, LK11].

The algorithm iteratively makes an initial path as straight as possible within the harvesting polygon, until a stopping criteria specified by ϵ is satisfied. The algorithm is shown to have guaranteed convergence with regards to precision.

With the optimal solution to the original problem calculated, we move on the the backup battery problem.

The backup battery problem. In most practical implementations, it is not possible to arbitrarily mix input energy from one or the other source, and therefore one needs to do a binary decision at every given moment about which energy source to use. Let us use the term switching interval as a lower bound on the time between

switching from using the backup battery to using the rechargeable energy storage and vice versa. Now, there are two questions to be answered: *What is a suitable switching algorithm that determines the energy source for the next time interval?* And, *what is a suitable switching interval?*

We first provide a sketch of the algorithm: *We use the backup battery only if there is the danger of the rechargeable energy storage being depleted*, and this is checked every switching interval. This approach is not optimal, of course, but it can be shown that optimality is regained if the capacity of the battery is increased by an easily computable amount. For realistic scenarios, this increase is practically negligible and on the order of a few percent.

Precisely speaking, we partition the length of a time interval of length 1 into S switching intervals of length $\Delta t = 1/S$. For each switching time interval $[t + \frac{i}{S}, t + \frac{i+1}{S})$ for $0 \leq i < S$, the algorithm decides whether to use the backup battery or the rechargeable energy storage. We use the backup battery only if there is the danger of energy scarcity so as to minimize the use of the backup battery. In other words, if $b_R^*(t + \frac{i}{S}) < \frac{u_R^*(t)}{S}$, we use the backup battery with the amount $\frac{u_R^*(t)}{S} - b_R^*(t + \frac{i}{S})$ for the time interval $[t + \frac{i}{S}, t + \frac{i+1}{S})$ in order to satisfy the requested energy demand even if no energy is harvested. Otherwise if there is no energy scarcity danger, we use the rechargeable energy storage.

The above algorithm is not optimal as the backup battery is possibly used more than necessary, if in an interval of length $1/S$ part of the energy could have come from the rechargeable energy storage instead of the backup battery. But it can easily be shown that optimality is regained if the capacity of the rechargeable energy storage is increased by the maximal value of $\frac{u_R^*(t)}{S}$ for all $0 \leq t < T$.

Example. We end this section with an example to illustrate the aforementioned use functions for the backup battery problem, and their optimization. In Figure 2.2, an energy harvesting function $p(t)$ is shown, while $u_{\mathsf{BASE}} = 0.5$ and $B_R = 24$. The total simulation horizon is 288 time steps. One can see how the optimal total use function $u^*(t)$ responds to the changes in the harvested input energy, and how it is never less than the base consumption u_{BASE}.

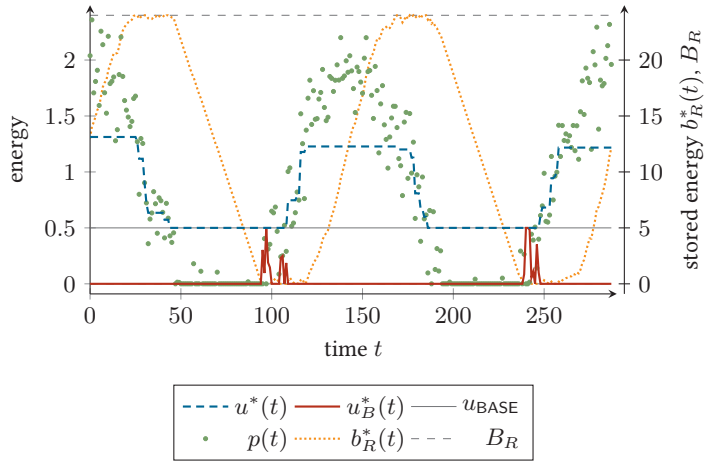


Figure 2.2: The unique optimal total energy use $u^*(t)$ (solution to Optimization problem 2.2), the backup battery use $u_B^*(t)$, and the corresponding energy in the rechargeable energy storage

The diagram also visualizes the partitioning of the use function as the amount of energy drawn from the backup battery $u_B^*(t)$ is shown as well. We use the discretized selection algorithm as explained in the previous paragraph, with a small switching interval $1/S \ll 1$.

2.5 Model Predictive Control Energy Use

In a realistic setting, a perfect estimate of future harvested energy is not available. Therefore, we employ the well known concept of model predictive control, as presented by Kwon and Han [KH06] for general models, and by Buchli et al. [BKT15] for the original problem. In contrast to the optimal solutions shown before, this heuristic approach can cope with deviations between the predicted and actual harvested, or scheduled and actually used energy. We name the approach FHC.

Algorithm 2.2: Computing the FHC use function for the original and backup battery problems

procedure get FHC ($\tilde{p}_t$, $b_R(t)$, $b_R(t+T_H)$, T_H, B_R, ϵ)
  $\tilde{u}_t \leftarrow$ get optimal ($\tilde{p}_t$, $b_R(t)$, $b_R(t+T_H)$, T_H, B_R, ϵ) ▷ Algorithm 2.1
  $u_R^{\text{FHC}}(t) \leftarrow \tilde{u}_t(0)$
  return $u_R^{\text{FHC}}(t)$ ▷ For the original problem
  $u^{\text{FHC}}(t) \leftarrow \max\{u_{\mathsf{BASE}}, \tilde{u}_t(0)\}$
  return $u^{\text{FHC}}(t)$ ▷ For the backup battery problem

The concept of finite horizon control is rather intuitive: At every time step t, the optimal use functions for the rechargeable energy storage and the backup battery are computed based on the current energy in the rechargeable storage and energy predictions. We only apply the optimal use functions in the next time interval $[t, t+1)$. At time $t+1$ and all future time steps, we repeat the same process. The optimal use functions are determined for a finite horizon of length T_H, namely for the time interval $[t, t+T_H)$. If T_H is sufficiently large, and the energy predictions are accurate, then the FHC solutions matches the optimal ones.

Finite horizon control. Let us now explain finite horizon control (FHC) more concretely. At some time t, we know the currently stored energy $b_R(t)$, and we have a prediction of harvested energy for the horizon T_H, $\tilde{p}_t(\tau)$. Then we use Algorithm 2.2 to determine the use function for next time interval $[t, t+1)$. This is $u_R^{\text{FHC}}(t)$ and $u^{\text{FHC}}(t)$ for the original and backup battery problem, respectively. In the second case, the partitioning of $u^{\text{FHC}}(t)$ among the rechargeable energy storage and the backup battery can then be done using the discrete switching algorithm as explained in the previous section.

Finally, note that Algorithm 2.2 also uses as input the required energy at the end of the horizon $b_R(T_H)$. Except when short horizons are used, this value does not impact the use function much. In the case that the horizon T_H is a one day, one year or

similar, it is reasonable to use $b_R(t + T_H) = b_R(t)$, while we found $b_R(t + T_H) = B_R/2$ to be fine in the general case.

The performance of FHC depends on the harvested energy prediction accuracy. Still, the FHC algorithm automatically considers estimation errors. For example, if the harvested energy was overestimated, then the rechargeable energy storage would initially be drained more aggressively. Later, the scheme would compensate by reducing the consumption, or using the backup battery. On the other hand, if the estimate was too conservative, the consumption would first decrease, and then increase as the rechargeable energy storage approaches its capacity.

Efficient implementation. The implementation of finite horizon control demands substantial computational resources. As this is not easily available on all embedded devices, we reference here an implementation based on LookUp Tables by Moser et at. [MTBB09] which greatly reduces the run-time computational requirements.

For a given estimate, the optimal use at time t for a set of energy storage levels $b_R(t)$ is determined. Therefore, for a fixed or periodic prediction of harvested energy, just a few parameters need to be stored to cover all initial rechargeable energy storage levels. In case of a reasonably small set of different predicted harvested energy traces, the necessary amount of storage and computation is small.

For an example scenario, one time interval is 10 minutes $T = 10\,\mathrm{min}$, and the prediction horizon is one hour $T_H = 1\,\mathrm{h}$. Assume that 6 different predicted harvested energy traces are sufficiently representative for this location, i.e., they cover an hour at night, with sunny and cloudy weather, etc. If one assumes a quadratic approximation is used for non-precomputed energy storage levels, then for each of them just three $32\,\mathrm{bit}$ values need to be stored, which results in $32\,\mathrm{bit} \cdot 3 \cdot 6 = 72\,\mathrm{B}$. If one assumes 25 discrete values for the rechargeable energy storage level, instead of using an approximation, one would have to store $32\,\mathrm{bit} \cdot 25 \cdot 6 = 600\,\mathrm{B}$.

Example. As was done before, we end this section with an example. We illustrate the FHC use function $u^{\mathrm{FHC}}(t)$, and compare it to the optimal solution for the backup battery problem. Figure 2.3, an energy harvesting function $p(t)$ is shown, while $u_{\mathsf{BASE}} = 0.5$

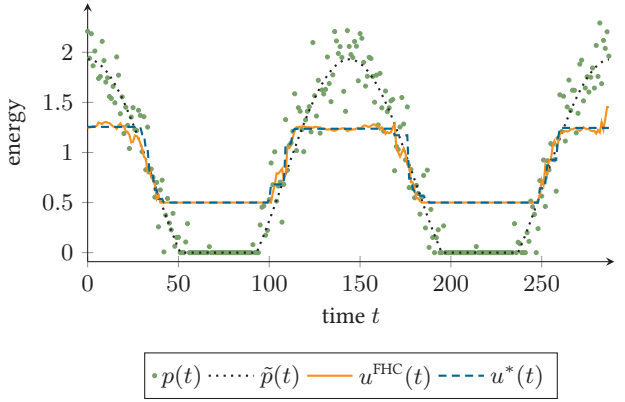


Figure 2.3: The finite horizon control total energy use $u^{\text{FHC}}(t)$, compared to the optimal total energy use $u^*(t)$ (solution to Optimization problem 2.2)

and $B_R = 24$. The total simulation horizon is 288 time steps. The estimate of future harvested energy is the same for every time of estimate t, $\tilde{p}_t(\tau) = \tilde{p}_s(\tau)$ for every time t and s. The estimate of harvested energy is accurate in this case, and one can see how the finite horizon control solution matches the optimal one closely. Still, $u^{\text{FHC}}(t)$ features nose, because the expected and actual harvested energy differ.

2.6 Solar Energy Prediction

As seen before, in order to supply a non-optimal but practical solution to the two optimization problems, finite horizon control needs a harvested energy estimate. This estimate aims to facilitate the system's adaption to its environment, by enabling the energy manager to switch modes of operation proactively.

A plethora of solar energy prediction algorithms exist in the state of the art. However, we focus only on ones suitable for the context of resource constrained energy harvesting systems, and

which can be used to estimate future harvested energy. We notably exclude schemes based on externally sourced information, such as externally computed weather forecasts, as they require additional infrastructure and are not applicable in the general case.

We first introduce existing estimators designed for short-term predictions, i.e., for each time-slot of the following day. Then, we present our own new schemes for the same time scale. We end this section by talking about long-term predictions, where a weekly prediction is made for the following year.

2.6.1 Commonly Used Estimators

Exponentially weighted moving average. The first energy prediction scheme, exponentially weighted moving average (EWMA) by Kansal et al. [KHZS07], is simple yet widely used. As its name suggests, it predicts energy $\tilde{p}_t(\tau)$ as an exponentially weighted moving average of energy harvested in the same time-slot in the previous days. The reader recalls, this means that the contribution of old data to the average is exponentially decreasing. In the most widely used case the prediction horizon T_H is one day, and then EWMA can be computed recursively using:

$$\begin{aligned} &\forall \tau \in (t, t+T_H] : \\ &\quad \tilde{p}_t(\tau) = \alpha \cdot \tilde{p}_{t-T_H}(\tau - T_H) + (1-\alpha) \cdot p(\tau - T_H) \end{aligned} \tag{2.15}$$

where α is a weighting factor between 0 and 1. The advantages of EWMA are clear as the scheme is very easy to implement, and it is effective when there are no major day-to-day weather changes. The main disadvantage is a high error when there are changing weather conditions, for example when sunny and cloudy days are alternating.

Weather-conditioned moving average. Another scheme better suited for alternating weather is weather-conditioned moving average (WCMA) by Recas Piorno et al. [RPBASR09]. This scheme observes the average energy harvested at a given hour in the past D days, and introduces a scaling factor GAP_k that quantifies how the current day's weather is with respect to the average. The scaling

factor is then used to make a prediction, where the horizon T_H is until the end of the day, using:

$$\forall \tau \in (t, t+T_H] : \\ \tilde{p}_t(\tau) = \alpha \cdot p(t) + GAP_k \cdot (1-\alpha) \cdot \sum_{i=1}^{D} \frac{p(\tau - d \cdot i)}{D} \tag{2.16}$$

where α is a weighting factor between 0 and 1, and d is the length of one day. WCMA responds to a weather change after one time-slot, while EWMA needs a full day to take such a change into account. Therefore, it is expected that the former scheme is more suitable for frequently changing weather conditions. This comes at a cost though, which is the computation time needed to derive the GAP_k factor, where k is a parameter that denotes the number of time-slots used for calculating the factor. We refer the reader to [RPBASR09] for details.

Profile energy prediction model. A third approach, introduced by Cammarano et al. [CPS12], is the profile energy prediction model (ProEnergy). Instead of utilizing certain average values as in the former schemes, ProEnergy takes a different approach by keeping D full days of observed energy harvesting traces, called profiles. Ideally, these D profiles are chosen as representatives of different weather conditions encountered. Thus, to make a prediction, we need to find the most similar day among the memorized D profiles. If π is this similar profile, α is a weighting factor between 0 and 1, and d is the length of one day, the predicted energy can be computed using (2.17).

$$\forall \tau \in (t, t+T_H] : \\ \tilde{p}_t(\tau) = \alpha \cdot p(t) + (1-\alpha) \cdot \pi(\tau) \tag{2.17}$$

where the horizon T_H is until the end of the day. Practically speaking, ProEnergy involves building and possibly updating the D representative profile list, then finding the most similar profile, and finally calculating the predicted value. By taking advantage of a representative list of profiles, ProEnergy promises to outperform both WCMA and EWMA. The drawback is a larger computation and

memory footprint needed to run the scheme. Note, for medium- and long-term energy predictions, the weighting factor α can be a function of the similarity of the current day and the most similar profile π.

2.6.2 Estimators Based on Atmospheric Transmittance

In this section we present our proposed solar energy prediction schemes. We first overview the extraterrestrial solar model. Following this, we propose solar energy prediction schemes that leverage the extraterrestrial solar model to yield significantly better prediction accuracy compared to state-of-the-art with low computation and memory overhead.

2.6.2.1 Extraterrestrial Solar Energy

Determining the position of the Sun in the sky is a well studied phenomenon. Using the Sun's position, one can deterministically calculate the energy harvested by any solar panel above the atmosphere, for any given time and geographical location, using the extraterrestrial irradiation model. This we name the extraterrestrial solar energy, and denote $p^{\mathrm{et}}(t)$. Regarding this matter, our calculations are taken from Iqbal's book [Iqb83]. To calculate $p^{\mathrm{et}}(t)$, parameters specified in Table 2.1 need to be used.

For our analysis and evaluations, we assume a *nominal* solar panel, which is $1\,\mathrm{m}^2$ and oriented horizontally, tangent to the Earth's surface. However, extending all results to panels of arbitrary size and orientation is possible using well established trigonometry.

2.6.2.2 Atmospheric Transmittance

Now we define atmospheric transmittance and explain how it can be computed and used to design energy prediction schemes. The average atmospheric transmittance during time interval t is the ratio between the energy harvested during time interval t, and the extraterrestrial energy harvested during the same interval (2.18).

Table 2.1: Parameters used to calculate the extraterrestrial solar energy $p^{\mathrm{et}}(t)$, for a nominal solar panel

Latitude	θ^{lat}
Longitude	θ^{lon}
Eccentricity correction factor of Earth's orbit on day d	ε_d
Day angle on day d	Γ_d
Rate of extraterrestrial energy	$I = 1353\,\mathrm{W\,m^{-2}}$
Solar declination angle on day d	δ_d
Equation of time on day d	EoT_d
Time zone of a location without daylight saving	time_zone
Apparent solar time at day d and time t	$\mathrm{AST}_{d,t}$
Solar angle at day d and time t	$\omega_{d,t}$
Solar angle at sunrise on day d	ω_d^{sr}
Solar zenith angle at day d and time t	$\theta_{d,t}^{z}$

$$s(t) = p(t)/p^{\mathrm{et}}(t) \tag{2.18}$$

The estimated value of atmospheric transmittance is analogously given below. Note that the extraterrestrial energy in the equation is not a predicted value, as the extraterrestrial energy of every interval is analytically obtainable.

$$\tilde{s}(t) = \tilde{p}(t)/p^{\mathrm{et}}(t) \tag{2.19}$$

Predicting $\tilde{s}(t)$ and using it to compute the value of $\tilde{p}(t)$ is expected to improve prediction accuracy over existing prediction schemes. This is because $p^{\mathrm{et}}(t)$ accurately models the diurnal solar cycle and the yearly seasonal variations. Therefore, any error caused by *predicting* these deterministic variations will be removed.

To illustrate this advantage, we compare the relative standard deviation of atmospheric transmittance and solar energy within a given day. Relative standard deviation is a measure of relative variation in data. For five years of data (2005-2009) at a given location SE, as introduced in Section 2.9.1, we measured the relative standard deviation of the two aforementioned values for each day, and we plotted the corresponding histograms. Data is taken from

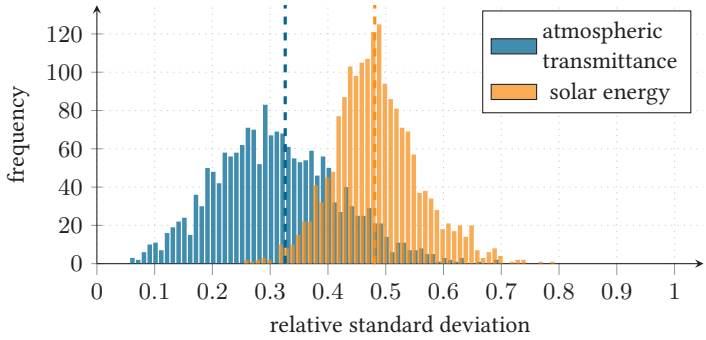


Figure 2.4: Histogram of the relative standard deviation of atmospheric transmittance and solar energy within a given day

the NSRDB [SXL+18]. Note that values less than 10 % of the given day's maximum value were omitted from computation.

As seen in Figure 2.4, the relative standard deviation of atmospheric transmittance has a lower mean value compared to the corresponding metric for solar energy. This is precisely because considering atmospheric transmittance removes variations caused by the diurnal solar cycle. Therefore, the prediction error in transmittance based schemes is expected to be less compared to the existing schemes, which work directly on higher-variance solar energy.

2.6.2.3 The New Estimators

Finally, we propose four prediction schemes based on atmospheric transmittance. Weather-conditioned moving average – transmittance (WCMA-T) and profile energy prediction model – transmittance (ProEnergy-T) are simple enhancements of state-of-the-art schemes, while exponentially weighted moving average – transmittance (EWMA-T) and delta – transmittance (Delta-T) are novel.

Enhancement of commonly used estimators. Let us demonstrate how transmittance is used to predict energy by explaining ProEnergy-T. First, remember that transmittance can not be directly

measured, but has to be computed from the observed harvested energy (2.18). Next, with the transmittance at time t in place, as well as D transmittance profiles memorized, one may predict the transmittance at time τ using:

$$\begin{aligned} &\forall \tau \in (t, t + T_H] : \\ &\quad \tilde{s}_t(\tau) = \alpha \cdot s(t) + (1 - \alpha) \cdot \pi^{\mathrm{S}}(\tau) \end{aligned} \tag{2.20}$$

where π^{S} is the most similar transmittance profile. Finally, by applying (2.19) we obtain a predicted harvested energy value. WCMA-T is defined in a similar manner, where (2.20) is replaced by the following equation:

$$\forall \tau \in (t, t + T_H] : \\ \tilde{s}_t(\tau) = \alpha \cdot s(t) + GAP_k \cdot (1 - \alpha) \cdot \sum_{i=1}^{D} \frac{s(\tau - d \cdot i)}{D} \tag{2.21}$$

Computationally, both schemes are the same as their original counterparts, with the added overhead of computing the transmittance.

Exponentially weighted moving average – transmittance. The idea behind EWMA-T is that transmittance does not change abruptly within a given day. Therefore it can be predicted as the exponentially weighted moving average of previous hours. It is thus defined as:

$$\begin{aligned} &\forall \tau \in (t, t + T_H] : \\ &\quad \tilde{s}_t(\tau) = \alpha \cdot \tilde{s}_{t-1}(t) + (1 - \alpha) \cdot s(t) \end{aligned} \tag{2.22}$$

This scheme promises to combine the benefit of predicting transmittance with the simplicity of EWMA.

Delta – transmittance. The intuition behind Delta-T is that the change in transmittance from time-slot $t - 1$ to time-slot t follows a similar pattern across the last D days. Therefore, for time horizon T_H of one day, the prediction can be formulated as follows.

$$\forall \tau \in (t, t + T_H] : \\ \tilde{s}_t(\tau) = s(t) \cdot \frac{\sum_{i=1}^{D} s(\tau - i \cdot T_H)}{\sum_{i=1}^{D} s(\tau - i \cdot T_H - 1)} \tag{2.23}$$

Because of the need to store $D \cdot t$ transmittance values, and to compute the associated sums, Delta-T requires somewhat more resources than EWMA-T.

2.6.3 Comparison of Computation and Memory Costs

Before ending this section, let us revisit all of the aforementioned schemes in order to compare their computation and memory costs. We thus first present Table 2.2, where the number of computations for all of the schemes are compared. In the table, o_{A} denotes an arithmetic operation (addition, subtraction, multiplication, or the like), while o_{T} denotes a division or a trigonometric function. For transmittance based schemes, the cost for calculating the current extraterrestrial energy $p^{\mathrm{et}}(t)$, as well as the cost of calculating transmittance from energy and vice versa, has *not* been included in the computational cost, but given separately as 'transmittance overhead'. Note that for ProEnergy and ProEnergy-T, the number of computations needed for a profile update is not given.

Table 2.3 displays the memory cost, i.e., the variables that need to be stored between two consecutive time intervals. While presenting the memory cost, the time horizon T_H is assumed to be one day.

2.6.4 Long-Term Estimation

In certain use cases when long-term operation is intended, it is reasonable to define that a time interval $[t, t + 1)$ is one week long. Here, we present a solar energy prediction scheme designed for this scenario, where the prediction horizon is one year.

Astronomical prediction (Astro) by Buchli et al. [BSBT14b] uses the extraterrestrial solar model to predict solar energy for every

Table 2.2: Computational costs of various solar energy estimators

Scheme		Computations per estimate	Additional daily computations
EWMA	(2.15)	$3o_{\mathrm{A}}$	
WCMA	GAP_k	$2k{\cdot}o_{\mathrm{A}} + o_{\mathrm{T}}$	
	(2.16)	$7o_{\mathrm{A}}$	
ProEnergy	choosing π	$6D{\cdot}o_{\mathrm{A}}$	
	(2.17)	$3o_{\mathrm{A}}$	
WCMA-T		Same as WCMA	
ProEnergy-T		Same as ProEnergy	
EWMA-T	(2.22)	$3o_{\mathrm{A}}$	
Delta-T	(2.23)	$3o_{\mathrm{A}} + o_{\mathrm{T}}$	
transmittance	$p^{\mathrm{et}}(t)$	$11o_{\mathrm{A}} + 2o_{\mathrm{T}}$	$42o_{\mathrm{A}} + 7o_{\mathrm{T}}$
overhead	(2.18) and (2.19)	$o_{\mathrm{A}} + o_{\mathrm{T}}$	

given week t in the following year T_H as:

$$\begin{aligned} &\forall \tau \in (t, t+T_H] : \\ &\quad \tilde{p}(\tau) = s \cdot p^{\mathrm{et}}(\tau) \end{aligned} \tag{2.24}$$

Due to the large time-scale, it is assumed that the atmospheric transmittance s is invariant over time. This value can be either observed in the first few weeks of operation. Note that because of this, the predicted energy value $\tilde{p}(\tau)$ is not a function of the prediction time t.

The computational and memory cost of this scheme is miniscule, as it is sufficient to statically store 52 predictions for each week of the year.

2.7 Non-Ideal System Model

Starting from our simple system model in Section 2.3, we considered already two implementation artifacts, namely that it is only possible to draw energy from one of the two energy sources (Section 2.4),

Table 2.3: Memory costs of various solar energy estimators

Scheme	Variables to be stored between estimations	Size
EWMA	Predicted energy, last T_H values	T_H
WCMA	Observed energy, last D days	$D \cdot T_H$
	Carryover values	$k + T_H$
ProEnergy	Observed energy, chosen D days	$D \cdot T_H$
	Observed energy, last T_H values	T_H
	Carryover values	D
WCMA-T	Same as WCMA	
ProEnergy-T	Same as ProEnergy	
EWMA-T	Predicted transmittance, last value	1
Delta-T	Observed transmittance, last D days	$D \cdot T_H$
	Carryover values	T_H
transmittance overhead	Daily and hourly parameters	16

and that the harvested and consumed energy may differ from estimations or consumption goals (Section 2.5). Here, we describe how to adapt the model in order to incorporate inefficiencies that may exist in the power management hardware without affecting the optimality of the solution.

Constraints. The following non-ideal behavior is modeled:

- If the harvesting device generates energy $\hat{p}(t)$ in $[t, t+1)$, then the energy in the rechargeable energy storage only increases by $\sigma_p \cdot \hat{p}(t)$.
- If the load uses rechargeable energy $\hat{u}_R(t)$ in $[t, t+1)$, then the energy stored in the rechargeable energy storage decreases by $\sigma_R \cdot \hat{u}_R(t)$. In addition, there is a leakage δ_R of the rechargeable energy storage.
- If the load uses backup energy $\hat{u}_B(t)$ in $[t, t+1)$, then the energy stored in the backup battery reduces by $\sigma_B \cdot \hat{u}_B(t)$. In addition, there is a leakage δ_B of the backup battery.

The last item is mainly relevant for computing the lifetime L of the node. It can be determined using the initial energy B_B in the

backup battery, and the accumulated use of the backup battery:

$$B_B = L \cdot \delta_B + E(0, L) = L \cdot \delta_B + \sigma_B \cdot \sum_{t \in [0,L)} \hat{u}_B(t) \tag{2.25}$$

Optimization problems. These extended relations consider the mentioned implementation artifacts:

$$\begin{gathered} \hat{u}_B(t) \geq 0 \\ \hat{u}_R(t) \geq 0 \\ \hat{u}(t) = \hat{u}_R(t) + \hat{u}_B(t) \geq \hat{u}_{\mathsf{BASE}} \\ 0 \leq \hat{b}_R(t+1) = \hat{b}_R(t) + \sigma_p \cdot \hat{p}(t) - \sigma_R \cdot \hat{u}_R(t) - \delta_R \leq \hat{B}_R \end{gathered} \tag{2.26}$$

If we perform the variable transformations given in (2.27), then we directly obtain from (2.26) the abstract relations in (2.6) and (2.7) and thus, all results which refer to the original and backup battery optimization problems apply as well.

$$\begin{gathered} u_R(t) = \sigma_B \cdot \hat{u}_R(t) \\ u_B(t) = \sigma_B \cdot \hat{u}_B(t) \\ u(t) = \sigma_B \cdot \hat{u}(t) \\ u_{\mathsf{BASE}} = \sigma_B \cdot \hat{u}_{\mathsf{BASE}} \end{gathered}$$

$$\begin{aligned} B_R &= \frac{\sigma_B}{\sigma_R} \hat{B}_R \\ b_R(t) &= \frac{\sigma_B}{\sigma_R} \hat{b}_R(t) \\ p(t) &= \frac{\sigma_B}{\sigma_R} (\sigma_p \cdot \hat{p}(t) - \delta_R) \end{aligned} \tag{2.27}$$

2.8 Implementation and Characterization

Here we demonstrate the accuracy of the non-ideal model, as presented in Section 2.7, by describing and characterizing an imple-

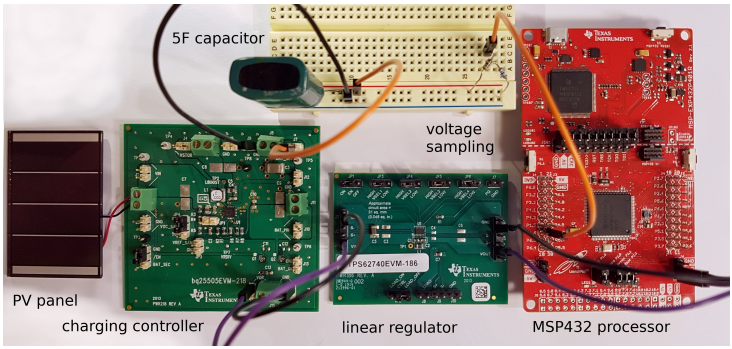


Figure 2.5: Components of our hardware implementation of the energy management system with a backup battery. Backup battery not shown

mentation of an energy harvesting embedded system with a backup battery.

2.8.1 Hardware and Software Setup

The hardware is shown in Figure 2.5, and consists of a photovoltaic energy harvester, a power management unit, a super-capacitor as the rechargeable energy storage, a DC power supply as the backup battery, and a microcontroller unit as the load. The photovoltaic panel is an AM-5412 from SANYO Semiconductor [San08]. A $C^{\mathrm{SC}} = 5\,\mathrm{F}$ super-capacitor [Eat17] is used, and during operation its voltage ranges between $3.3\,\mathrm{V}$ and $4.3\,\mathrm{V}$. The backup battery was emulated with a DC power supply, providing a constant $3.3\,\mathrm{V}$ when needed. Primary batteries like the SAFT LS 14500 [Saf19] show an almost constant voltage throughout their lifetime. Therefore, there is no need to consider additional, time-dependent efficiencies. In addition, a battery's temperature-dependent capacity impacts the system's lifetime, but does not change the optimality of our approach.

The power management unit consists of a bq25505 harvesting management chip [Tex19a], and a TPS62740 step-down converter

[Tex14]. The bq25505 harvesting management chip performs passive maximum power point tracking for maximum energy transfer from the photovoltaic panel to the super-capacitor. The chip is also a power multiplexer, switching between the rechargeable and backup batteries in the following way. If the super-capacitor's voltage is lower than 3.3 V, the chip switches to the backup battery. And when the super-capacitor's voltage is larger than 3.4 V, the chip switches to the rechargeable energy storage. In practice, this means that the backup battery is used while the super-capacitor's voltage increases from 3.3 to 3.4 V. This operation approximates the model described in Section 2.4.2.

The microcontroller unit is a MSP432P401R board from Texas Instruments [Tex19b], commonly found in embedded applications. Two tasks are implemented on it, the 'power manager' and the 'adaptive consumer'. The 'power manager' implements the scheduling scheme and harvested energy estimator. This task includes sampling of the super-capacitor's voltage, which allows to determine the currently stored energy $b_R(t)$. The sampling is done by connecting an ADC port to the super-capacitor through an appropriate high-resistance voltage divider. The 'adaptive consumer' can consume different energy values, which emulates sensing or communicating with different rates, different data processing algorithms or actuating LEDs.

2.8.2 Characterization

To characterize the described system, we conducted four experiments. Interpreting results of these experiments, we express the harvested energy as a function of illumination, and quantify the parameters of the non-ideal model as presented in Section 2.7.

Experiments involving illumination are done in a laboratory environment. There, a solar test-bed built by Sigrist [Sig20] recreates the desired illumination levels. The test-bed completely encloses a photovoltaic panel, such that it is exposed to a desired illumination trace using a programmable light source. All measurements are made using a RocketLogger [SGL+17], which monitors various voltage and current channels in parallel. Most importantly, the PV panel's harvested power P_{out}^{PV} and the microcontroller's power

consumption P_{in}^{MSP} are obtained through voltage and current measurements, while the rechargeable energy storage's voltage V^{SC} is used to estimate the net power flow in the rechargeable energy storage P_{est}^{SC}:

$$P_{est}^{\text{SC}} = \frac{1}{2} C^{\text{SC}} \cdot \frac{\partial}{\partial t} V^{\text{SC}\,2} \tag{2.28}$$

PV panel characterization. We first characterize the AM-5412 photovoltaic energy harvesting panel together with the bq25505 which performs passive maximum power point tracking. This enables us to model how much energy is harvested depending on the illuminance level. A total of 13 illuminance levels are used, from 0 to $110\,\text{klx}$ in even steps. For each illuminance level E_v, we charged the super-capacitor from 3.3 to $4.2\,\text{V}$, and measured the power produced by the PV panel P_{out}^{PV}. The following linear fit can be made:

$$P_{out}^{\text{PV}} = 0.3791 \cdot 10^{-3}\,\text{W}\,\text{lx}^{-1} \cdot E_v\ [\text{W}]$$

This fit is used in Section 2.9 to create a harvested energy trace $p(t)$ from a trace of measured illuminance.

Charging characterization. Here we characterize the σ_p parameter of the non-ideal model, introduced in Section 2.7, which encapsulates losses in power point tracking on the bq25505, as well as charging losses of the super-capacitor. The characterization is performed with the microcontroller and linear regulator disconnected. First, we expose the PV panel to 13 illuminance levels (from 0 to $120\,\text{klx}$ in even steps), in order to measure the power produced by the PV panel P_{out}^{PV}, as well as the power stored in the rechargeable energy storage P_{est}^{SC}. Then we construct a linear fit that minimizes the root mean square error and links the harvested power P_{out}^{PV} and the stored power P_{est}^{SC}:

$$P_{est}^{\text{SC}} = 0.8529 \cdot P_{out}^{\text{PV}}\ [\text{W}]$$

Figure 2.6 depicts the measurements as well as the linear fit. The fit is not perfect due to measurement noise as well as other factors such as a non-linear behavior of the power point tracking. Still, the fit is highly accurate with a root mean square error of only

$4.313 \cdot 10^{-4}$. As a result, we can model the charging efficiency of this particular implementation as $\sigma_p = 0.8529$.

Discharging characterization. Now we present the characterization of the σ_R parameter of the non-ideal model, see Section 2.7, which quantifies the loss in power of the linear regulator, the power multiplexer, as well as the discharging losses of the super-capacitor. In this measurement setup the energy harvesting PV panel is disconnected after the super-capacitor is fully charged, and the microcontroller is set to consume at one of 7 power levels spaced regularly between 9.7 and $92.3\,\mathrm{mW}$. We measure the power consumed by the microcontroller P_{in}^{MSP}, and the power discharged from the super-capacitor P_{est}^{SC}. A linear fit that minimizes the root mean square error yields:

$$P_{est}^{\mathrm{SC}} = 1.0789 \cdot P_{in}^{\mathrm{MSP}}\ [\mathrm{W}]$$

As Figure 2.6 shows, the fit appears to be sufficiently accurate as the root mean square error is $1.020 \cdot 10^{-3}$. As a result, the discharging efficiency for our implementation is set to be $\sigma_R = 0.9269 = 1/1.0789$.

Rechargeable energy storage leakage characterization. The final measurement quantifies the decrease in stored energy caused by the super-capacitor's internal leakage and the sampling of the super-capacitor's voltage. For this measurement, the rechargeable energy storage was initially fully charged, and both the PV energy harvester and the consumer were disconnected. Over 38 hours, an average power dissipation of $P_{est}^{\mathrm{SC}} = 93.50\,\mu\mathrm{W}$ was observed. As the value is low relative to the chosen experimental setup, this leakage is not taken into account in the modeling of the system.

2.8.3 Overhead for Finite Horizon Control

We evaluated the direct implementation of the finite horizon control use function, according to Algorithm 2.2. An implementation on the chosen MSP432 microcontroller resulted in the following findings: For a $T_H = 8\,\mathrm{h}$ horizon with $1\,\mathrm{h}$ time steps, using single-precision

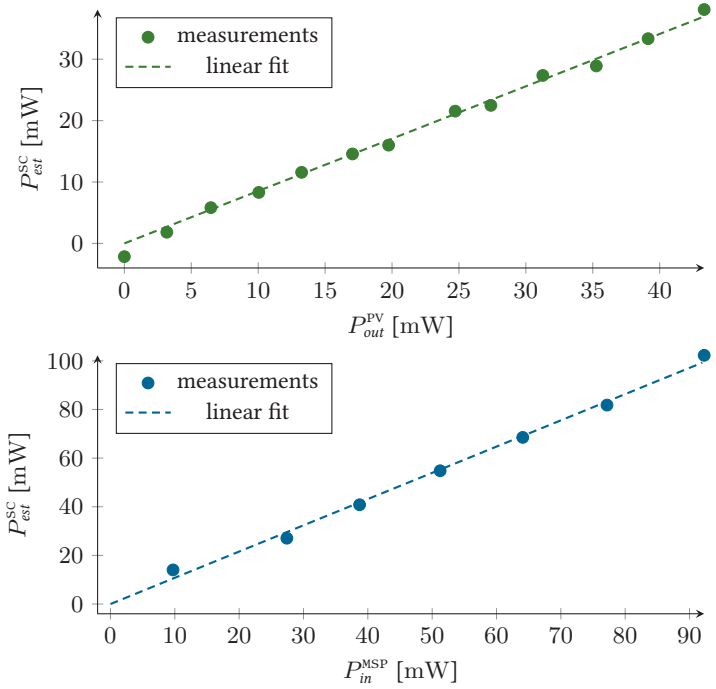


Figure 2.6: Characterizing the charging efficiency σ_p (top) and discharging efficiency σ_R (bottom)

floating point arithmetic, and permitting an $\epsilon = 0.01$ error in the use function u^{FHC} which corresponds roughly to an error of $0.5\,\%$ relative to the maximum value, it takes $0.2\,\text{s}$ to compute $u^{\text{FHC}}(t)$.

2.9 Trace-Based Experimental Results

In this section, we conduct two groups of simulated experiments in order to evaluate our aforementioned work. First we look into the precision of solar energy estimators in detail, and demonstrate the benefit of our atmospheric transmittance based schemes. Then

Table 2.4: An overview of used locations

Abbr.	Location	Latitude	Longitude	Alt. [m]	ID
UT	Utqiagvik	71.32° N	156.77° W	9	700260 [3]
AN	Anchorage	61.17° N	150.02° W	35	26451 [1]
AD	Adak	51.88° N	176.65° W	5	704540 [3]
SE	Seattle	47.47° N	122.32° W	122	727930 [3]
FA	Fargo	46.93° N	96.82° W	274	727530 [3]
PO	Portland	45.58° N	122.58° W	12	24229 [1]
BH	Benton Harbor	42.13° N	86.43° W	196	726355 [3]
NY	New York	40.65° N	73.80° W	5	744860 [3]
BA	Baltimore	39.17° N	76.67° W	47	93721 [1]
LA	Los Angeles	34.05° N	118.23° W	82	722874 [3]
PH	Phoenix	33.45° N	111.98° W	337	722780 [3]
EP	El Paso	31.89° N	106.40° W	1194	23044 [1]
CC	Corpus Christi	27.77° N	97.50° W	13	12924 [1]
HO	Honolulu	21.32° N	157.93° W	2	911820 [3]

we simulate our whole system in several fundamentally different scenarios. Here we compare the FHC use function with the optimal solution as well as a simple reactive consumption function, we demonstrate the performance of FHC when combined with future energy estimators of various accuracy, and we illustrate the system's performance as well as the use of the backup battery.

Before going into experiments, let us introduce the used data sets for solar energy harvesting.

2.9.1 Harvested Energy Data

Outdoor locations. Data from a total of 14 geographical locations were used, all from the open National solar radiation database (NSRDB) [SXL+18], which collects data from many measurement stations throughout the USA. Table 2.4 introduces locations we used, with their name, geographical information, and identification number. Note that [1] means we used a measurement station from the first version of the database, NSRDB v1, which collects data from 1961 to 1990, while [3] refers to the third version of the database NSRDB v3, and especially data from 1998 to 2010.

Table 2.5: A description of used locations

Abbr.	SD [h]	SD [%]	Köppen climate description
UT	1858.7	30	Tundra (Et)
AN	2061.2	46	Subarctic (Dfc)
AD	–	–	Cold-summer Mediterranean (Cfc)
SE	2169.7	49	Temperate oceanic (Cfb)
FA	2629.1	59	Hot-summer humid continental (Dfa)
PO	2340.9	52	Warm-summer Mediterranean (Csb)
BH	–	–	Hot-summer humid continental (Dfa)
NY	2534.7	57	Humid subtropic (Cfa)
BA	2581.7	58	Humid subtropic (Cfa)
LA	3254.2	73	Warm-summer Mediterranean (Csb)
PH	3871.6	87	Hot desert (Bwh)
EP	3762.5	85	Cold desert (BWk)
CC	2636.3	59	Humid subtropic (Cfa)
HO	3035.9	68	Hot semi-arid (BSh)

To illustrate the diversity of chosen locations, Table 2.5 gives a comment on the climate and sunshine duration. Specifically, column SD [h] gives the mean number of hours with sunshine per year, while SD [%] gives the former number relative to the total number of daylight hours per year.

Indoor setup. Data describing harvestable photovoltaic energy indoors is not commonly available. We rely on the Indoor solar harvesting dataset [SGT19], which collects up to 3 years (2017-2020) of data at six locations inside a university office building in Zürich, Switzerland. In this chapter, we work with data observed at three locations, named L14, L16 and L17. At location L14, from time to time there is direct sun exposure, particularly in the morning hours and during summer. This deployment is placed on a wall near a north-east facing window.

Both measurement stations L16 and L17 are located in the same office, however in different surroundings. L16 is placed on a table facing up, and it has significant exposure to light throughout the day. L17 is mounted on a wall high up, and near a window which

is sometimes partially blocked by curtains. No direct exposure to sunlight was observed on either location.

2.9.2 Precision of Energy Estimators

In this section, we evaluate all aforementioned short-term estimators for future harvested solar energy: EWMA, WCMA, ProEnergy, WCMA-T, ProEnergy-T, EWMA-T, and Delta-T. In this sense, short-term means time-slots are less than a day.

Four simulated experiments are used for the evaluation. First, we show how estimators compare at different locations. Then, with a fixed location, we investigate the performance of each scheme at different times of day, and for different time-slot lengths. Finally, we end the section with an analysis of an estimator's precision versus computation complexity.

For all experiments, harvested energy is expressed per meter square, and for an ideal solar panel.

Setup. Throughout this section, we focus seven locations. Ordered from the pole to the equator, these are UT, AD, SE, FA, NY, PH, and HO. For each location, five years of data (2005-2009) were used. One year of data (2005) was used for training each individual scheme, or picking the respective parameters. These parameters are as follows. The weighting factor α for EWMA and EWMA-T. The weighting factor α, number of past days D, and k parameter for WCMA and WCMA-T. The weighting factor α, number of stored days D, and the stored days for ProEnergy and ProEnergy-T. And finally, the number of past days D for Delta-T. All of these parameters were then fixed, and evaluation was conducted on the remaining four years.

Metrics. When evaluating an estimator's precision, we evaluate a prediction made for the following time slot unless specified otherwise, i.e., $\tilde{p}_{t-1}(t)$. Two metrics are used: the mean absolute error (MAE) and the mean absolute percentage error (MAPE). They are

defined as:

$$\text{MAE} = \frac{1}{T}\sum_{t=0}^{T-1}|\tilde{p}_{t-1}(t) - p(t)| \tag{2.29}$$

$$\text{MAPE} = \frac{1}{T}\sum_{t=0}^{T-1}\left|1 - {\tilde{p}_{t-1}(t)}/{p(t)}\right| \tag{2.30}$$

MAPE is usually present in related work to show the relative accuracy of a solar energy prediction scheme. However, we note that a relative measure might not be sufficient to give a complete overview of the estimators performance. Most notably, MAPE gives little information when harvested energy is close to zero. We therefore introduce MAE, which comments on the absolute accuracy of the prediction scheme, to supplement our evaluation.

2.9.2.1 Precision at Various Locations

We start off by evaluating the estimators' precision, in terms of MAE and MAPE, at the seven locations.

Setup. While evaluating both metrics, night time and low light time-slots are omitted. These are defined to be time-slots in which the energy harvested is less than 10 % of the daily maximum. This practice is common in related work [RPBASR09, CPS12]. A separate training and evaluation was conducted for the MAPE and MAE comparison.

Evaluation. The performance of all of the schemes, on the seven evaluation locations, is given in Figure 2.7 and Table 2.6. The figure shows the MAE and MAPE for every estimator and location, while the table presents the mean, standard deviation (σ), and 99-percentile value (P_{99}) for all locations together.

Regarding the existing schemes, what we can observe first is that EWMA is, with regards to MAPE, the least precise scheme overall. This holds as well as for individual locations, except PH and HO where it outperforms WCMA. For all other locations, WCMA is

Table 2.6: Precision of existing and proposed solar energy estimators, aggregated for 7 diverse locations

	MAE	σ^{MAE}	$\text{P}_{99}^{\text{MAE}}$	MAPE	σ^{MAPE}	$\text{P}_{99}^{\text{MAPE}}$
	$[\text{W h/m}^2]$			[%]		
EWMA	104.41	99.83	443.48	37.58	44.78	224.86
WCMA	85.02	85.86	388.22	29.65	33.88	155.72
ProEnergy	71.60	77.50	360.01	24.24	29.61	147.33
EWMA-T	69.90	83.43	391.05	22.68	29.01	149.50
WCMA-T	71.11	79.31	373.22	23.39	27.62	140.63
ProEnergy-T	63.88	75.48	354.10	20.97	27.44	140.86
Delta-T	69.63	85.42	395.20	22.21	30.25	154.51

more precise than EWMA. Regardless of the metric and location, ProEnergy preforms best of the non-transmittance based schemes.

Out of the transmittance based schemes, with regards to MAPE, the best scheme overall is ProEnergy-T, having the best precision for all evaluated locations except FA. WCMA-T has the least precision and performs arguably as good as ProEnergy (e.g., performing worse than it in PH, and better than it in SE). EWMA-T and Delta-T both perform slightly better than WCMA-T, though the exact amount depends on the actual location (e.g., for FA Delta-T is better, followed by EWMA-T and WCMA-T, while for AD it is EWMA-T followed by WCMA-T and Delta-T).

The results for MAE are similar to the MAPE case. However, we see the effect of the amount of harvestable solar energy. On the one side, HO has a lot of sunshine, so the absolute error is high for most of the schemes even though the relative one is not. On the other side, UT is located inside the Arctic Circle, thus the absolute error for all of the schemes is low.

2.9.2.2 Precision at Various Times of Day

To supplement the evaluation at different locations, in this experiment we evaluate the performance of all the estimators for different times of day.

Setup. Figure 2.8 shows the MAPE and MAE for every estimator, at location SE, averaged for each hour of a day. Here, every data point with positive harvested energy has been taken into account. A separate training and evaluation was conducted for the MAPE and MAE comparison.

Evaluation. With regards to MAPE, all schemes perform the worst in the early morning hours, up to 8 o'clock. The exception is EWMA, as its precision is roughly the same throughout the day. Nevertheless, ProEnergy-T has the best performance in the morning, arguably followed by EWMA-T and WCMA-T. Considering the midday hours, ProEnergy-T has the best performance here as well, while other transmittance based estimators slightly outperform the rest. During afternoon and evening hours, after 15 o'clock, transmittance based schemes improve prediction accuracy considerably. In this case, the four transmittance based schemes perform similarly, with Delta-T preforming best.

The results for MAE complement the above observations, by showing that the absolute errors in prediction are low in the morning and evening, and high during midday.

2.9.2.3 Precision for Various Interval Lengths

We complete our analysis of precision with the effect the prediction interval length makes.

Setup. Figure 2.9 shows the MAPE and MAE precisions for every estimator at location SE. As in the first experiment, night time and low light time-slots are omitted, which are time-slots in which the harvested energy is less than $10\,\%$ of the daily maximum. Also as before, separate training and evaluation was conducted for the MAPE and MAE comparison.

So far, all precision metrics refer to the next hour's estimate, $\tilde{p}_{t-1}(t)$ for $\Delta t = 1\,\mathrm{h}$. In this experiment, we evaluate additionally the precision for time intervals of lengths $\Delta t = 2\,\mathrm{h}$ and $\Delta t = 4\,\mathrm{h}$. Functionally, the latter two are the same as $\tilde{p}_{t-1}(t) + \tilde{p}_{t-1}(t+1)$ and $\tilde{p}_{t-1}(t) + \tilde{p}_{t-1}(t+1) + \tilde{p}_{t-1}(t+2) + \tilde{p}_{t-1}(t+3)$ for $\Delta t = 1\,\mathrm{h}$.

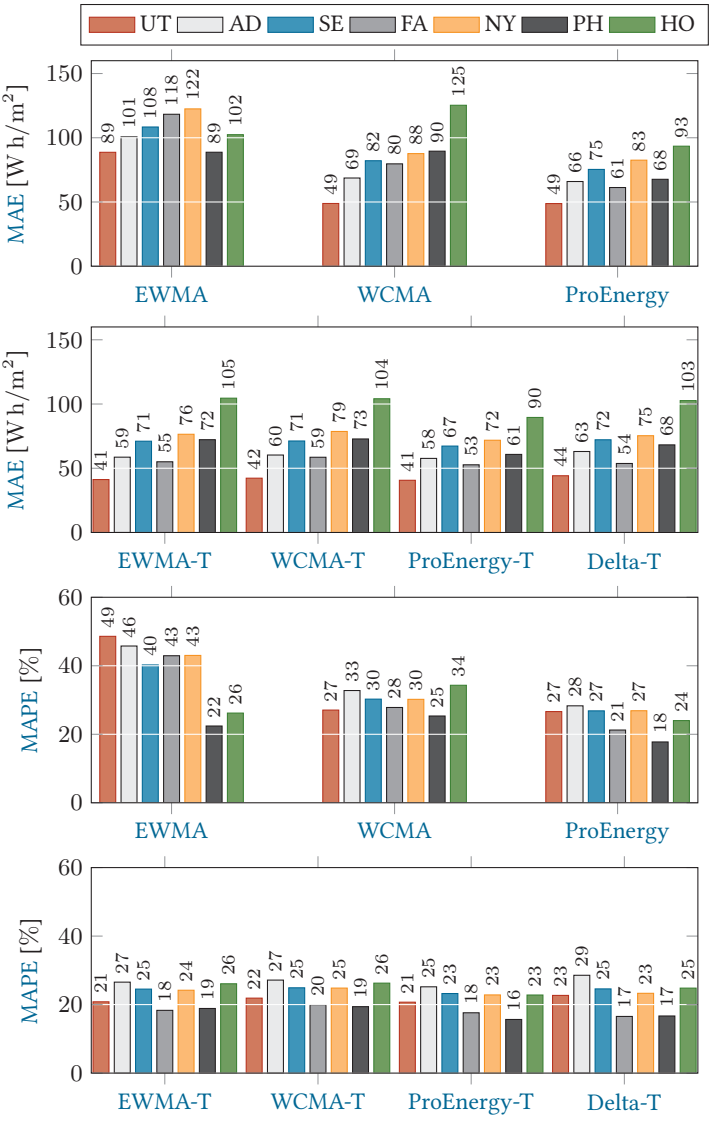


Figure 2.7: Precision of existing and proposed solar energy estimators, for 7 diverse locations

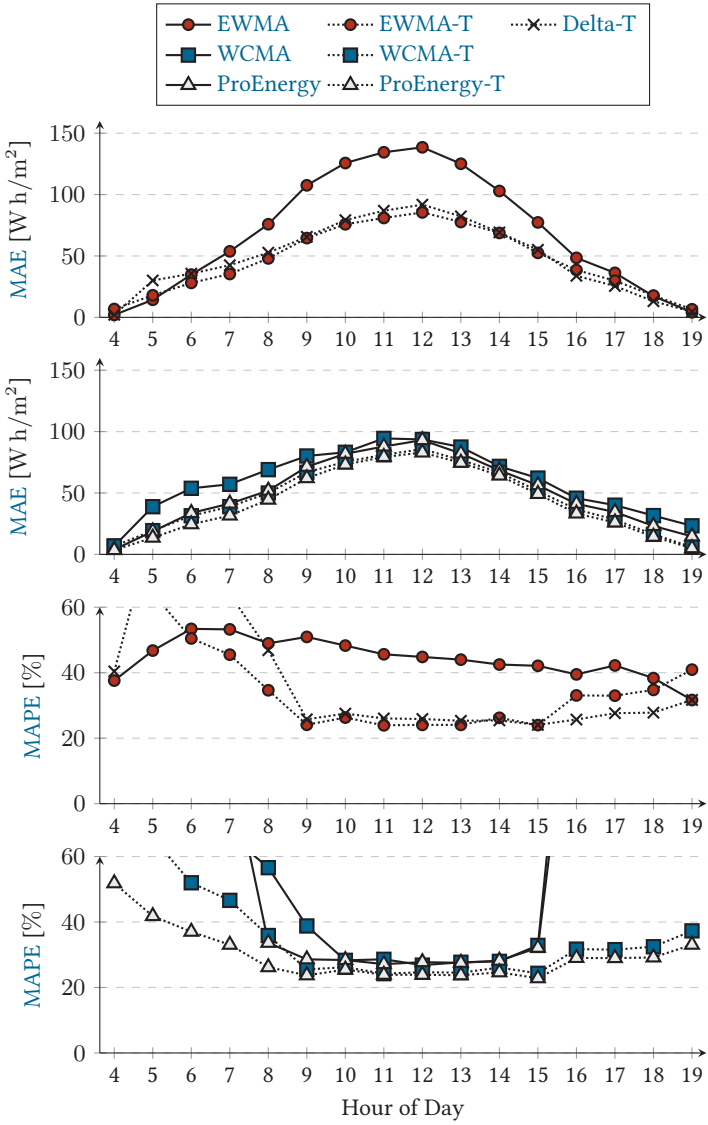


Figure 2.8: Precision of existing and proposed solar energy estimators, for each daylight hour of the day

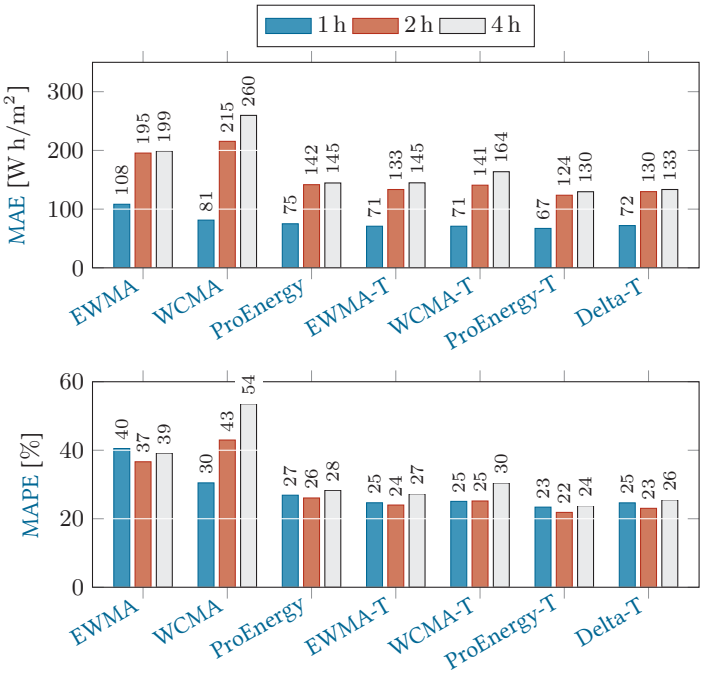


Figure 2.9: Precision of existing and proposed solar energy estimators, for various prediction intervals

Evaluation. The MAPE varies little in each scheme, except for WCMA which is less precise for longer prediction intervals. The MAE, however, is roughly double when two or four hour prediction intervals are used, as opposed to one hour intervals. This is primarily because more energy is harvested in larger intervals, though due to the diurnal cycle the increase is not linear.

2.9.2.4 Precision Versus Complexity

Finally, we end this section by finding Pareto-optimal solar energy estimators, when prediction accuracy and estimator complexity form the design space.

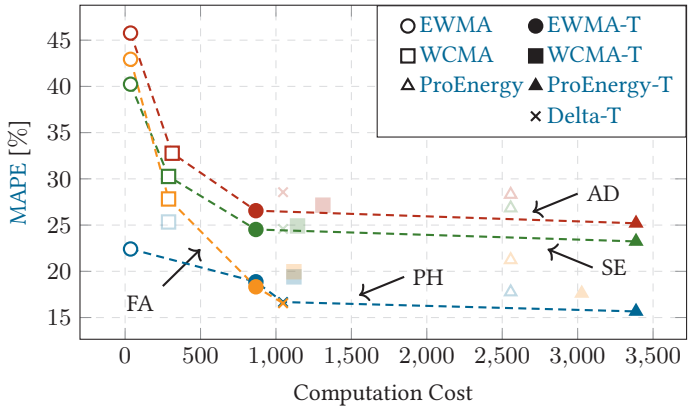


Figure 2.10: Pareto plot showing MAPE and computation complexity of existing and proposed solar energy estimators

Setup. To give complexity measures, we assume that complex floating point operations o_{T} (trigonometric operations, division) take 15 time units, while basic arithmetic operations o_{A} (addition, subtraction, multiplication) take 1 time unit. MAPE values are taken from previous measurements, for locations AD, SE, FA and PH.

Figure 2.10 plots the MAPE and computation cost for predicting twelve one-hour time-slots in a single day. In the figure, all schemes that are Pareto dominated for a given location are dimmed. The Pareto fronts for all locations are illustrated by dashed lines. Note that some schemes do not have the same computation cost at all locations, as location-dependent optimization parameters may impact them.

Evaluation. First of all, Figure 2.10 shows that EWMA has the minimum computation cost for all locations. However, it has significantly higher MAPE compared to other schemes, with the exception of location PH. Next, it should be noted that the new transmittance based schemes (EWMA-T, Delta-T) have a smaller MAPE than the most accurate existing scheme ProEnergy, at a reduced computation cost. Finally, ProEnergy-T is the most accurate scheme (with the

exception of FA where Delta-T outperforms it). However, the accuracy of Delta-T and EWMA-T is comparable to ProEnergy-T, with significantly lower computation cost.

As a final remark, we note that the computation cost for the transmittance based schemes can be significantly lowered if the $p^{\mathrm{et}}(t)$ values of extraterrestrial solar energy are stored in advance for the entire year. This would require a memory cost $24 \cdot 365 = 8760$ read-only floating point numbers, which is feasible for most embedded platforms.

2.9.3 System's Performance Experiments

Here we use simulations to illustrate the difference between the optimal solution to the backup battery problem $u^*(t)$, the finite horizon control (FHC) solution $u^{\mathrm{FHC}}(t)$, and a baseline simple heuristic approach that does not make use of energy predictions $u^{\mathrm{EMX}}(t)$. We investigate a hypothetical system deployed in a remote outdoor area with the goal of operating for multiple years, a scenario similar to Buchli et al. [BKT15], as well as a system deployed indoors. In the first environment, a relatively large rechargeable energy storage is used to reasonably cover a drop of energy in the winter months, the consumption of the system is adjusted once a week, while precise future energy estimators exist due to the limited non-determinism in harvesting accompanying this time scale. In the second case, we use the concrete system implementation introduced in Section 2.8 that operates in an indoor environment. In this more dynamic surrounding, the consumption is updated every 10 minutes, and it is generally harder to predict future energy as it depends on non-deterministic human interference such as switching lights, closing doors that block outside light, and using sun shading devices at windows. These setups are chosen because they are very different in requirements and characteristics.

These experiments illustrate how the choice of base consumption impacts the backup battery usage at two locations, our energy management with a backup battery, and its reactivity to sudden and non-deterministic changes.

Baseline scheme. The simple baseline control algorithm is the purely reactive scheme EnoMax [VGB07], where the system tries to maintain a predefined state of charge of the rechargeable energy storage $\alpha \cdot B_R$. It can be described as follows:

$$\forall t: \; u^{\text{EMX}}(t) = \max(b_B(t) - \alpha \cdot B_R), u_{\mathsf{BASE}}) \tag{2.31}$$

where we used the recommended target of stored energy $\alpha = 0.6$.

Utility. Although any strictly concave function would suffice, the utilization function $U(0,T)$ is chosen as the following normalized function:

$$U(0,T) = \frac{\sum_{i=0}^{T-1} \sqrt{u(t)+1}}{\sum_{i=0}^{T-1} \sqrt{u^*(t)+1}} \tag{2.32}$$

2.9.3.1 Performance at Various Outdoor Locations

For this simulation, we assume a system deployed outdoors, with a one week time step. A total of 7 locations were used, for which we evaluate the system's performance in terms of backup battery usage and utility. These are, ordered from the pole to the equator, AN, PO, BH, BA, LA, EP, and CC.

Setup. A rechargeable energy storage with capacity $B_R = 1\,\text{MJ}$ or about $1650\,\text{mW} \cdot \text{week}$ is used. The solar panel's efficiency is assumed to be $20\,\%$, while its size at locations CC and LA is $10\,\text{cm}^2$, at AN it is $20\,\text{cm}^2$, and otherwise $15\,\text{cm}^2$. These values have been chosen such that the yearly average of the harvested power is around $600\,\text{mW}$. A base consumption of $u_{\mathsf{BASE}} = 320\,\text{kJ} \approx 530\,\text{mW} \cdot \text{week}$ is used. The base consumption is chosen to be slightly lower then the yearly average harvested energy per week.

The first year of data for all locations was used to calculate the EWMA harvested energy estimates for the FHC scheme, while the simulated time frame is the following 11 years, $T = 11 \cdot 52$. FHC's time horizon is one year, $T_H = 52$.

Evaluation. Tables 2.7 and 2.8 summarize the results of this simulated experiment. First, let us interpret results for the total used

Table 2.7: Evaluation of the backup battery use, for 11 years at 7 locations, and the average harvested energy per week

Abbr.	Total Backup Battery Use [MJ]			Average Energy [mW · week]			
	E^*	E^{FHC}	E^{EMX}	$\overline{u_B^*(t)}$	$\overline{u_B^{\text{FHC}}(t)}$	$\overline{u_B^{\text{EMX}}(t)}$	$\overline{p(t)}$
AN	48.10	50.13	52.04	138.66	144.51	150.02	526.62
PO	26.23	28.45	30.24	75.61	82.01	87.17	591.93
BH	18.78	20.55	22.81	54.14	59.24	65.75	624.01
BA	6.06	6.89	10.15	17.47	19.86	29.26	669.48
LA	8.81	10.23	12.90	25.40	29.49	37.19	605.82
EP	0.87	1.64	4.21	2.51	4.73	12.14	531.58
CC	10.56	12.16	14.64	30.44	35.05	42.20	673.94

Table 2.8: Evaluation of the total utility, for 11 years at 7 locations

Abbr.	Total Utility			Abbr.	Total Utility		
	U^*	U^{FHC}	U^{EMX}		U^*	U^{FHC}	U^{EMX}
AN	1.0000	0.9962	0.9395	LA	1.0000	0.9994	0.9625
PO	1.0000	0.9989	0.9386	EP	1.0000	1.0003	0.9979
BH	1.0000	0.9978	0.9377	CC	1.0000	0.9989	0.9436
BA	1.0000	0.9973	0.9359				

backup battery energy E. The optimal use function $u^*(t)$ minimizes the use of the backup battery (Lemma 2.4), and therefore the total use of the backup energy $E^*(0,T)$ is the smallest in this case. The total backup energy use values for FHC $u^{\text{FHC}}(t)$ are close to the optimal values, and they are also better than the baseline $u^{\text{EMX}}(t)$. Remember that the optimal values are not achievable as the harvested energy differs from its estimation.

With finite horizon control, the reduction in used backup energy compared to the reactive scheme EnoMax is between 5 and $10\,\text{mW}$, on average over 11 years. If the optimal solution were implementable, one would save on average from 9 to $12\,\text{mW}$ of backup power.

Now, let us comment on the total utility values. Overall, the total utility values are similar, but is also due to the use of a

square-root utility function which only mildly penalizes high power consumption. The utility of FHC is close to that of the optimal solution, while EnoMax's utility is clearly worse.

Notably, the total utility of the optimal solution $u^*(t)$ is not necessarily the largest value. This is a consequence of the sub-optimality of FHC and EnoMax. Namely, when the total backup use is larger then necessary, then the extra energy may increase the system's total utility.

2.9.3.2 Sensitivity to Base Consumption, Outdoor Locations

For our second experiment, we evaluate the usage of the backup battery in more detail. Namely, we evaluate the impact the base consumption u_{BASE} has on the total used backup energy $E(0,T)$ and the total utility $U(0,T)$ over 11 years of operation.

Setup. We evaluate the backup battery use at two locations, BH and EP. At BH the base consumption u_{BASE} varies from 170 to 570 kJ (280 to 940 $\mathrm{mW} \cdot$ week), while at EP u_{BASE} varies from 250 to 520 kJ (410 to 860 $\mathrm{mW} \cdot$ week).

Two other locations, PO and LA, have been picked for evaluating the impact on the total utility $U(0,T)$. The base consumption u_{BASE} at PO is ranging from 100 to 600 kJ (170 to 990 $\mathrm{mW} \cdot$ week), while at LA it is from 200 to 500 kJ (330 to 830 $\mathrm{mW} \cdot$ week).

As in the previous outdoor setup, the time step is one week. A solar panel of size $15\,\mathrm{cm}^2$ harvests energy with $20\,\%$ efficiency (except for LA where size is $10\,\mathrm{cm}^2$), and a $B_R = 1\,\mathrm{MJ} \approx 1650\,\mathrm{mW} \cdot$ week rechargeable energy storage is used. The FHC algorithm uses an EWMA estimator for predicting future energy, for which the first year of data was used to calculate estimates. The simulated time is the following 11 years, $T = 11 \cdot 52$.

Evaluation. Figures 2.11a and 2.11b show the difference in total backup energy between the optimal solution $u^*(t)$, and the two practically implementable schemes $u^{\mathrm{FHC}}(t)$ (finite horizon control) and $u^{\mathrm{EMX}}(t)$ (optimal reactive scheme under ENO constraints).

Let us start by commenting on the two schemes. With FHC,

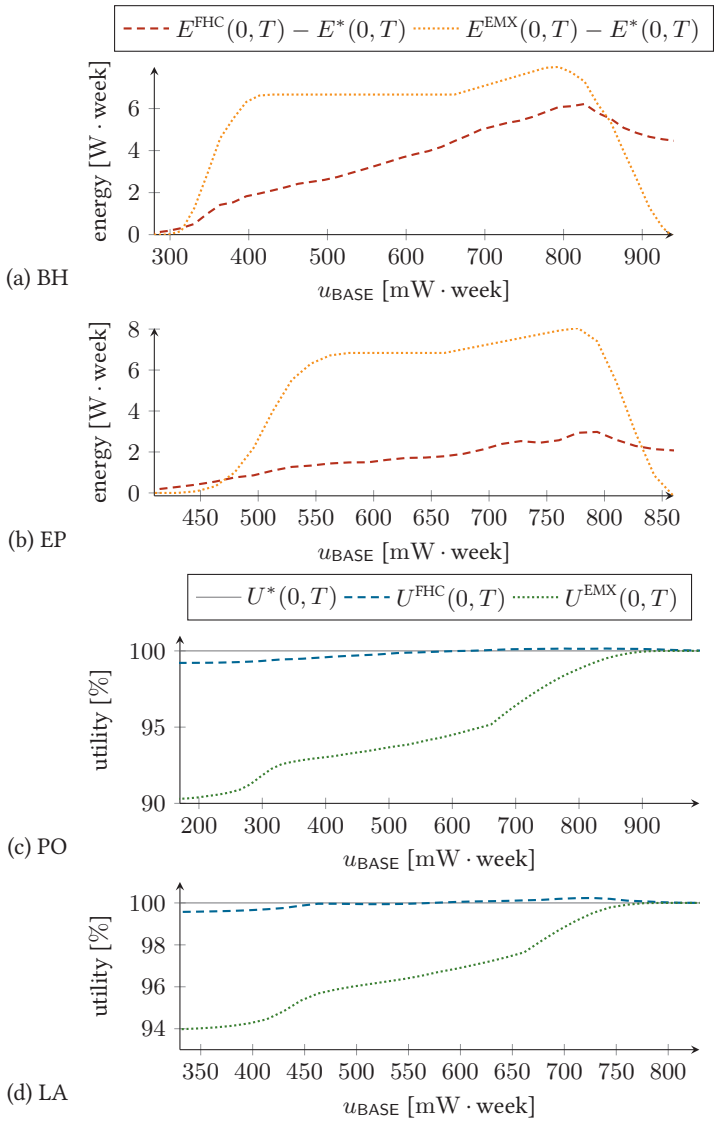


Figure 2.11: Sensitivity of the backup battery use and utility to the base consumption, for several outdoor locations over 11 years

additional energy is consumed from the backup battery in two situations. First, it happens when there is an optimistic miss-prediction, which can not be corrected by lowering the use function due to the $u^{\text{FHC}}(t) \geq u_{\mathsf{BASE}}$ constraint. Second, a pessimistic miss-prediction may lead to wasted energy, if it causes the energy in the rechargeable storage to reach its capacity. In this case, this wasted energy might need to be resupplied from the backup battery some time later.

The baseline solution EnoMax has the goal of maintaining a constant state of charge throughout the year. In the general case, the energy in the rechargeable storage is thus usually higher then optimal during summer weeks, and this may lead to wasted energy. Also, the stored energy is often less than optimal in the beginning of winter, which causes the system to use the backup battery more then it would be necessary otherwise.

Now, we shall observe results shown in Figures 2.11a and 2.11b, and validate our commentary. Starting with small values of the base consumption u_{BASE}, namely less than $350\,\mathrm{mW} \cdot \text{week}$ for location BH and less than $500\,\mathrm{mW} \cdot \text{week}$ for location EP, we can see that all solutions have similar performance as the backup battery is rarely used.

When the base consumption is in the middle range, 350 to $750\,\mathrm{mW} \cdot \text{week}$ for location BH and 500 to $850\,\mathrm{mW} \cdot \text{week}$ for location EP, there is a prominent advantage in favour of the finite horizon control solution.

When the base consumption is large, more than $750\,\mathrm{mW} \cdot \text{week}$ for location BH and $850\,\mathrm{mW} \cdot \text{week}$ for location EP, the backup battery is used every time instance, $u^*(t) = u_{\mathsf{BASE}}$ for all $t \in [0, T)$. Thus, any solution which does not lead to wasted energy performs as good as the optimistic one. FHC is sub-optimal in this case, as the prediction algorithm and its pessimistic miss-predictions during summer weeks lead to some wasted energy.

Finally, we examine how the total utilities of the three use functions compare with varying base consumption. These results are visualized in Figures 2.11c and 2.11d. Remember that the total utilization is a secondary objective, and comes after minimizing the backup battery usage. This means that the optimal solution $U^*(0, T) = 1$ does not have the maximal utilization among all

use functions that comply with the constraints defined by the backup battery optimization problem (2.6). This can be observed in both figures for several base consumption values, when the total utilization of the finite horizon $U^{\text{FHC}}(0, T)$ is larger than 1. This is because a miss-prediction of EWMA causes additional energy to be drawn from the backup battery, which contributes to the total utilization, ultimately making it larger.

We can see in the figures that the overall differences in total utilization for the presented solutions are small, as the used utilization slightly penalizes large energy consumptions. Nevertheless, the total utilization for FHC is closer to the optimal solution's value than the baseline EnoMax. These differences are shown to decrease as the base consumption increases, and as the backup battery ends up being used more and more. Ultimately, for a large enough base consumption u_{BASE}, all solutions become simply $u(t) = u_{\mathsf{BASE}}$.

2.9.3.3 Sensitivity to Base Consumption, Indoor Locations

For our third experiment, we visit two indoor locations and repeat the former sensitivity analysis. As in the outdoor scenario, we evaluate the usage of the backup battery, as well as the impact on the total utility.

Even though both indoor locations are in the same room, the harvesting characteristics are fundamentally different. For the first location, L16, the harvester is placed on a table facing up, without many obstacles surrounding it. In this environment, the prediction of the future energy made by EWMA is reasonably precise. For the second location, L17, the harvester is mounted on a wall facing west, and near a window. Close by, a curtain sometimes partly obstructs the window. Here the EWMA predictor showed a large deviation between predicted and harvested energies, mainly because it could not predict the movement of the curtain.

Setup. We used the system implementation as introduced in Section 2.8, however a smaller rechargeable energy storage with capacity $B_R = 1.35\,\text{J} = 2.25\,\text{mW} \cdot 10\,\text{min}$ was assumed, in order to adapt to the smaller amount of harvested energy. One day is used to calculate the harvested energy estimates, while the simulated

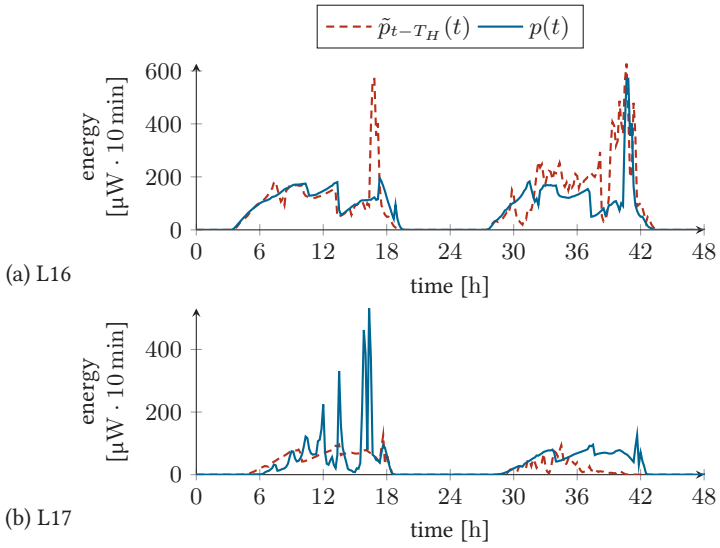


Figure 2.12: The EWMA estimate of future energy, used for FHC, and the harvested energy, over two days at locations L16 and L17

time frame is 12 days, each with 144 10-minute intervals, $T = 12 \cdot 144$. The FHC's horizon length is one hour $T_H = 6$. The base consumption is varied from 0 to $0.15\,\mathrm{mW} \cdot 10\,\mathrm{min}$ at L16, and to $0.10\,\mathrm{mW} \cdot 10\,\mathrm{min}$ at L17.

Evaluation. Figures 2.13a and 2.13b visualize the difference in total backup energy between the optimal solution, and the two implemented algorithms FHC and EnoMax, while Figures 2.13c and 2.13d show how the total utilities of the three use functions compare. Figures 2.12a and 2.12b illustrate the EWMA estimator of harvested energy. At L16, the predictor achieves realistic predictions, while at L17 it can not sufficiently deal with the very dynamic and non-deterministic environment.

For location L16, where EWMA prediction is good, we see there is a prominent benefit in using FHC compared to the baseline solution, for low and medium values of the base consumption

$u_{\mathsf{BASE}} < 100\,\mu\mathrm{W} \cdot 10\,\mathrm{min}$. When the u_{BASE} is larger than that value, the backup battery is used every time instance as $u(t) = u_{\mathsf{BASE}}$ for all $t \in [0, T)$. Thus, any solution which does not lead to wasted energy performs as good as the optimal one. FHC is sub-optimal in this case, as the prediction algorithm and its pessimistic miss-predictions lead to some wasted energy.

At location L17, finite horizon control under-performs the baseline. This is due to the almost useless harvested energy estimates from EWMA. During the simulated days, there are both significant over- and under-estimations of future energy. Without a viable future energy predictor, we can conclude that it does not make sense to use any prediction-based scheduler, and a reactive power management solution works better.

In summary, FHC for systems with a backup battery performs well if the energy estimation is sufficiently precise, otherwise a simple reactive approach should be used.

2.10 Real World Experimental Results

In this section, we further validate the realism of our non-ideal model. To this end, we deployed the energy harvesting embedded system, introduced in Section 2.8, in a controlled laboratory environment. We compare the simulated and measured harvested energy, consumed energy, and super-capacitor's charge. For comparison, we use the L2 norm of the difference between respective emulated and measured traces. We thus close the loop on the abstract model, considering non-ideal behavior, component characterization and measured system behavior under realistic conditions.

Setup. Environment. Using the solar testbed [Sig20], we closely recreate three days of indoor illumination conditions, as recorded at location L14 of the Indoor Solar Harvesting database [SGT19] on 29 - 31 May 2018. The recreated trace features significant natural light, especially during morning hours. The solar test-bed enables this recreation, because it encapsulates the a photovoltaic panel and exposes it to desired illumination levels.

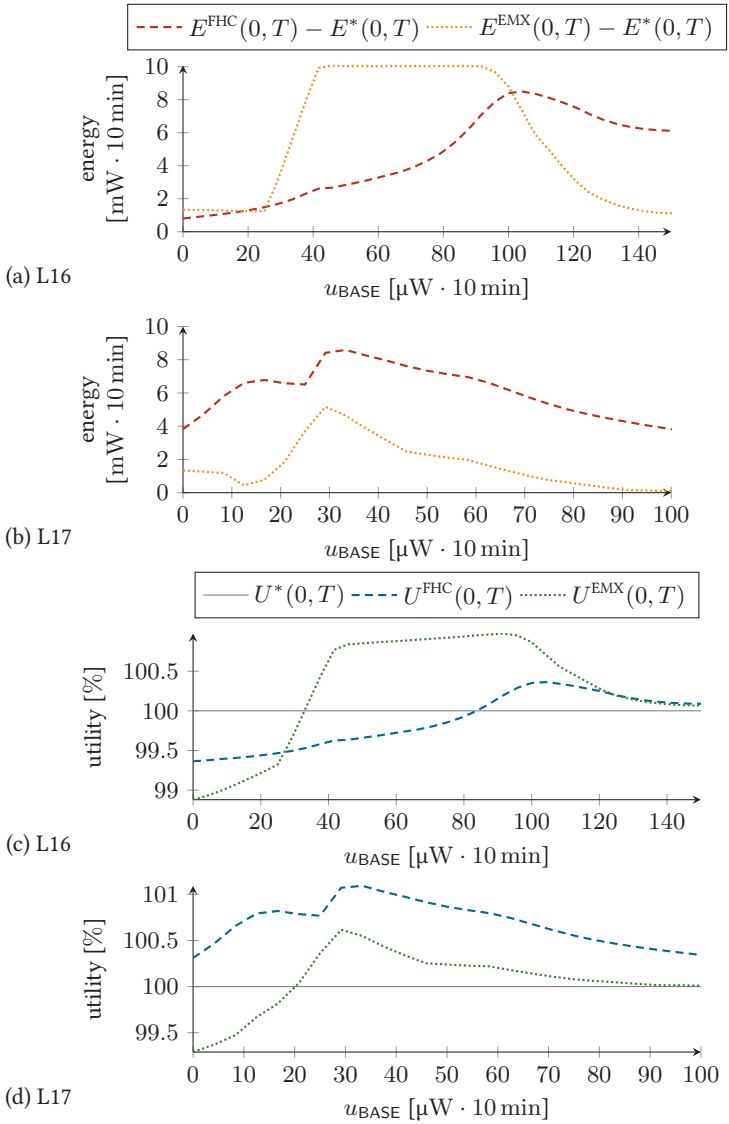


Figure 2.13: Sensitivity of the backup battery use and utility to the base consumption, for several indoor locations over 12 days

The three-day trace was slightly adjusted for practical purposes. First, the trace was sped up by a factor of 5 times, so the experiment lasted 14.4 hours. Then, the recorded trace was divided into two-minute intervals, over which the illumination level was averaged, i.e., the illumination level was updated every two minutes. In order for the experiment to be more dynamic, such that we record the rechargeable energy storage getting fully charged and discharged, we scaled the illumination levels by a factor of 8 times. Functionally, this is equivalent to having an 8 times larger photovoltaic panel.

Practical system. The energy harvesting hardware and software as introduced in Section 2.8.1 is used as the real world system. The 'power manager' updates the 'adaptive consumer' task every two minutes. The used consumption trace was determined by the finite horizon control (FHC) scheme with a non-perfect future energy estimate. The consumption varies from $2.01\,\mathrm{mW}$ ($2.01\,\mathrm{mW} \cdot 120\,\mathrm{s} = u_{\mathsf{BASE}}$) to $17.64\,\mathrm{mW}$. Current and voltage measurements are done using a RocketLogger [SGL+17], which measures in parallel: the harvested power, the rechargeable energy storage's voltage, and the system's power dissipation.

Simulated system. To compare our implemented hardware-software system with model-based simulation results, we made use of the non-ideal model introduced in Section 2.7, and considered the coefficients obtained by the component characterization as described in Section 2.8.2: $\sigma_p = 0.8529$ and $\sigma_R = 0.9269$. We applied the same trace of harvested energy and consumption, and emulated the rechargeable energy storage's charge.

Evaluation. Figure 2.14 shows the measured and simulated values side by side: the harvested and consumed power, and the charge of the super-capacitor. Energy values in Joules are obtained by multiplying the corresponding power with the $120\,\mathrm{s}$ time-interval length. The harvested energy's average L2 difference between measured and emulated results is $12.51\,\mathrm{mJ}$, which translates to a difference of $104.24\,\mu\mathrm{W}$ in power. This difference is $1.16\,\%$ of the average measured power. The L2 norm of the difference between the

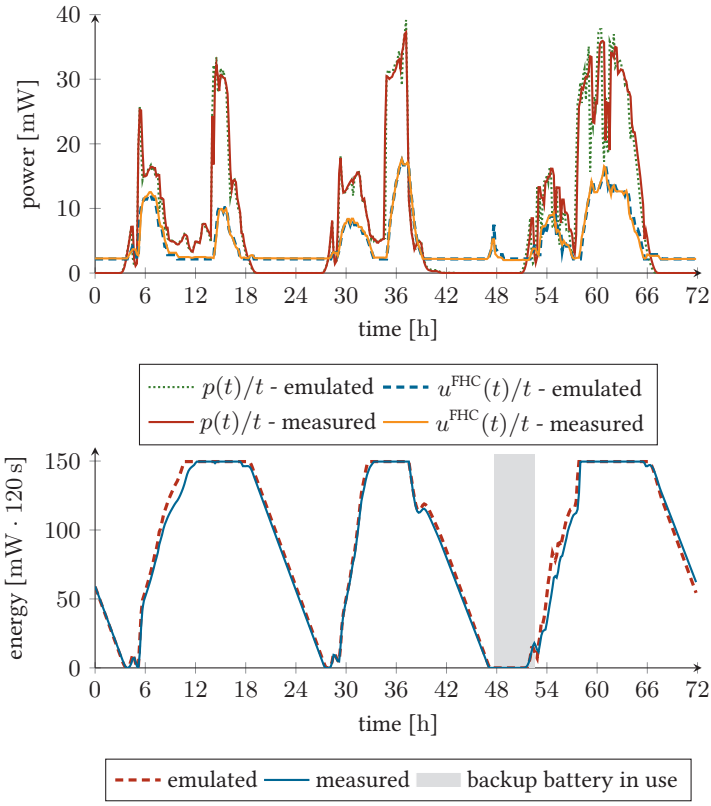


Figure 2.14: Comparison between the real-world embedded system and emulation results, over three days at indoor location L14

measured and emulated consumed energy is $2.75\,\mathrm{mJ}$ on average, corresponding to a $22.88\,\mu\mathrm{W}$ difference in power. This constitutes a $0.47\,\%$ difference to the average measured consumption.

Finally, the measured rechargeable energy storage is found to behave similarly to its emulated counterpart as well. The L2 difference between the two traces is $30.18\,\mathrm{mJ}$ or $251.5\,\mu\mathrm{W}\cdot 120\,\mathrm{s}$, which is $0.17\,\%$ of the super-capacitor's maximal charge. Note that the backup battery was used once during this experiment, in the night between the second and third days. In summary, based on this comparison we can strongly corroborate the practical applicability of the proposed models and energy management algorithms. The differences between the abstract model and the considered non-ideal system behavior, and the actual implementation and measurements are negligible.

2.11 Summary

This chapter is concerned with a worst-case analysis of energy harvesting systems. With the uncertainty of the harvesting environment artificially removed, we provide an analytic optimal solution for energy management. Using a backup battery, uninterrupted operation under energy-neutral operation constraints is ensured, a minimal service level is guaranteed, and both the lifetime and the long-term utilization are maximized. To date, no energy control algorithm is known to consider such energy management subsystems that make use of a backup battery.

In practice, the environment's uncertainness can not be disregarded, and the optimal solution can only be approximated. Proactive energy management strategies such as finite horizon control aim to realize just that, by using estimators of future harvestable energy. Regarding such estimators, we present new mechanisms for predicting solar energy that exploit the extraterrestrial solar model. This model yields significant improvements in prediction accuracy (ProEnergy-T has a $14.5\,\%$ improvement in relative accuracy compared to existing schemes), at a small additional computation cost. We also propose computationally light-weight estimators (EWMA-T, Delta-T) with reasonable prediction accuracy.

Extensive trace-based simulation results as well as experiments with an implemented hardware solution are used to quantify and model non-ideal behavior, and to validate the usefulness of our approach in terms of guarantees, lifetime and run-time overhead.

3

Stochastic Analysis of Energy Harvesting Systems

Now that we have a formal model of energy harvesting embedded systems, as well as a way to obtain their worst-case performance metrics, we move on to the stochastic analysis. While designing or evaluating energy management strategies targeting long-term operation, one finds probabilistic performance metrics useful: the probability of failure, the expected energy consumption and energy storage level, and robustness to design parameters or changing environments. This chapter proposes a stochastic modeling technique and corresponding analysis, which is able to provide precisely these metrics.

Our developed analysis is based on Markov chains. By modeling harvested energy with random variables, which is possible in many harvesting scenarios due to the availability of extensive datasets, a range of energy management policies is analyzed. We also consider the variability in actual energy consumption when a given service is required, which stems from the unpredictable timing behavior of tasks. Based on the analysis results, we propose a new adaptive energy management strategy inspired by the class of mixed-criticality systems. With this strategy, the system may degrade or

drop several less important tasks in real time, in order to ensure a level of operation even in adverse conditions. This new approach is compared to a number of existing alternatives.

In the experimental section, we show the usefulness of our analysis method and the novel energy management strategy by solving several optimization problems regarding adjustment of the management policy and choosing design parameters to fit the desired use case and environment. Extensive simulation results are conducted for both indoor and outdoor environments, where our own management strategy is found to match or outperform the state of the art.

3.1 Introduction

Previously, we have seen how recent technological advances have made the internet of things, wireless sensor networks (WSNs) and cyber-physical systems considerably more pervasive. Energy harvesting has emerged as a way to power these systems indefinitely, particularly because of the low maintenance involved, the scalability, the potential mobility, and ability to deploy systems in remote and hard to reach areas. But a paramount challenge in deploying these energy harvesting systems is ensuring a level of reliability and determinism in their operation [BASM16], due to the volatility and intermittence of harvested energy. Yet for many safety-critical applications there are few alternatives to energy harvesting, as WSN-based landslide [AAA+07] and natural hazard [MFCP+19] early warning systems illustrate.

Even though recent studies discuss reliability in long-term deployments [BSBT14b, GJKZ19], we notice that a formal analysis of energy management schemes with a focus on probabilistic metrics is missing. Metrics such as the probability of failure, the expected energy consumption and energy storage level, and robustness or sensitivity to design parameters or changing environments, all are critical for characterizing the utility and reliability of energy harvesting embedded systems. This chapter aims to fill this gap by defining a stochastic model and formalizing a Markov analysis. Our

stochastic model is able to incorporate the natural uncertainness of harvested energy with a random variable that changes periodically, as well as the variability in consumed energy. Furthermore, the developed analysis is able to evaluate many state of the art energy management strategies, and can be used to construct novel ones.

As an example scenario, we focus on solar energy harvesting embedded systems, both in outdoor and indoor long-term deployments. Concretely, we propose a heuristic energy management strategy inspired by the class of mixed-criticality systems, which enables the system to reduce performance over periods with low energy availability, by dropping or reducing non-essential tasks in real time. Additionally, we propose a mechanism for exploiting high surpluses of harvested energy, which in this case occur over summer, by having an enhanced service level then.

Strictly speaking, in this chapter we are interested in developing a statistics-based analysis method, and in using it to aid the construction of energy management policies, and to enable design space exploration. We thus formulate the following problems: Given a system which receives its energy from a harvesting source whose statistical properties are known through historical data. Given a known random variable describing the energy consumption of each possible mode of operation. Given an energy management subsystem which controls the power flow between the energy harvester, consumer, and rechargeable energy storage.

(A) Define a stochastic system model that takes into account relevant design variables: the harvester's dimension and the capacity of the rechargeable energy storage.
(B) Formalize an analysis method that provides metrics such as the probability of failure due to lack of available energy, and the probability that the system consumes a certain energy value over a given interval.
(C) Design a good energy management policy, which aims to minimize the probability of failure, while maximizing the long-term utility.
(D) Compare different energy management policies.

As in the last chapter, non-ideal behavior of a real-world energy management subsystem should also be considered, with phenom-

ena such as energy storage leakage or charging efficiency.

Responding to these problems, our work presents the following contributions.

1. We propose a stochastic model and corresponding Markov analysis for energy management strategies targeting long-term operation. The analysis can provide several key metrics such as: the steady state probability of failure, i.e., rechargeable storage depletion, the steady state stored energy levels, the expected energy use and corresponding system utility, and others.
2. The stochastic model allows for adaptive energy consumption. We especially focus on analyzing an emergency mode of operation, where the energy consumption is reduced if the stored energy level is low. We show by simulation the advantage of adaptive management policies in terms of depletion probability and robustness.
3. Furthermore, the stochastic model allows modeling of several artifacts of actual implementations such as energy storage level-dependent leakage, and charging and discharging efficiencies.
4. For the solar energy harvesting use case, we perform extensive trace-based simulations, outdoors for several geographical locations, and indoors for diversely lighted rooms.
5. We demonstrate solving the system dimensioning problem, i.e., given a required energy use, and an acceptable tolerance on the probability of failure, we determine the set of feasible pairs of solar panels and energy storage capacities.

Organization of this chapter. First, Section 3.2 defines the novel stochastic system model, as an answer to problem (A). In Section 3.3 we define our Markov analysis for solving the long-term power management problem, and also introduce an adaptive energy management scheme. This section provides answers to problems (B) and (C). Then, in Section 3.4, we sketch how the system model and analysis can be extended to include non-ideal behavior present in practical energy management implementations. Finally, Section 3.5 provides trace-based simulations, illustrating major results of the

paper. Particularly, it illustrates a design space exploration, and a compares our own energy management scheme with the state of the art, answering problem (D).

3.2 System Model

In this section, we start off by defining the used models: a deterministic model is presented first in Section 3.2.1, before we introduce our stochastic model in Section 3.2.2. We also deal with discretization of both values and time, and prove that the stochastic model is pessimistic with regards to these operations. This allows us to base the statistical analysis in the following section on this stochastic model.

Here we present idealized models, while in Section 3.4 we comment on extensions that include non-ideal behavior of a physical implementation, namely lossy energy storage and charging as well as discharging efficiencies.

Notation. We use the following notations: Bold characters represent vectors and matrices, while non-bold characters represent scalars. Subscripts are used to reference individual elements of matrices or vectors, e.g., $H_{k,l}$ denotes the element in k^{th} row and l^{th} column of matrix $\boldsymbol{H}$, and T_i denotes the i^{th} element of vector $\boldsymbol{T}$. We start numbering rows and columns from 1.

3.2.1 Deterministic System Model

We start with a system model in continuous time. Next, we introduce a corresponding discretized model in time and values, and prove that it is pessimistic in comparison to the continuous model.

3.2.1.1 Continuous System Model

Let us use $\tilde{\cdot}$ to denote variables defined in continuous time. The rechargeable energy storage has stored energy $\tilde{b}(t)$ at time $t \in \mathbb{R}_{\geq 0}$,

and a maximum capacity of $\hat{b}$. The stored energy is governed by the following equation:

$$\frac{\partial \tilde{b}(t)}{\partial t} = \begin{cases} \tilde{p}_h(t) - \tilde{p}_u(t) & \text{if } \left(\tilde{b}(t) < \hat{b} \wedge \tilde{p}_h(t) > \tilde{p}_u(t)\right) \vee \\ & \quad \left(\tilde{b}(t) > 0 \wedge \tilde{p}_h(t) < \tilde{p}_u(t)\right) \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

Note that the second case defining $\partial\tilde{b}(t)/\partial t$ to be 0 ensures that the stored energy is always between the maximum capacity ($\hat{b}$) and none (0).

At time t, the harvested power is $\tilde{p}_h(t)$ and the consumed power is $\tilde{p}_u(t)$, which is also denoted as the used power. Moreover, the system enters a failure state for all t where the storage is empty $\tilde{b}(t) = 0$ and there is a negative power balance $\tilde{p}_h(t) - \tilde{p}_u(t) < 0$. In other words, the system fails if the intended power consumption $\tilde{p}_u(t)$ can neither be provided by the storage nor the energy harvester.

3.2.1.2 Time-discrete System Model

The stochastic Markov analysis technique, covered in Section 3.3, uses a discrete time model that follows the update interval of the considered energy management system. But, power is a continuous variable in time in the physical domain. Thus the goal now is to determine a discrete-time version of (3.1) that provides worst-case results.

We define the unit time interval δ, which corresponds to the update interval of the energy management. Discrete time n maps to its continuous counterpart via $t = n \cdot \delta$. The main challenge in the discretization in time is due to the fact that we only know the relevant quantities like harvested energy and used energy at $t = n \cdot \delta$, but the model in (3.1) needs a continuous knowledge of the power values. In order to provide reasonable bounds, we make the following assumption on the used power:

- The used power within an interval of length δ is constant, and therefore $\tilde{p}_u(t)$ is piece-wise constant.

This assumption is justified as the used power is changed based on the commands of the energy management only at discrete time steps. In terms of harvested energy, we differentiate between two cases:

- Suppose that we harvest a certain energy within an update interval.
  - Then $\tilde{p}_h(t)$ can not change arbitrarily in this time interval, but just within provided bounds.
  - Then $\tilde{p}_h(t)$ can change arbitrarily in this time interval.

We find the first case more interesting and practical. For example, if a certain energy is harvested by a solar cell within a week, it is not realistic to assume that the whole energy is provided in the last hour of that week. It is more appropriate to assume an upper and a lower bound of the power that can be harvested at any time. As well as this, the second case can be seen as a limit of the first one. Therefore, we now proceed with the assumption that bounds on the harvested energy apply, while later we return and comment on the arbitrary harvesting case.

The assumptions on harvested and consumed power lead to the following relations and definitions. The stored energy at time instance n is denoted as $e_b(n)$, and the harvested and used energies within the time intervals are defined as:

$$e_h(n) = \int_{n\cdot\delta}^{(n+1)\cdot\delta} \tilde{p}_h(t)dt \quad , \quad e_u(n) = \int_{n\cdot\delta}^{(n+1)\cdot\delta} \tilde{p}_u(t)dt \tag{3.2}$$

for all discrete time instances $n \in \mathbb{Z}_{\geq 0}$, see Figure 3.1a. Following the assumptions described above, we have also:

$$\begin{aligned} &\forall n\cdot\delta \leq t < (n+1)\cdot\delta \ : \\ &\quad \tilde{p}_u(t) = \frac{e_u(n)}{\delta} \quad , \quad \check{\sigma}\cdot\frac{e_h(n)}{\delta} \leq \tilde{p}_h(t) \leq \hat{\sigma}\cdot\frac{e_h(n)}{\delta} \end{aligned} \tag{3.3}$$

In other words, the harvested power $\tilde{p}_h(t)$ can not arbitrarily deviate from the average power $e_h(n)/\delta$ in the interval of length δ, but just by a factor within the interval $[\check{\sigma}, \hat{\sigma}]$ with $0 \leq \check{\sigma} < \hat{\sigma}$. Note that determining estimations of the harvested energies $e_h(n)$ and

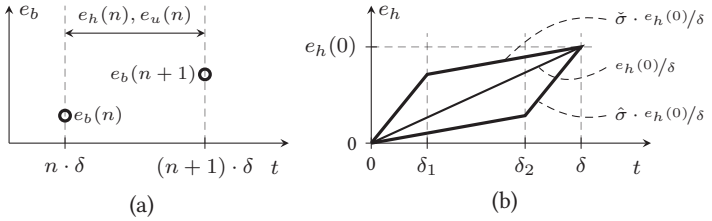


Figure 3.1: Graphical representations of: (a) the discrete model with stored energy $e_b(n)$ at time $n \cdot \delta$, and harvested and used energy $e_h(n)$, $e_u(n)$ in $(n \cdot \delta, (n+1) \cdot \delta]$. (b) the upper and lower harvested energy bounds within a single time interval $n = 0$

the upper and lower deviation factors $[\check{\sigma}, \hat{\sigma}]$ is done by analyzing traces of harvested energy, as shall be illustrated in later sections.

The reader recollects, the purpose of this section is to determine a worst-case discrete time abstraction of (3.1). To this end, we first require that $e_b(n) \leq \tilde{b}(n \cdot \delta)$ for all $n \in \mathbb{Z}_{\geq 0}$. Second, we require that if the system described by (3.1) is in a failure state at any point in the interval $(n \cdot \delta, (n+1) \cdot \delta]$, then the discrete model needs to be in a failure state at n as well.

We first look at a single interval of length δ and then derive the discrete time abstraction, see also Figure 3.1. The constraints in (3.2) and (3.3) can be used to determine upper and lower envelopes of the feasible harvested energy within a time interval. In particular, using the slope constraints on the harvested power $\tilde{p}_h(t)$, one determines:

$$\delta_1 = \delta \cdot \frac{1 - \check{\sigma}}{\hat{\sigma} - \check{\sigma}} \quad , \quad \delta_2 = \delta \cdot \frac{\hat{\sigma} - 1}{\hat{\sigma} - \check{\sigma}} \tag{3.4}$$

Looking at (3.1), one can observe that in general we find $e_b(1) = e_b(0) + e_h(0) - e_u(0)$ as the energy in the storage is increased by harvesting and decreased by consumption. But there are several exceptions to this simple relation, namely the charge is constraint by the maximal capacity $0 \leq e_b(1) \leq \hat{b}$. Additionally, there is an energy overflow and energy will be lost if $\tilde{b}(t) = \hat{b}$ and $\tilde{p}_h(t) > \tilde{p}_u(t)$, and likewise there is a failure state if $\tilde{b}(t) = 0$ and $\tilde{p}_h(t) < \tilde{p}_u(t)$. Therefore, we can distinguish between the following three cases.

- *Failure State*: The system is in a failure state if the energy in the storage attempts to drop below 0. The worst case clearly happens for the lower envelope in Figure 3.1b, and therefore if either condition $e_b(0)+\delta_2(\check{\sigma}\frac{e_h(0)}{\delta}-\frac{e_u(0)}{\delta})<0$ or condition $e_b(0)+e_h(0)-e_u(0)<0$ holds. The first condition corresponds to a violation at time δ_2, and the second at time δ. As a pessimistic simplification, we define that in case of a failure we have $e_b(1)=0$.
- *Intermediate Energy Overflow*: Energy overflow may happen at or before time δ_1, and then not anymore until the end of the interval. Therefore, if condition $e_b(0)+\delta_1(\hat{\sigma}\frac{e_h(0)}{\delta}-\frac{e_u(0)}{\delta})>\hat{b}$ is satisfied, we find

  $$e_b(1)=\hat{b}+\min\{0,(\delta-\delta_1)(\check{\sigma}\frac{e_h(0)}{\delta}-\frac{e_u(0)}{\delta})\}$$

  The minimum-operator takes care of a possible energy overflow of the storage.
- *Normal Case*: In all other cases, we find that

  $$e_b(1)=\min\{\hat{b},e_b(0)+e_h(0)-e_u(0)\}$$

  holds, as again the minimum operator takes care of a possible energy overflow of the rechargeable storage.

Note that for the above relations to hold, we suppose that we can not have an overflow and a underflow of the storage happening within the same interval, i.e., $e_h(0)\leq\hat{b}$.

Combining all of these considerations and applying them to every time interval, we can express the worst-case time-discrete version of (3.1) as follows.

$$e_b(n+1)=\begin{cases}0 & \text{if } e_b(n)+\delta_2(\check{\sigma}\frac{e_h(n)}{\delta}-\frac{e_u(n)}{\delta})<0\ \vee \\ & \quad e_b(n)+e_h(n)-e_u(n)<0\\ \hat{b}+\min\{0,(\delta-\delta_1)(\check{\sigma}\frac{e_h(n)}{\delta}-\frac{e_u(n)}{\delta})\} & \\ \quad \text{if } e_b(n)+\delta_1(\hat{\sigma}\frac{e_h(n)}{\delta}-\frac{e_u(n)}{\delta})>\hat{b} & \\ \min\{\hat{b},e_b(n)+e_h(n)-e_u(n)\} & \\ \quad \text{otherwise} & \end{cases} \tag{3.5}$$

for all discrete time instances $n\in\mathbb{Z}_{\geq 0}$. The first case also indicates a potential failure in the time interval. Most importantly, the above considerations also prove the following lemma.

Lemma 3.1: *(3.5) is a pessimistic time-discretization of (3.1) under the assumptions of (3.2) and (3.3).*

Without bounds on harvested power. We now return to the limit case when no bounds on harvested power can be made. In this case, (3.4) degenerates to:

$$\delta_1 = 0 \quad , \quad \delta_2 = \delta \tag{3.6}$$

Applying this into the failure and intermediate overflow cases, and taking into account the limit values for $\check{\sigma}$ and $\hat{\sigma}$ as well, we see that (3.5) simplifies:

- *Failure State*: A failure may happen within an update interval if condition $e_b(0) - e_u(0) < 0$ holds.
- *Intermediate Energy Overflow*: If $e_b(0) + e_h(0) > \hat{b}$ is satisfied, then harvested energy may be lost due to an overflow during an update interval, and we find $e_b(1) = \hat{b} - e_u(0)$.

The normal case is the same as before.

3.2.1.3 Value-discrete System Model

To conclude the deterministic system model, let us focus on the discretization of values. Clearly, (3.5) is monotone increasing for $e_h(n)$, and monotone decreasing for $e_u(n)$. Therefore, using $\lfloor e_h(n) \rfloor$ instead of $e_h(n)$, $\lfloor e_b(n) \rfloor$ instead of $e_b(n)$, and $\lceil e_u(n) \rceil$ instead of $e_u(n)$ in (3.5) produces pessimistic results. As a consequence, without losing pessimism we can from now on safely assume that n, $\hat{b}$, $e_b(n)$, $e_u(n)$, and $e_h(n)$ are discrete values from $\mathbb{Z}_{\geq 0}$.

3.2.2 Stochastic System Model

The analysis introduced in following sections is based on a discrete Markov model. To this end, we model the harvested energy and used energy with random variables. As a consequence, the energy in the rechargeable storage at some time n, or the occurrence of a failure state, are modeled with random variables as well.

3.2.2.1 Assumptions

We assume that random variables describing harvested energies at times n are known and statistically independent. The later assumption needs to be validated for each harvesting scenario. For example, it is known that this holds for outdoor solar radiation if the length of the updated interval δ is a week, see for example the comment on atmospheric circulation systems by Foken [Fok17], page 5. Otherwise, if the time interval is short, i.e., shorter than 3 to 6 days, temporal weather patterns cause a potential correlation between consecutive intervals, because consecutive days have similar weather. Note that we model the time *between* updates using a worst-case scenario according to (3.3) using upper and lower bounds on the deviation from a constant harvesting power, Therefore, we consider any possible statistical dependency *between* updates.

In case of periodic harvesting patterns, one can use different random variables within such a period and then repeat the aggregated model periodically. For example, in the case of solar energy we can model each week of the year with a separate random variable, and iterate the corresponding yearly Markov model.

The used energy is assumed to be modeled with independent random variables as well. A single random variable models each mode of operation, or equivalently, the used energy level. We also assume that the mode of operation is a function of only the current rechargeable energy storage level, and the current time.

3.2.2.2 Harvested Energy

We construct random variables for modeling harvested energy during each time interval based on known data. Specifically in our use case, we use historical information of solar energy, and construct statistical models using maximum likelihood estimation. The analysis presented in this chapter works with any statistical model of harvested energy that can be obtained by matching a distribution to the historical data of harvested energy. We have empirically observed that the normal distribution provides a close match with the historical data set in case of outdoor solar harvesting data, while a different class of distributions is used to model indoor solar

harvesting data. This is all described further in Section 3.5.1.

If $e_h(n)$ is a sample from the corresponding random variable at time instance n, then the probability that this sample is realized is:

$$E_h(v, n) = \Pr(e_h(n) = v) \tag{3.7}$$

3.2.2.3 Used Energy

The energy management system uses the available information, such as the current time period n and the energy in the storage $e_b(n)$, in order to determine a target used energy for the forthcoming period of time $(n \cdot \delta, (n + 1) \cdot \delta]$. A statistical model allows us to describe uncertainties in the actual used energy following the conditional probability

$$E_u(v, w, n) = \Pr(e_u(n) = v \mid e_b(n) = w) \tag{3.8}$$

This statistical model of the target energy $E_u(v, w, n)$ deserves some more explanation as it allows to model a large class of relevant energy management policies. Usually, the energy management is not statistical. However, when the system is in a certain mode of operation as defined by the energy manager, its energy consumption may still vary from one execution to the next due to run-time variations of tasks, or different requests from the environment. Examples of the latter are events to react on or packet re-transmissions in case of missing acknowledgments. The formulation in (3.8) allows us to model most well known energy management policies, as explained below.

- *Constant*: If there is only one mode of operation, and the target energy is r for every time n, then for all w and n we have $E_u(v, w, n) = 1$ if $v = r$ and $E_u(v, w, t) = 0$ otherwise.
- *Continuous Modes*: Many known policies choose from a continuous number of modes of operation, and corresponding target energy consumption values. Such is the FHC scheme, introduced in the previous chapter. With an update interval of one week, estimates of future harvested energy are static and repeat with a yearly period. This means that all target consumption values are functions of time n and stored energy

w. Here we just have $E_u(r(w,n),w,n) = 1$ for all w, n and $E_u(v,w,n) = 0$ for all $v \neq r(w,n)$.
Let us further illustrate having continuos modes of operation with the case where all energy above a certain value is targeted for consumption. Suppose that the threshold charge is $s(n)$, and that the target energy consumption for this storage level is $r(n)$. Then, the actual used energy is $E_u(r(n) + \max\{(w - s(n), 0\}, w, n) = 1$ for every w and n, and 0 otherwise.

- *Discrete Modes*: We also deal with energy management strategies where there is a finite number of modes of operation available, which is inspired by mixed-criticality systems.
  In the simplest case with two modes of operation, we commonly find the following strategy. If the energy in the storage is less or equal than a certain threshold value $c(n)$, then the system is set to operate in an emergency mode. In this mode, only important tasks are executed, while others may be dropped or executed in a reduced manner. Suppose that the normal target used energy is $r(n)$, and the reduced target used in emergency mode is $r_c(n)$, then we have $E_u(r(n),w,n) = 1$ if $w > c(n)$, $E_u(r_c(n),w,n) = 1$ if $w \leq c(n)$, and 0 otherwise.

3.2.2.4 Rechargeable Energy Storage Model

We start with the time-discrete system model (3.5) with discrete values for the energy in the storage. The distribution of the energy in the storage as well as the occurrence of a failure state is modeled by a row vector $\boldsymbol{B}(n) \in \mathbb{R}^{(\hat{b}+2)}$ which is a probability mass function of the random variable that describes the state of the system at time $n \cdot \delta$.

In particular, $B_1(n)$ is the probability of the system to be in a failure state and having an empty storage, $e_b(n) = 0$. Furthermore, $B_j(n)$ for $2 \leq j \leq \hat{b} + 2$ is the probability of having the stored energy $e_b(n) = j - 2$ at discrete time n. Note that the sum of all elements of $\boldsymbol{B}(n)$ is 1 as the states are mutually exclusive, i.e., being in a failure state with probability $B_1(n)$ and $e_b(n) = 0$ or being not in a failure state and having energies $e_b(n) = j - 2$ with

probabilities $B_j(n)$ for $2 \leq j \leq \hat{b} + 2$.

Based on our inclusive model, the proposed analysis methodology is able to statistically analyze and compare all of the aforementioned energy management policies in terms of utility, failure rate, or sensitivity or robustness to changing harvesting conditions. In the next section, we move on to the corresponding Markov analysis governed by (3.5), (3.7) and (3.8).

3.3 Stochastic Analysis

To first recapitulate, in Section 3.2.2 we defined a statistical model of the harvested energy and of the target energy use, and we defined various energy management policies. We suppose that harvested energy in consecutive time steps can be modeled with independent random variables. Now we explain the analysis technique. It is based on a Markov chain, where the states of the rechargeable energy storage, modeled by $\boldsymbol{B}(n)$, represent Markov chain states at time n.

The statistical analysis performed in this section is performed in 'two directions'. First, in Section 3.3.1, we determine the state transitions of the Markov chain for a known system, which we use ultimately to derive the system's probability of failure. Then, in Section 3.3.2, we present a method to optimize parameters of the energy management policy, such that defined bounds on the probability of failure are respected.

3.3.1 Markov Analysis

We start by defining the transition matrix, which describes how the energy storage transitions from time n to time $n + 1$. We then define the global transition matrix, a combination of periodically repeating transition matrices which describe transitions made over the period. In our solar energy harvesting example, the repeating period is one year, and the global transition matrix consists of 52 weekly transition matrices. With these values known, we obtain the steady state energy storage at the beginning of any discrete time n.

Finally, we derive the probability that we have at least one failure of the system over any interval.

3.3.1.1 Transition Matrix

The state transition matrix from time instance n to $n+1$ can be directly derived from (3.5), due to the independence assumption on the harvesting and target use energy probabilities. The failure state needs special attention as the resulting storage level at $n+1$ is supposed to be $e_b(n+1) = 0$. For a concise description of the state transition in matrix form, we use the abbreviations:

$$
\begin{aligned}
c_1(e_b, e_h, e_u) &= (e_b + \delta_2(\check{\sigma}\frac{e_h}{\delta} - \frac{e_u}{\delta}) < 0 \ \vee \ e_b + e_h - e_u < 0) \\
c_2(e_b, e_h, e_u) &= (e_b + \delta_1(\hat{\sigma}\frac{e_h}{\delta} - \frac{e_u}{\delta}) > \hat{b}) \\
c_3(e_b, e_h, e_u) &= (e_b + \delta_2(\check{\sigma}\frac{e_h}{\delta} - \frac{e_u}{\delta}) \geq 0 \ \wedge \ e_b + e_h - e_u \geq 0 \\
&\qquad \wedge \ e_b + \delta_1(\hat{\sigma}\frac{e_h}{\delta} - \frac{e_u}{\delta}) \leq \hat{b}) \\
f_1(e_h, e_u) &= \hat{b} + \lfloor \min\{0, (\delta - \delta_1)(\check{\sigma}\frac{e_h}{\delta} - \frac{e_u}{\delta})\} \rfloor \\
f_2(e_b, e_h, e_u) &= \min\{\hat{b}, e_b + e_h - e_u\} \\
e_b(n+1) &= \begin{cases} 0 \\ \qquad \text{if } c_1(e_b(n), e_h(n), e_u(n)) \\ f_1(e_h(n), e_u(n)) \\ \qquad \text{if } c_2(e_b(n), e_h(n), e_u(n)) \\ f_2(e_b(n), e_h(n), e_u(n)) \\ \qquad \text{if } c_3(e_b(n), e_h(n), e_u(n)) \end{cases}
\end{aligned}
\tag{3.9}
$$

This leads us to:

$$
\boldsymbol{B}(n+1) = \boldsymbol{B}(n) \cdot \boldsymbol{T}(n) \quad , \quad \boldsymbol{T}(n) = \boldsymbol{T}^a(n) + \boldsymbol{T}^b(n) + \boldsymbol{T}^c(n) \tag{3.10}
$$

where the partial transition matrices $\boldsymbol{T}^a(n), \boldsymbol{T}^b(n), \boldsymbol{T}^c(n)$ correspond to the three cases in (3.9), respectively, written in matrix form:

$$T^a_{i,j}(n) = \begin{cases} \sum\limits_{\substack{(e_h,e_u):c_1(0,e_h,e_u)\\ j=1}} E_h(e_h,n)\cdot E_u(e_u,0,n) \\ \qquad \text{if } i=1 \\ \sum\limits_{\substack{(e_h,e_u):c_1(i-2,e_h,e_u)\\ j=1}} E_h(e_h,n)\cdot E_u(e_u,i-2,n) \\ \qquad \text{if } i>1 \end{cases}$$

$$T^b_{i,j}(n) = \sum_{\substack{(e_h,e_u):c_2(i-2,e_h,e_u)\\ j-2=f_1(e_h,e_u)}} E_h(e_h,n)\cdot E_u(e_u,i-2,n)$$

$$T^c_{i,j}(n) = \begin{cases} \sum\limits_{\substack{(e_h,e_u):c_3(0,e_h,e_u)\\ j-2=f_2(0,e_h,e_u)}} E_h(e_h,n)\cdot E_u(e_u,0,n) \\ \qquad \text{if } i=1 \\ \sum\limits_{\substack{(e_h,e_u):c_3(i-2,e_h,e_u)\\ j-2=f_2(i-2,e_h,e_u)}} E_h(e_h,n)\cdot E_u(e_u,i-2,n) \\ \qquad \text{if } i>1 \end{cases} \tag{3.11}$$

State update example. We now show an example, explaining how the transition matrix is constructed and the rechargeable storage state is updated from time n to time $n+1$. To this end we suppose a storage capacity of $\hat{b}=4$, an initial storage level of $e_b(n)=3$. The energy target use is deterministic, with $e_u=0$ if $0\le e_b\le 2$ and $e_u=4$ if $3\le e_b\le 4$. In other words, if the storage level is 2 or below, no energy is spent, and if it is 3 or above, energy corresponding to the storage capacity is requested for use. The harvested energy can be described by a normal distribution with average 2 and variance 1. This can be all written as:

$$E_u(w,i,n) = \begin{cases} 1 & \text{if } w=0 \wedge 0\le i\le 2 \\ 1 & \text{if } w=4 \wedge 3\le i\le 4 \\ 0 & \text{otherwise} \end{cases}$$

$$\boldsymbol{B}^T(n) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \ , \ E_h(v,n) = \begin{cases} 0.023 & \text{if } v = 0 \\ 0.136 & \text{if } v = 1 \\ 0.341 & \text{if } v = 2 \\ 0.341 & \text{if } v = 3 \\ 0.159 & \text{if } v = 4 \end{cases}$$

We also suppose that the average power can only deviate by a factor within the interval $[\check{\sigma}, \hat{\sigma}] = [0.5, 2.0]$. Applying (3.10), we obtain the partial state transition matrices:

$$\boldsymbol{T}^a(n) = \begin{bmatrix} 0 & 0\,0\,0\,0\,0 \\ 0 & 0\,0\,0\,0\,0 \\ 0 & 0\,0\,0\,0\,0 \\ 0 & 0\,0\,0\,0\,0 \\ 0.023 & 0\,0\,0\,0\,0 \\ 0 & 0\,0\,0\,0\,0 \end{bmatrix} \ , \ \boldsymbol{T}^b(n) = \begin{bmatrix} 0\,0\,0 & 0 & 0 & 0 \\ 0\,0\,0 & 0 & 0 & 0 \\ 0\,0\,0 & 0 & 0 & 0 \\ 0\,0\,0 & 0 & 0 & 0.159 \\ 0\,0\,0 & 0.159 & 0 & 0 \\ 0\,0\,0 & 0.5 & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{T}^c(n) = \begin{bmatrix} 0 & 0.023 & 0.136 & 0.341 & 0.341 & 0.159 \\ 0 & 0.023 & 0.136 & 0.341 & 0341 & 0.159 \\ 0 & 0 & 0.023 & 0.136 & 0.341 & 0.5 \\ 0 & 0 & 0 & 0.023 & 0.136 & 0.682 \\ 0 & 0.136 & 0.341 & 0.341 & 0 & 0 \\ 0 & 0.023 & 0.136 & 0.341 & 0 & 0 \end{bmatrix}$$

from which we get the complete transition matrix. Thus we can also compute $\boldsymbol{B}(n+1)$, the state of the system after one transition:

$$\boldsymbol{T}(n) = \begin{bmatrix} 0 & 0.023 & 0.136 & 0.341 & 0.341 & 0.159 \\ 0 & 0.023 & 0.136 & 0.341 & 0341 & 0.159 \\ 0 & 0 & 0.023 & 0.136 & 0.341 & 0.5 \\ 0 & 0 & 0 & 0.023 & 0.136 & 0.841 \\ 0.023 & 0.136 & 0.341 & 0.5 & 0 & 0 \\ 0 & 0.023 & 0.136 & 0.841 & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{B}^T(n+1) = (\boldsymbol{B}(n) \cdot \boldsymbol{T}(n))^T = \begin{bmatrix} 0.023 \\ 0.136 \\ 0.341 \\ 0.5 \\ 0 \\ 0 \end{bmatrix}$$

3.3.1.2 Global Transition Matrix and Steady State Distribution

As previously stated, in our example we assume that the unit time interval is one week. Due to the annual solar cycle, the energy harvesting weekly statistical models are periodic, i.e., $E_h(v,n) = E_h(v,n+i\cdot 52)$ where $v,n,i \in \mathbb{Z}_{\geq 0}$, and we use a simplification where each year consists of exactly 52 weeks. The energy consumption statistical distributions are under the control of the energy manager, and can therefore be assumed to be periodic with a period of 52 weeks as well: $E_u(v,n) = E_u(v,n+i\cdot 52)$ where $v,n,i \in \mathbb{Z}_{\geq 0}$.

Now, since both of these models are periodic, the transition matrices are also periodic, i.e., $\boldsymbol{T}(n) = \boldsymbol{T}(n+i\cdot 52)$. Thus we define the global transition matrix, which is invariant for each period, and can be computed as:

$$\boldsymbol{T} = \prod_{0\leq n\leq 51} \boldsymbol{T}(n) \tag{3.12}$$

In this case, this is also referred to as the yearly transition matrix. Using this matrix, and that $\boldsymbol{B}(n)$ us a row vector representing the state at time n, we can compute the steady state rechargeable storage level, both at the start of the year and at the start of a given week n. It is computed as the stationary distribution of the Markov chain:

$$\begin{aligned} &\boldsymbol{B}^{\boldsymbol{\infty}}(0)\cdot\boldsymbol{T} = \boldsymbol{B}^{\boldsymbol{\infty}}(0) \\ &\boldsymbol{B}^{\boldsymbol{\infty}}(n) = \boldsymbol{B}^{\boldsymbol{\infty}}(0)\cdot \prod_{0\leq i<n} \boldsymbol{T}(i) \quad \forall n\in\{0,1,\ldots 51\} \end{aligned} \tag{3.13}$$

Several methods for solving (3.13) exist [PSW75].

3.3.1.3 Steady State Probability of Failure

The system encounters a failure if it attempts to use more energy than available. Between the failure time and the end of the update interval, additional energy may be harvested. Nevertheless, in Section 3.2.2.4 we introduced a safe simplification that no energy is available at the update time after a failure, $e_b(n) = 0$.

Using that $B_1(n)$ is the worst-case probability that the system had a failure in the preceding time interval, we can denote the steady state probability of failure at time n as $B_1^\infty(n)$. We now define an upper bound on the yearly probability of failure:

$$\lambda = \min\left\{\sum\nolimits_{0\leq n<52} B_1^\infty(n),\ 1\right\} \tag{3.14}$$

The upper bounding is safe because, whenever finding the probability that at least one out of two events happens, we have $\Pr(a \cup b) \leq \Pr(a) + \Pr(b)$ and $\Pr(a \cup b) \leq 1$.

Consecutive Failures The yearly probability of failure λ, as defined above, does not imply any distribution of failures in the time domain. If we want to avoid having the system consecutively entering the failure state, the system would need to fulfil an additional constraint: *the system should not exit the failure state, until the energy storage is sufficiently charged.* In this sense, 'sufficiently charged' is defined to mean that the probability of failure in the first week of operation, with a condition that the initial energy level is not less than the 'sufficient' level, is less than the steady state probability of failure for the same week. Our system model can not consider such a failure recovery mechanism, because it can not model hysteresis behavior necessary for failure recovery. We discuss this limitation, and a possible way to overcome it, in Section 3.6.

3.3.2 Energy Management Optimization

In this section, we explain computing the target energy use for several energy management strategies. Assume that a constraint on the yearly probability of failure $\hat{\lambda}$ is given.

- *Constant:* When there is one mode of operation, we aim to maximize the target energy consumption of the mode r.
- *Constant with Safe Charges:* Here, there are continuos options for the system's mode. The most important is the *nominal* mode, with a target energy use of r. Otherwise, if the stored energy level at the start of a time interval is higher than a *safe charge* $s(n)$, then the system aims to consume this excess

energy in the next interval as well. In this scenario, we aim to maximize the target use energy $r + \max\{e_b(n) - s(n), 0\}$, for every n.

- *Variable with Safe Charges and an Emergency Mode:* Again, we assume continuos modes of operation. The nominal target energy consumption $r(n)$ is now a function of the time n. As before, if the stored energy level at the start of a time interval is higher than the corresponding safe charge $s(n)$, then this excess energy is consumed in the next time interval. Additionally, if the stored energy level at the start of a time interval is *less or equal* to the nominal target use energy $r(n)$, then emergency mode is entered. In this mode, the target energy use is reduced to a constant *reduced nominal* r_c. In this whole scenario, we aim to maximize the actual target use energy $r(n) + \max\{e_b(n) - s(n), 0\}$ for every n, assuming r_c is given.

Let us now focus on each policy in detail.

Constant. It is clear from (3.10) that λ is monotonically increasing with the target use energy r. Intuitively, this is because the stored energy levels for all weeks decrease if r is increased. Therefore, given a constraint on the failure probability $\hat{\lambda}$, the maximum target use energy r can be determined efficiently by a simple binary search.

Constant with safe charges. Here we present a heuristic algorithm for optimizing a energy management policy with safe charges. Obviously, introducing safe charges can not reduce the yearly probability of failure λ. Consecutively, we start by calculating the maximum nominal target use energy r, given a constraint on the failure probability $\hat{\lambda}$. Then, we introduce safe charges such that the increased failure probability is again at most $\hat{\lambda}$. The safe charges are calculated starting from one week *after* the week with the highest probability of failure, and proceeding to one week after the week with the second highest probability of failure, and so on. Intuitively, the energy storage should be the lower after these critical weeks, so it makes sense to start introducing safe charges in this order. This is all as described in Algorithm 3.1.

Algorithm 3.1: Computing the nominal target use, as well as safe charges

1 **procedure** get safe charges $(E_h(v,n), \hat{\lambda})$
2 $\quad \forall n: \ s(n) = \hat{b}$,
3 $\quad \mathbb{S} = \emptyset$
4 $\quad$ Using binary search, maximize r such that $\lambda \leq \hat{\lambda}$
5 $\quad$ **repeat**
6 $\quad\quad \hat{i} = \arg\max_{0 \leq i < 52 \ \wedge \ i \notin \mathbb{S}} B_1^{\infty}(i)$
7 $\quad\quad$ Using binary search, minimize $s((\hat{i}+1) \mod n)$ such that $\lambda \leq \hat{\lambda}$
8 $\quad\quad$ Add $\hat{i}$ to $\mathbb{S}$
9 $\quad$ **until** $|\mathbb{S}| == n$
10 $\quad$ **return** $r, \quad \forall n: \ s(n)$

Variable with safe charges and an emergency mode. Here we provide an algorithm which optimizes safe charges with variable nominal target use energy. The algorithm is the same as Algorithm 3.1, but we introduce additional steps between lines 4 and 5. Precisely, we split time intervals into several groups based on their average harvested energy. In the solar harvested energy example, we have found 7 groups to be practical. Then, we disregard the group with the least average energy, and maximize the requested energy use $r(n)$ for all the other groups. This process repeats iteratively until all groups are disregarded. Note that in this case, useful metrics are both the yearly probability of failure λ, and the yearly probability of being in emergency mode η, where:

$$\eta = \sum\nolimits_{0 \leq n < 52} \sum_{0 < i \leq r_c} B_{i+2}^{\infty}(n)$$

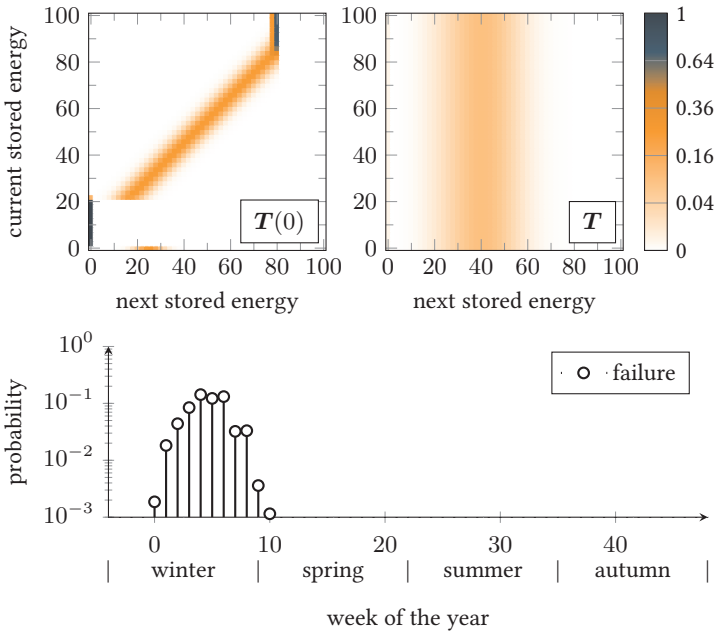
3.3.3 Example

We now illustrate the yearly transition matrices and probability of failure. Assume first that the harvesting models are given. We used models as described in Section 3.5.1 for location MI, with an appropriate solar panel such that the weekly average of harvested energy is 37. The harvesting model for 2 weeks at this location

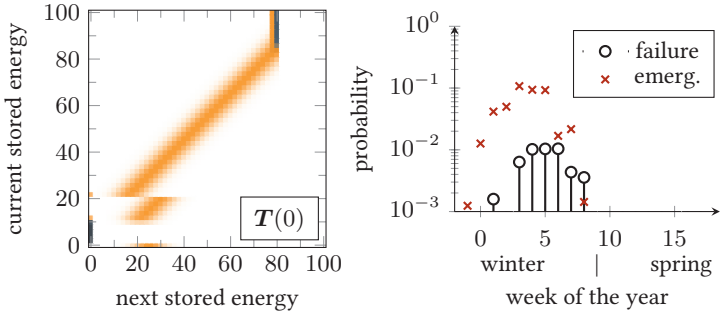
is shown in Figure 3.3a. Next, assume that the target energy use of the application is $E_u(w,i,n) = 0.5$ if $w > 0 \wedge i = 30$, and $E_u(w,i,n) = 0.25$ if $w > 0 \wedge (i = 29 \vee i = 31)$, or zero otherwise. In other words it is 30 ± 1, except it is zero when the energy storage is empty because the system is assumed to be in a failure state then. Finally, assume the rechargeable energy storage capacity is $\hat{b} = 100$. This information allows us to determine the week transition matrices $\boldsymbol{T}(n)$ using (3.10), and the yearly transition matrix $\boldsymbol{T}$.

Transition matrices $\boldsymbol{T}(0)$ and the yearly $\boldsymbol{T}$ are shown in Figure 3.2a. We see that for week 0, stored energy states lower than 21, excluding state 0, surely lead to a failure state. Otherwise, the combination of harvested and stored energy could be enough to power the system through the week. The yearly transition matrix is very similar for all rows. This means that the initial state has little impact on the state after one year. Figure 3.2a also shows the steady state probability of failure values for each week. It is typical for the failure probability to be the highest towards the end of winter. For this case, we have a yearly failure probability $\lambda = 0.615$.

Now we proceed to illustrate emergency mode. Let us define that emergency mode is entered when the energy storage is charged less than 21, and the target use energy then is 10 ± 1. Mathematically, $E_u(w,i,n) = 0.5$ if $w \geq 21 \wedge i = 30$, $E_u(w,i,n) = 0.25$ if $w \geq 21 \wedge (i = 29 \vee i = 31)$, $E_u(w,i,n) = 0.5$ if $0 < w < 21 \wedge i = 10$, $E_u(w,i,n) = 0.25$ if $0 < w < 21 \wedge (i = 9 \vee i = 11)$, or zero otherwise. With this target use function, the transition matrix for week 0 is shown in Figure 3.2b. Furthermore, we can see that the yearly failure probability is $\lambda = 0.048$, while the yearly probability of being in emergency mode is $\eta = 0.440$. This illustrates that we can essentially trade-off the aggregate of failure probability, with the probability of operating in emergency mode. Though the service is reduced sometimes, the system is less likely to fail, and instead enters emergency mode in periods of energy scarcity. This helps in improving the robustness of the system.



(a) Constant consumption



(b) Constant consumption with an emergency mode ($r_c = 20$)

Figure 3.2: Select transition matrices and the probability of being in failure or emergency mode, for various consumption strategies

3.4 Non-Ideal System Model

Our system model so far was simplified so not to take into account non-ideal behavior of the energy storage. Fortunately, the model can be extended to incorporate inefficiencies that may exist in the power management hardware, without affecting results of our stochastic analysis technique.

Deterministic system model. If $\tilde{p}_h(t)$ is the harvested power and $\tilde{p}_u(t)$ is the target use power at time t of the continuous system model, then the following non-ideal behavior is modeled:

- If the harvesting device generates power $\tilde{p}_h(t)$ at time t, the power entering the energy storage is only $\sigma_h(b(t)) \cdot \tilde{p}_h(t)$.
- If the target use power is $\tilde{p}_u(t)$ at time t, then the power drawn from the energy storage is $\sigma_u(b(t))^{-1} \cdot \tilde{p}_u(t)$. In addition, there is a leakage $\delta_u(b(t))$ of the energy storage.

The functions $\sigma_h(b(t))$ and $\sigma_u(b(t))$ are charging and discharging efficiencies, and they can be any function of the storage level $b(t)$. In practice, these functions can be easily estimated through characterization of the hardware. To get both the non-ideal continuous and discretized system models, we just need to add these efficiencies wherever the power values are, e.g., in (3.1).

Stochastic system model. In the case with the non-ideal stochastic system model, we likewise need to add the efficiencies at appropriate places. When the efficiencies are added, values $e_h(t)$ and $e_u(t)$ from (3.7) and (3.8) become functions of the storage level $b(n)$ as well. Especially, (3.7) becomes:

$$E_h(v, w, t) = \Pr(e_h(t) = v \mid e_b(t) = w)$$

while $E_u(v, w, t)$ was a function of the storage level anyway. After these two values are redefined, the whole stochastic analysis technique applies.

To demonstrate how an non-ideal system can be characterized, we refer the reader to the real-world system developed in the previous chapter.

3.5 Trace-Based Experiments

This section illustrates our stochastic analysis through several simulations. We investigate two hypothetical scenarios, when a solar energy harvesting embedded system is deployed outdoors or indoors, each with the goal of operating for multiple years. To this end, we start by demonstrating how historical data is used to build energy harvesting models in Section 3.5.1. For these experiments, we use the simplified model of harvested energy with no constraints present.

Then we move on to the sensitivity experiments and design space exploration. In both cases, we deal with a system with two design parameters: the solar panel size and the rechargeable energy storage capacity $\hat{b}$, and a constraint on the yearly probability of failure $\hat{\lambda}$. As well as this, in both cases, a constant target use energy r is used. In the sensitivity experiments in Section 3.5.2, we fix two of the aforementioned values and vary the third, and then for each design point we maximize the constant target use energy r. The most interesting result here is the sensitivity of the target use energy r with regards to the probability of failure constraint $\hat{\lambda}$. The design space exploration in Section 3.5.3 illustrates a scenario where the target use energy r and constraint $\hat{\lambda}$ are given, and we search for satisfactory design points, i.e., valid combinations of the solar panel size and storage capacity $\hat{b}$.

The experiments are presented for 3 outdoor and 3 indoor locations. The outdoor locations feature a desert, a hot continental, and a temperate oceanic climate. The indoor locations are close to a window with direct sunlight sporadically, close to a window but without direct sunlight, and inside an artificially lighted hallway.

Afterwards, we focus our attention on target use functions. Particularly in Section 3.5.4, we evaluate the performance of our most complex energy management scheme, which features safe charges and an emergency mode of operation. It is compared to three schemes from the state of the art, a reactive one, a proactive one, and an practically unfeasible optimal one. Finally, in Section 3.5.5 we validate our stochastic model, by comparing an energy management scheme characterized by our stochastic analysis, with the same scheme simulated in continuous time.

Abbr.	Location Name	Latitude	Climate
ZH	Zürich, Switzerland	47.22 N	temperate oceanic / continental (Cfb/Dfb)
MD	Madison, USA	43.07 N	hot-summer humid continental (Dfa/Dfb)
KZ	Karezat, Pakistan	30.45 N	hot desert (BWh)

(a) Outdoor locations

Abbr.	Location Name	Environment
L14	NE Window	significant natural light with potential direct sunlight
L17	W Window	significant natural light with no direct sunlight
L18	Hallway	no natural light, indirect artificial light

(b) Indoor locations

Table 3.1: An overview of used locations and their characteristics.

3.5.1 Harvested Energy Models

We now explain how harvested energy models are obtained, both for the indoor and outdoor scenarios. For the purpose of this experiments, we took no assumptions on the harvested energy, i.e., we did not assume that harvested energy has to stay withing bounds defined by factors $\check{\sigma}$ and $\hat{\sigma}$.

Outdoor data. The outdoor data has been taken from two public databases, the national solar radiation database (NSRDB) [SXL+18] and the HelioClim-1 database [Wal13]. The first database features solar radiation data for 22 years (1998-2019) for North American locations and 15 years (2000-2014) for South Asian locations. The second database has 21 years (1985-2005) of solar radiation, and together these two databases cover most of Earth. For simulated experiments we tested many locations, diverse in both climate, latitude, and altitude. Due to brevity, in the reminder of this section we focus on three locations we found representative, ZH from HelioClim-1, and MD and KZ from NSRDB, and they are described in Table 3.1a.

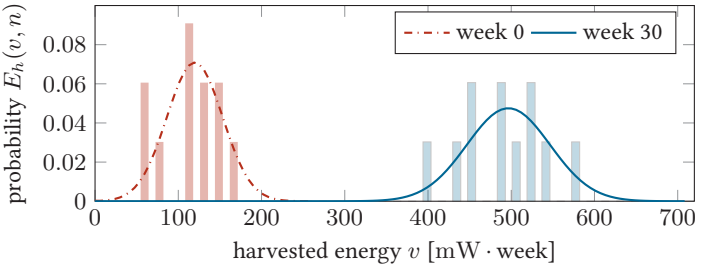
For each outdoor location, harvested energy models $E_h(v, n)$ have been constructed for each week using maximum likelihood estimation on the first 11 years of historical data. A normal distribution of samples was assumed, as you can see in Figure 3.3a. The used solar panel size depends on the experiment in question (usually $100\,\mathrm{cm}^2$), while we assumed that $20\,\%$ of the solar radiation energy is harvested.

Indoor data. Indoor solar harvesting data is much more scarce. We used the Indoor Solar Harvesting Dataset [SGT19] which features up to 3 years (2017-2020) of data at six locations inside a university office building in Zürich. Here we also focus on three locations representative locations, listed in Table 3.1b.

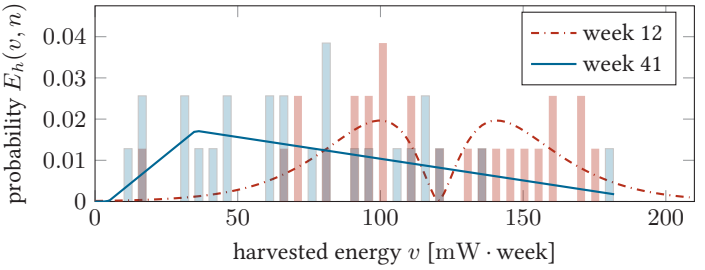
To compensate for little training data, the harvested energy model $E_h(v, n)$ for week n has been constructed using maximum likelihood estimation on the first 2 years of historical data, but for weeks $n - 3$ to $n + 3$. Next, we searched for fitting distributions to construct the model, and one of these was empirically found to be suitable for each week and location: normal, laplace, gompertz, hypergeometric, triangular, beta, double gamma, or double weibull. In Figure 3.3b you can see two distributions, and historical data from which they were constructed. The indoor dataset features energy harvested on a sample solar panel. We assumed the same type of panel and harvesting subsystem, but changed the panel's size based on the experiment.

3.5.2 Sensitivity Experiments

Here we do three experiments the following way. Observe three system parameters: the storage capacity $\hat{b}$, the solar panel size, and the constraint on the yearly probability of failure $\hat{\lambda}$. In each experiment, we fix two of the values while we vary the third, and we measure the impact on the maximal target energy use r. Remember, here we assume a system which consumes a constant target energy throughout the year, while the harvesting models are derived from historical data as described in Section 3.5.1. The experiments are presented for all six locations from Table 3.1. Let us now focus on each sensitivity experiment individually.



(a) Weeks $n = 0$ and 30 at location MD, modeled with normal distributions

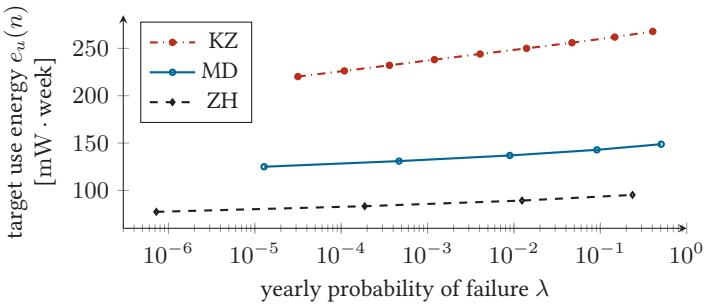


(b) Weeks $n = 12$ and 41 at location L17, modeled with a triangular and double gamma distribution, respectively

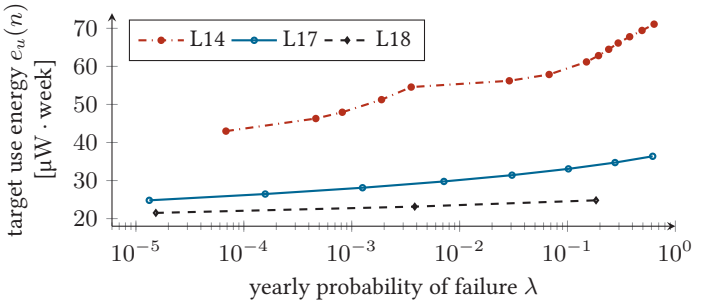
Figure 3.3: Historical data on harvested energy (bars), and the corresponding fitted models (lines)

3.5.2.1 Probability of Failure

Setup. For the outdoor simulation setup, we assume a solar panel of size $100\,\mathrm{cm}^2$, and a rechargeable energy storage with capacity $\hat{b} = 360\,\mathrm{kJ}$, which translates to about $600\,\mathrm{mW}$ over a week. Indoors, we took a solar panel of size $49.5\,\mathrm{cm}^2$ ($198\,\mathrm{cm}^2$ for location L18), and a rechargeable energy storage with capacity $\hat{b} = 100\,\mathrm{J}$, which is about $165\,\mu\mathrm{W}$ over a week. Both indoors and outdoors we tried out different target use energy values, and then recorded corresponding probabilities of failure. The results are shown in Figure 3.4.



(a) Outdoor locations



(b) Indoor locations

Figure 3.4: The target use energy as a function of the yearly probability of failure, for various locations

Evaluation. We observe immediately that the target use energy is very sensitive with regards to the probability of failure. Or in other words, the maximal target use with which the system never runs out of energy, is very close to the minimal target use with which the system always runs out of energy.

The slope of this change is similar for all three outdoor locations, which is peculiar as the climates are so different. At location KZ, the plotted target use energy values are from 220 to 268 $\mathrm{mW} \cdot \mathrm{week}$, the difference being about 22 %. At the other two locations, the values are from 125 to 150 $\mathrm{mW} \cdot \mathrm{week}$ (a 19 % difference) and from 77 to 95 $\mathrm{mW} \cdot \mathrm{week}$ (a 23 % difference). Indoors, we observe

more diverse behavior. Location L14, which sometimes has direct sunlight, has target use energy values from 43 to $71\,\mu\mathrm{W}\cdot\mathrm{week}$ (a $63\,\%$ difference), while for the artificially-lighted location L18 these values are from 21.5 to $25\,\mu\mathrm{W}\cdot\mathrm{week}$ (a $15\,\%$ difference).

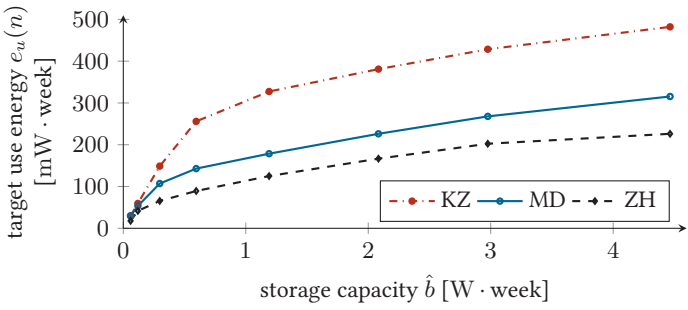
3.5.2.2 Energy Storage Capacity

Setup. For this experiment, we take the constraint on the yearly probability of failure $\hat{\lambda}$ to be $10\,\%$, meaning one failure is expected to happen no more than once every 10 years of operation. The solar panel size is the same as before, which means $100\,\mathrm{cm}^2$ for outdoor locations, $49.5\,\mathrm{cm}^2$ for locations L14 and L17, and $198\,\mathrm{cm}^2$ for location L18. Outdoors, the rechargeable energy storage capacity $\hat{b}$ was varied from 36 to $2700\,\mathrm{kJ}$, which translates to from about $60\,\mathrm{mW}$ per week to about $4.5\,\mathrm{W}$ per week. Indoor, the capacity was varied from 10 to $750\,\mathrm{J}$, or from about $16.5\,\mu\mathrm{W}$ per week to about $1.2\,\mathrm{mW}$ per week. The corresponding target use energy values are shown in Figure 3.5.

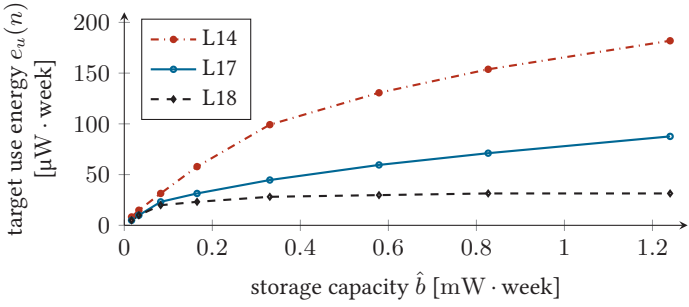
Evaluation. As the storage capacity increases, the target use energy increases monotonically towards a limit. This makes sense, as a large enough rechargeable storage would yield the yearly average of harvested energy to be the constant target use, which is the theoretical upper bound. In other words, we find that increasing the storage capacity has diminishing returns.

3.5.2.3 Solar Panel Size

Setup. In the final sensitivity experiment, illustrated by Figure 3.6, we varied solar panel sizes. Outdoors they range from 10 to $500\,\mathrm{cm}^2$, while indoors they are from 8.25 to $99\,\mathrm{cm}^2$. The other two parameters are as in the previous experiments: the constraint on the probability of failure is $\hat{\lambda} = 10\,\%$, outdoors the storage capacity is $\hat{b} = 360\,\mathrm{kJ} = 600\,\mathrm{mW}\cdot\mathrm{week}$, while indoors this value is $\hat{b} = 100\,\mathrm{J} = 165\,\mu\mathrm{W}\cdot\mathrm{week}$.



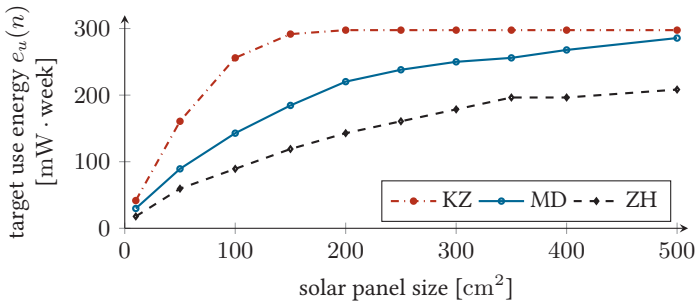
(a) Outdoor locations



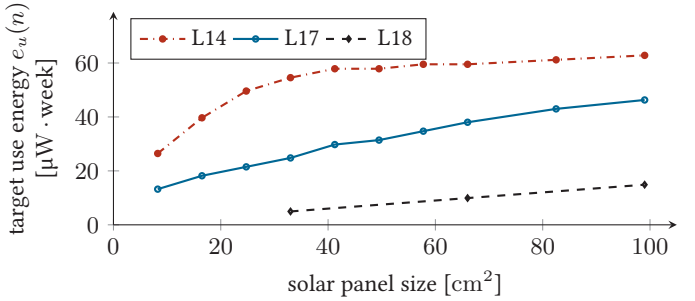
(b) Indoor locations

Figure 3.5: The target use energy as a function of the energy storage capacity, for various locations

Evaluation. Overall these results are as expected, increasing the solar panel size enables a higher target use energy. However, an interesting corner case is revealed at location KZ, when the solar panel is big with regards to the rechargeable storage capacity. Specifically, when the target use energy r is greater than half of the storage capacity, $r > \hat{b}/2$, our analysis evaluates that the failure state is surely entered. This is because we assumed a lack of constraints on the harvested energy. This reasoning is as follows. Because we defined our transition matrices in a worst case fashion, among other things they model the case when all of the target use energy from one week is spent right at the end of the week, and



(a) Outdoor locations



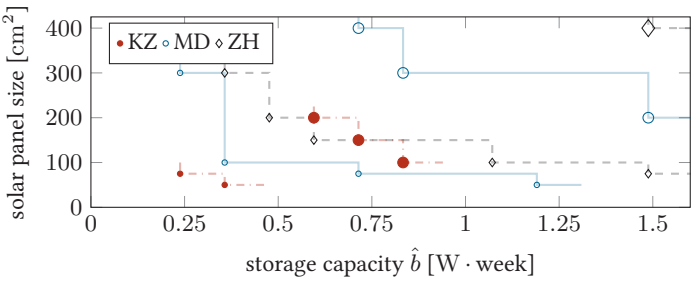
(b) Indoor locations

Figure 3.6: The target use energy as a function of the solar panel size, for various locations

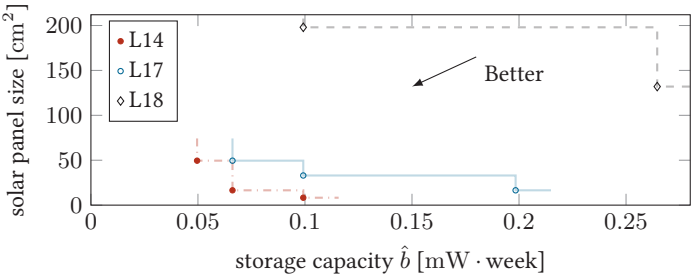
then all of the target use energy from the following week is spent at the beginning of the next. Thus, without any harvesting constraints, the stochastic analysis views having $r > \hat{b}/2$ as a guarantee to drain the whole storage.

3.5.3 Design Space Exploration

Setup. The overall point is to illustrate that design space exploration is possible, so we searched for valid combinations of solar panel size and energy storage capacity $\hat{b}$, for all locations from Table 3.1. The constraint on the yearly probability of failure $\hat{\lambda}$



(a) Outdoor locations for two target use $e_u(n)$ values: 120 mW · week (smaller points, lower fronts) and 300 mW · week (bigger points, higher fronts)



(b) Indoor locations for target use energy $e_u(n) = 20\,\mu\text{W} \cdot \text{week}$

Figure 3.7: Pareto optimal design solutions with regards to solar panel size and storage capacity, for various locations and target use energy values

was taken to be 10 %. For target use energy values r, which we assume in this case to be constant throughout the year, we tested two values outdoors $r = 120$ and 300 mW · week, and one value indoors $r = 20\,\mu\text{W} \cdot \text{week}$.

Evaluation. Figure 3.7 shows that multiple design points are available for every target use energy and location. Outdoors, the knee point seems to be around the same ratio for all locations and target use energy values, $200 \sim 300$ [cm^2/W · week]. Furthermore,

we confirm our previous results that increasing the battery has diminishing benefits. It is clear from the figures that adding a larger storage capacity saves only a bit on the solar panel size. On the other end of the spectrum, despite our increase of the solar panel size, we still need a certain energy storage to enable energy use over low energy harvesting winter weeks.

3.5.4 Energy Management Performance

Setup. In this simulated experiment, we compare our proposed energy management scheme with safe charges and an emergency mode, with three state of the art solutions. Our scheme, which we denote here $e_u^{prop\cdot}(n)$, is constructed as described in Section 3.3.2, with a constraint on the yearly probability of failure $\hat{\lambda} = 0.01$ and an emergency target use of half of the minimal use energy $r_c = \min\left(e_u^{prop\cdot}(n)\right)/2$.

The state of the art schemes we make a comparison to are already known from the previous chapter. First we have EnoMax [VGB07], a reactive control algorithm, and its target use $e_u^{\text{ENO}}(n)$ is designed to keep the stored energy level constant. Next, finite horizon control (FHC) is based on model predictive control. Its target use energy $e_u^{\text{FHC}}(n)$ depends on the current stored energy $e_b(n)$ and an estimate of stored energy in the following time intervals, and its design goal is to maximize the lowest target use energy value. Finally, we have the optimal target use energy $e_u^{opt}(n)$. This scheme can not be implemented in practice as it needs complete knowledge of how much energy will be harvested in the future, and is used here as a theoretical upper limit of performance.

The energy management schemes were simulated on a system with a $100\,\text{cm}^2$ solar panel and a rechargeable storage of capacity $\hat{b} = 1.2\,\text{W}\cdot\text{week}$. Results are presented for location MD, where performance for 11 years (2009-2019) was evaluated, and location KZ, where the experiment simulated 4 years (2011-2014). These test years were not used in the construction of the corresponding energy harvesting models.

Three metrics were used to make comparisons, and the first is *total utility*. In principle, the utility function can be any non-decreasing concave function of target use energy. Using that $[0, T)$

Table 3.2: Comparison of energy management schemes for outdoor locations MD and KZ with $T = 11$ and 4 years of simulated data, respectively. Short notation $\min$ means $\min e_u(n)$ $[\mathrm{mW} \cdot \mathrm{week}]$, while 99% means $\min_{99\%} e_u(n)$ $[\mathrm{mW} \cdot \mathrm{week}]$

	$e_u^{prop.}(n)$			$e_u^{\mathrm{FHC}}(n)$		
Abbr.	$\sum U$	min	99%	$\sum U$	min	99%
MD	0.971	89	173	0.906	89	161
KZ	0.974	327	327	0.903	315	315
	$e_u^{\mathrm{ENO}}(n)$			$e_u^{opt}(n)$		
Abbr.	$\sum U$	min	99%	$\sum U$	min	99%
MD	0.914	0	65	1	186	186
KZ	0.887	0	250	1	382	382

is the length of our experiment, we define the utility to be:

$$\forall n \in [0, T): \quad U(e_u(n)) = \sqrt{e_u(n)} \cdot \left(\sum_{\forall m \in [0,T)} \sqrt{e_u^{opt}(m)} \right)^{-1}$$

Then, the total utility $\sum U$ is defined to be the sum of $U(e_u(n))$ for all $n \in [0, T)$, and by definition it is a number between 0 and 1. The second objective we use is the minimal target use energy $\min e_u(n)$, where n is in $[0, T)$. The third objective is the 1-percentile of the target use energy, denoted $\min_{99\%} e_u(n)$. We introduce this metric as the constraint on the probability of failure is $\hat{\lambda} = 0.01$, so it makes sense evaluate the best $99\,\%$ of cases.

Evaluation. As seen in Table 3.2, our proposed energy management is always better in terms of total utility $\sum U$, compared to the practically implementable schemes. This is because FHC is conservative with regards to spending excess energy in the summer months, so some energy gets wasted due to energy storage filling up. By its conception, EnoMax is designed to keep the stored energy level constant. Therefore, harvested energy is rarely wasted,

making EnoMax comparable to FHC in this regard. The second and third metric are related to the minimal used energy, $\min e_u(n)$ and $\min_{99\%} e_u(n)$. For both locations, our proposed target use energy leads to results slightly better than the state of the art. EnoMax does not perform well with regards to these metrics, per design.

3.5.5 Pessimism of the Stochastic Analysis

Setup. In the final experiment, we validate our model and quantify the pessimism of the analysis, by viewing one energy management scheme both in the stochastic and the continuous domain. We assume a system with a solar panel of size $10\,\mathrm{cm}^2$, and a $\hat{b} = 0.6\,\mathrm{W}\cdot\mathrm{week}$ rechargeable storage capacity. As in the last experiment, we used our proposed energy management scheme with safe charges and an emergency mode $e_u^{prop.}(n)$, which is constructed as described in Section 3.3.2. The emergency target energy use is given to be a low value of $r_c = 60\,\mathrm{mW}\cdot\mathrm{week}$. For such a system, we simulated its operation over 11 years (2009-2019) at location MD. Note that these years were not used in the construction of energy harvesting models at this location.

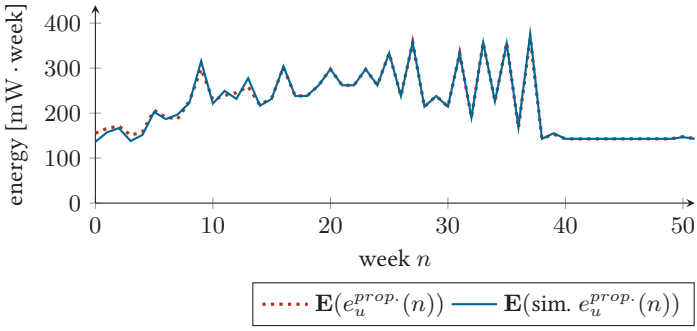
Evaluation. Figure 3.8 shows the target use energy, averaged over multiple years, $\mathbf{E}(e_u^{prop.}(n))$, and the energy level in the rechargeable storage, also averaged over multiple years, $\mathbf{E}(e_b(n))$. The two values are presented both when they are derived via our stochastic analysis, and when they are simulated. Most importantly, results show a close match between both set of values, especially for the stored energy.

3.6 Summary

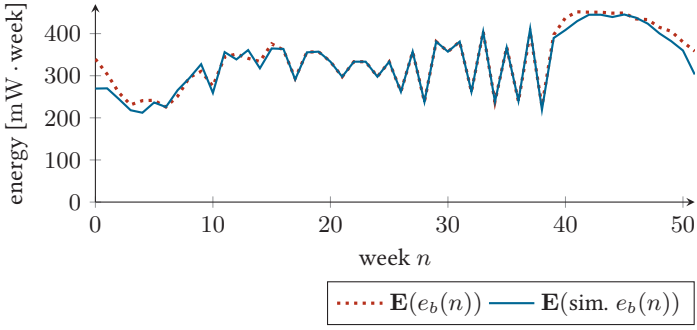
In this chapter, we defined a first stochastic analysis of energy harvesting embedded systems based on Markov chains that takes into account both the uncertainness in the harvesting environment, as well as in the consumed energy. The analysis enables us to pessimistically analyze long-term energy management strategies,

especially when complex adaptive energy management schemes are used. Among others, results of this analysis include the probability of system failure, the probability the system enters an emergency mode of operation, as well as steady state charge of the rechargeable storage. To demonstrate the applicability of our analysis to real-world deployments, we sketched how to deal with implantation artifacts in practical systems. Finally, extensive trace-based simulations for two harvesting scenarios, harvesting solar energy indoors and outdoors, illustrated a robustness or sensitivity analysis with regards to design parameters, then design space exploration, and finally evaluation of our developed energy management scheme. Our own scheme is shown to outperform the state of the art in terms of utility, while it never under performs with regards to the minimal energy used.

In terms of limitations, at this stage we can not model nor analyze any failure recovery mechanism, as it involves hysteresis behavior. In a future work it is possible to model hysteresis as well, by expanding the number of possible states the system is allowed to be in. Furthermore, we require random variables modeling the harvested and consumed energies to be independent from one time interval to the next. Lifting this requirement, and modeling certain forms of dependance, is a natural next step.



(a) Harvested energy and the target use energy



(b) Stored energy in a $\hat{b} = 600\,\text{mW} \cdot \text{week}$ rechargeable storage

Figure 3.8: The target use energy and the rechargeable energy storage level, derived using our analysis analysis and simulated continuously (denoted sim.)

4

Stochastic Analysis of Mixed-Criticality Scheduling

Mixed-criticality systems need to fulfill strict real-time standards that dictate different requirements for each criticality level, for example, given in the 'probability of failure per hour' format. A recent trend suggests designing these kinds of systems by jointly scheduling tasks of different criticality levels on a shared platform. When this is done, an adaptive task scheduler can degrade tasks of lower criticality when a higher criticality task needs more resources, for example, when it overruns a bound on its execution time. However, a way to quantify the impact this degradation has on the overall system is not well understood.

Meanwhile, to improve schedulability and avoid excessive provisioning of resources due to overly pessimistic WCET estimates, a new paradigm emerged where tasks' execution times are modeled with random variables. In this chapter, we analyze a system with probabilistic execution times, and propose metrics that are inspired by safety standards. Among these are the probability of deadline miss per hour, the expected time before degradation happens, and the duration of the degradation. We argue that these quantities provide a holistic view of the system's operation and schedulability.

Table 4.1: Failure rate specifications for different criticality levels, in the case of avionics systems' standard DO-178B

Level	Failure Condition	Failure Rate per hour
A	Catastrophic	10^{-9}
B	Hazardous	10^{-7}
C	Major	10^{-5}
D	Minor	10^{-3}
E	No effect	not defined

4.1 Introduction

Mixed-criticality (MC) systems are real-time systems that feature tasks of different criticality levels. Typical application domains include avionics and automotive [BD17]. In MC systems, each task has an associated criticality level. Depending on the criticality level, a failure of a task, for example due to deadline miss, can have a more or less severe impact on the overall safety of the system. Due to possible catastrophic consequences of a system failure, MC systems for some application domains are subject to certification standards. For example, DO-178C [RE12] is a standard for avionics systems. It defines five criticality levels, 'A' to 'E', with 'A' being the highest criticality level. Here, a failure of a task of criticality 'B' can have a negative impact on the overall safety of the aircraft, while a failure of a task of criticality 'D' may only slightly increase the aircraft crew's workload. Quantitatively, an application's criticality correlates to a tolerable failure rate under a given certification standard. The failure rates of all tasks, under their respective criticality levels, have to be guaranteed for certification of the overall system. As an example, Table 4.1 states the tolerable failure rates for DO-178B, the predecessor of DO-178C.

Traditionally, industry has favoured physical segregation of tasks based on their criticality level [TSP15]. This implies, for example, that tasks of each criticality level execute on their own hardware, and tasks of different criticality levels do not interfere. However, such a physical separation based on criticality levels can

lead to system under-utilization and complex distributed multi-processor architectures. Recently, there has been a push towards integrating tasks of different criticality levels on a single hardware platform [BD17]. The advantages for such consolidation include reduction in cost, power dissipation, weight, as well as maintenance.

Unfortunately, the described consolidation of criticality levels makes isolating tasks of different criticality levels problematic. Essentially, a low criticality level 'D' task may hinder the execution of a higher criticality level 'B' task, possibly resulting in a deadline miss – which can be considered as a type of failure. To counter this, researchers have proposed several schemes which are covered in detail in Section 4.2. Broadly speaking, the approaches are based on an execution time abstraction proposed by Vestal [Ves07]. Vestal's model builds on the worst-case execution time (WCET) abstraction. He assumes that tasks have a set of WCET estimates with different levels of confidence. The system is required to meet the deadline of a criticality level 'A' task for the highest confidence and most pessimistic WCET estimates. For lower criticality tasks, correct execution needs to be guaranteed for less pessimistic WCET estimates. Prominent proposed approaches that build on Vestal's model feature mode-based scheduling schemes that ensure that the system executes tasks of all criticality levels correctly when less pessimistic WCET estimates are not overrun, while reduced service to lower criticality tasks is in place when this is not the case.

In this chapter, instead of taking a single WCET estimate as in the traditional real-time model, or taking a criticality dependent set of WCET estimates as per Vestal's model, we assume a stochastic model of execution times. For each task, the execution time is modeled with a random variable. This additional information on the execution time allows us to have improved schedulability due to the so called multiplexer gain, i.e., the likelihood of high execution times of many tasks occurring simultaneously is very small. Under the proposed scheme there is a non-zero probability of a high criticality task missing its deadline. If the probability is less than the failure rate specification of the criticality level, see for example Table 4.1, then the MC system can still be schedulable according to the probabilistic bounds on deadline misses.

Individual tasks are assumed to be periodic with constrained

deadlines. The platform is assumed to have a single core. We assume a dual-critical model, where the criticality of tasks can be either LO or HI. The system is also assumed to have two modes of operation: LO- and HI-criticality mode. In the LO-criticality mode, all tasks are executed normally. In the HI-criticality mode, newly released jobs of LO tasks are started in a degraded mode so that preference is given to HI tasks.

The application of stochastic execution to MC systems is not new and several recent works exist [MDCGE17, Mas16, GSY15]. However, existing results do not provide a holistic analysis and scheduling scheme covering all execution modes and transitions. A detailed accounting of existing schemes and their limitations is given in Section 4.2. From now on, we suppose that a MC scheduling scheme fulfills the following requirements:

- A schedulability analysis of tasks is provided for each criticality level in each system mode.
- Conditions that should trigger a mode switch are defined.
- An analysis of the time spent in each system mode is provided.
- A method to consolidate these individual components and compute a metric comparable to the Probability of Failure per Hour for tasks of each criticality level is given.

In this chapter, we address all of these individual components. Specifically, we make the following contributions:

1. We propose conditions that trigger a mode switch, both from LO- to HI-criticality mode (LO→HI), and from HI- to LO-criticality mode (HI→LO).
2. We formulate a detailed stochastic analysis of LO-criticality mode. Using the analysis, the *Probability of Deadline Miss per Hour* in this mode is computed for tasks of both criticality levels.
3. We provide a first stochastic analysis of HI-criticality mode. Using the analysis, the maximal time spent in HI-criticality mode is obtained, along with the *Probability of Deadline Miss per Hour* for tasks of both criticality levels. Also taken into account is the probability the system enters HI-criticality mode.

4. Using contributions **1.-3.**, we compute the overall *Probability of Deadline Miss per Hour* values for all tasks by consolidating the respective values for LO- and HI-criticality mode. This allows us to compare these probabilities with the permitted ones found in typical certification standards.
5. We determine the probability that a LO task is started in its degraded mode.

Due to these contributions, we claim that this is the first work which provides a system-wide approach to MC scheduling, while considering a stochastic model of task execution times.

Organization of this chapter. Section 4.2 highlights the related research in MC scheduling and in stochastic analysis. It also highlights the limitations of existing research which are addressed by this work. Section 4.3 states our system model. The model includes the task model and the model of the MC system. This is followed by Section 4.4, which states and explains important definitions and operations for stochastic analysis of systems with non-deterministic execution times. Section 4.5 covers the proposed analysis for getting *Probability of Deadline Miss per Hour* values, both for all LO and for all HI tasks. This section also has important intermediate results such as the duration of LO- and HI-criticality mode, and the probability of each event that causes a system mode switch. The complexity of the analysis is explored in Section 4.6. Results are covered in Section 4.7. In this section, we evaluate various schedulability metrics and design trade-offs for MC systems. Limitations of and extensions to our work are discussed in Section 4.8. Finally, a summary is given in Section 4.9.

4.2 Related Work

Vestal's paper [Ves07] is the first paper that presents the MC model, where safety-critical tasks have multiple WCET estimates with different levels of assurance. Based on the model, a preemptive fixed priority scheduling scheme for sporadic task sets is presented: static

mixed-criticality (SMC). In the widely examined dual-criticality case, hard guarantees are given to HI tasks, but LO jobs might miss their deadline if a HI job overruns its optimistic WCET. As well as this, a LO job is de-scheduled if it overruns its WCET.

Baruah et al. [BBD11b] introduced an important fixed priority scheduling scheme, adaptive mixed-criticality (AMC), which defines a system that can operate in different modes. The system starts in LO-criticality mode where all tasks are scheduled to execute according to their optimistic WCET estimates. If any job overruns its optimistic WCET, a switch to HI-criticality mode happens, where all LO tasks are de-scheduled. This way, HI tasks are guaranteed to meet their deadlines all the time, whereas LO tasks have this guarantee only in LO-criticality mode.

EDF scheduling has been adapted to Vestal's model as well. Baruah et al. [BBD+11a] propose a scheduling scheme for sporadic task sets based on EDF, called EDF-VD. In this scheme, the deadlines of all HI tasks are scaled down by a single scaling factor so that an overrun is detected early. Once an overrun is detected, the system enters HI-criticality mode where all LO tasks are de-scheduled. In this scheme, all tasks meet their deadlines if no optimistic WCET is overrun, while only HI tasks meet their deadlines if some of them are overrun. Ekberg and Yi [EY12, EY14] use demand-bound functions to scale the deadlines of HI tasks individually, by a heuristic search strategy. Deadlines are chosen so that the schedulability of the system is maximized. The LO- and HI-criticality mode model in this scheme is similar to the one used in [BBD+11a]. Huang et al. [HGST14] amend EDF-VD to include degraded service for low criticality tasks while the system is in HI-criticality mode. The paper also presents an upper bound on the duration of this mode. Park and Kim [PK11] present another EDF-based scheme, CBEDF. Here, high criticality tasks are always guaranteed to execute, while some guarantees are given to tasks of low criticality using offline empty slack location discovery. Vestal's model with two modes of operation was also investigated for time-triggered scheduling, most notably in Baruah and Fohler [BF11]. For a comprehensive overview of research into mixed-criticality, we refer the reader to the review by Burns and Davis [BD17], while for a discussion on the applicability of mixed-criticality systems to industry and its safety-

critical practices see Ernst and Di Natale [EDN16].

As for probabilistic MC systems, related work often models them with probabilistic worst-case execution time (pWCET) distributions, which are seen as extending Vestal's model such that each task has a large number of WCETs with various levels of confidence [BD17, DCG19b]. A pWCET distribution comes from either the randomness inherent in a system and its environment, or the lack of knowledge we have about a system, or possibly both [DBG17]. To derive these distributions, well established methods like static probabilistic timing analysis (Devgan and Kashyap [DK03]) or measurement based probabilistic timing analysis techniques (Cucu-Grosjean et al. [CGSH+12]) already exist. Ideally, modeling tasks with pWCET distributions removes every dependency between them, so any task-set can be analyzed as though all tasks have independent execution times. In practice, by using pWCET distributions, these dependencies are reduced but not removed completely. This still poses a significant problem in applying pWCET methodologies for real-time computing: every inter-dependency between execution times needs to be described and carefully taken into account. As it is a challenge to precisely bound dependencies, this often again introduces major pessimism. An approach based on Fréchet bounds, which can deal with every type of (including worst-case) task inter-dependency, is given by Ivers and Ernst [IE09]. For an extensive survey of timing analysis techniques, we refer the reader to Davis and Cucu-Grosjean [DCG19b]. In our work, we assume that tasks' execution times are modeled with independent random variables which are given, and these random variables can be seen as an abstraction of ideal pWCETs.

For the analysis of probabilistic MC systems, obtaining probabilistic response times is key. The survey on probabilistic schedulability analyses by Davis and Cucu-Grosjean [DCG19a] lists various approaches to response time analysis. This chapter builds mainly upon the work of Díaz et al. [DGK+02, DLG+04], as their analysis of real-time systems is pessimistic, even though execution times are strictly required to be independent of one another. Using probabilistic analysis, existing work often presents scheduling schemes where individual tasks have certain permissible deadline miss probabilities. Examples are Maxim et al. [MDCGE17] and Abdeddaïm

and Maxim [AM17], were SMC and AMC scheduling are adapted to a probabilistic MC model, demonstrating the improvement in schedulability. Masrur [Mas16] proposes a scheme with no mode switches, where LO tasks have a soft guarantee on meeting their deadline as well. Alahmad and Gopalakrishnan [AG16, AG18] use a Markov decision process to provide probabilistic guarantees to jobs, and also formulate an optimization problem that provides the scheduling policy. Santinelli et al. [SG15, SG18, SGG16] examine probabilistic MC systems by doing a sensitivity analysis, which focuses on the impact made by varying execution times. However, we observe that a holistic characterization of probabilistic mixed-criticality systems remains largely unexplored in the state-of-the-art. Deadline miss probabilities of individual jobs are often not aggregated into system-wide metrics, for example in [Mas16, MDCGE17]. We note that giving soft guarantees to individual tasks is not equivalent to guaranteeing a probability of deadline miss per hour. Another related work, Guo et al. [GSY15], analyzes a simple probabilistic model, where a HI task has just two WCETs and their corresponding probabilities of occurrence. Using the model, they propose a EDF-based scheduling algorithm which has an allowed probability of a timing fault happening system-wide. Finally, Küttler et al. [KRHV17] consider a model where some guarantees are available to tasks of lower criticality. They propose lowering the priorities of lower criticality tasks in certain modes of operation. Still, without characterizing the duration of modes, we believe that the impact of degradation of LO tasks can not be properly quantified. In this chapter, all the aforementioned limitations of the state-of-the-art are addressed.

4.3 System Model

We start this section with an informal overview of our system model, before precise definitions are presented. The model is an extension of Vestal's original model [Ves07], and as is with adaptive mixed-criticality [BBD11b], there are two modes of operation, LO- and HI-criticality mode.

LO-criticality mode can be considered a normal mode of operation, and the system is expected to operate in this mode most of the time. HI-criticality mode can be considered an emergency mode, where newly instantiated LO jobs are started and running in degraded mode so preference is given to the execution of HI jobs. More specifically, HI criticality tasks are not affected by the mode of operation, these task are always released and executed until their completion. LO criticality tasks have two variants: each LO job can be released in degraded or regular mode. They always finish in the mode they started with. Though LO tasks are never dropped, they are released with degradation when the system is in HI-criticality mode. In practice, this means that there are two implementations of each task, and the degraded variant offers a reduced functionality. For example, the numerical result is computed with less precision. Vestal's original model specifies dropping LO jobs when HI jobs need more resources, and our model can be seen as a generalization where not executing a job is the extreme case.

The system starts in LO-criticality mode, and remains there until a mode switching event occurs. The first mode switching event is the only one discussed for non-probabilistic MC systems, and is thus found in previous work, for example [BBD11b, EY14, HGST14, MDCGE17]: a HI job's execution lasts longer than a provided threshold. The second mode switching event is when a HI job misses its deadline. It is introduced to reduce the probability of consecutive deadline misses of HI jobs. Note that a HI job might miss its deadline without overrunning its threshold execution time, for example because it was blocked by jobs of higher priority. Finally, the third mode switch event is when a long backlog of LO jobs accumulates, which could in turn produce an arbitrarily high backlog when entering HI mode. Once in HI-criticality mode, the system switches back to LO-mode the first time it is idle.

Using this model, we say a task-set to be schedulable using fixed priority preemptive scheduling, if the probability that any job misses its deadline during an hour of operation is sufficiently small, and if the ratio of LO jobs released in degraded mode is acceptable.

General notation on random variables. This work deals with discrete random variables, and they are denoted using calligraphic

symbols, for example $\mathcal{A}$. The probability function of $\mathcal{A}$, noted $p_{\mathcal{A}}(\cdot)$, tells us the probability that $\mathcal{A}$ takes a specific value u: $p_{\mathcal{A}}(u) = \Pr(\mathcal{A} = u)$. Without loss of generality, we assume that the possible values of all random variables span the full range of natural numbers. If the maximal and minimal values with non-zero probability of $\mathcal{A}$ exist, and are noted $u_{\max}$ and $u_{\min}$, then the probability function can be represented in vector notation:

$$p_{\mathcal{A}} = [p_{\mathcal{A}}(u_{\min}), \dots p_{\mathcal{A}}(u_{\max})]^{\intercal}$$

Let us define a relation to compare two random variables $\mathcal{A}$ and $\mathcal{B}$, as was done by Díaz et al [DGK+02].

Definition 4.1 *First-Order Stochastic Dominance*: $\mathcal{A}$ is greater or equal than $\mathcal{B}$, written as $\mathcal{A} \succeq \mathcal{B}$, if and only if

$$\forall l \geq 0 : \quad \sum_{u=l}^{\infty} p_{\mathcal{A}}(u) \geq \sum_{u=l}^{\infty} p_{\mathcal{B}}(u) \tag{4.1}$$

Note that probability densities can be incomparable.

We introduce a shorthand notation for the probability that a variable modeled with random variable $\mathcal{A}$ has a value greater than scalar s. Instead of the cumbersome expression $\sum_{s<i} \Pr(i = \mathcal{A})$, we use $\Pr(s < \mathcal{A})$.

Finally, we introduce a simple notation $[s]^1$ to indicate that a scalar or expression s is limited to a maximum value of 1, $[s]^1 = \min(s, 1)$.

Task model. We take that a task-set Π consists of N independent tasks. Each task is periodic, constrained deadline, with an initial phase and a criticality level. A single task τ_i is characterized by tuple $(T_i, D_i, \phi_i, \chi_i, \mathcal{C}_i)$, where T_i is the period, D_i is the relative deadline, ϕ_i is the phase, $\chi_i \in \{\text{LO}, \text{HI}\}$ is the task's criticality level, and $\mathcal{C}_i$ models the probabilistic execution time. $\mathcal{C}_i$ has a maximal value with non-zero probability, which is the WCET, noted $C_i^{\max}$. Tasks with criticality level LO and HI are referred to as 'LO tasks' and 'HI tasks', respectively. An instance j of task τ_i is called a job, and denoted as $\tau_{i,j}$. Each job $\tau_{i,j}$ has a release time $r_{i,j} = \phi_i + (j-1) \cdot T_i$, and an absolute deadline $d_{i,j} = r_{i,j} + D_i$. The hyperperiod HP of

a set of tasks is defined to be the least common multiple of all task periods.

We model the execution times of each task τ_i with known independent and identically distributed random variables $\mathcal{C}_i$. This means that there is no dependency between the execution times of any two jobs, regardless of whether they are of the same task or not, and execution times of all jobs of one task are modeled with the same random variable. Independence is an important requirement for our provided analysis, otherwise results would not be worst-case. However, the provided analysis is safe, i.e., if the computed bounds hold for a given set of probabilistic execution times, they also hold if the execution times are smaller or equal according to Definition 4.1. Therefore, the probabilistic execution times $\mathcal{C}_i$ can also be regarded as ideal probabilistic worst-case execution times, which would remove the requirement that execution times of jobs are independent.

In the standard MC model [Ves07], HI tasks have an optimistic and a pessimistic WCET estimate, and LO tasks are executed by the processor only if HI tasks meet their optimistic WCET estimates during operation. The reasoning behind this is the assumption that most of the time HI tasks will not execute for longer than their optimistic WCET estimate, so less computational resources are needed for the correct operation of the system. In this work, we assume that the distribution of the execution time of each task $\mathcal{C}_i$ is known. Therefore, instead of the optimistic WCET estimate, for each HI task we define a threshold execution time value C_i^{thr}. We assume this value is a given design choice. Note that the probability that a HI task executes for longer than this threshold is $\Pr(\mathcal{C}_i > C_i^{\text{thr}})$. The precise way this threshold is used in scheduling of jobs is described later in this section. Additionally, instead of not executing LO jobs in order to free up resources, we introduce that each LO job can be released in degraded or regular mode. If it executes with degradation, its WCET is C_i^{deg}. The C_i^{deg} value is assumed to given as a design choice. It could be zero if the task is not to be run in HI-criticality mode, or it can be any value less than its WCET: in this case it is assumed that a lower functionality is provided.

For the execution time of HI tasks, it is useful to introduce the following random variable that describes a worst-case behavior as

long as the analyzed system is still in LO-critical mode.

Definition 4.2 *Trimmed Execution Time*: Random variable $\mathcal{C}_i^{\text{LO}}$ models the execution time of HI tasks τ_i, but modified such that they do not execute for longer than C_i^{thr}:

$$p_{\mathcal{C}_i^{\text{LO}}}(u) = \begin{cases} p_{\mathcal{C}_i}(u) & u < C_i^{\text{thr}} \\ \sum_{v=C_i^{\text{thr}}}^{C_i^{\max}} p_{\mathcal{C}_i}(v) & u = C_i^{\text{thr}} \\ 0 & u > C_i^{\text{thr}} \end{cases} \tag{4.2}$$

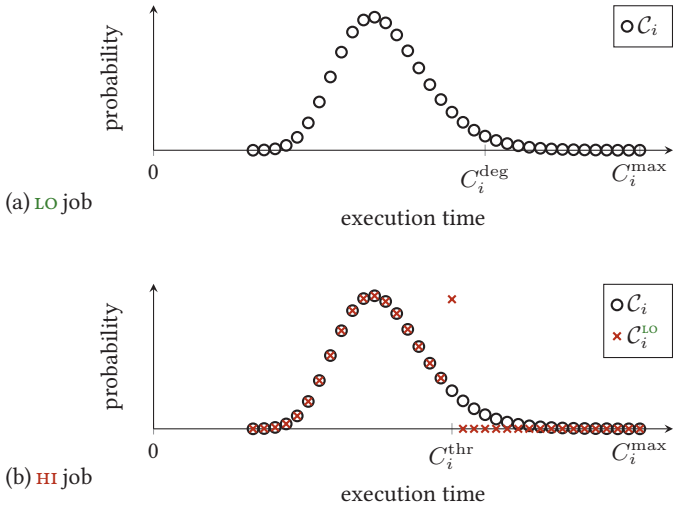
Figure 4.1a illustrates the $\mathcal{C}_i$ of a LO task, as well as the WCET denoted as C_i^{deg} in degraded mode. Figure 4.1b illustrates the $\mathcal{C}_i$ of a HI task as well as the trimmed execution time $\mathcal{C}_i^{\text{LO}}$ with the corresponding C_i^{thr} and $C_i^{\max}$ values.

This definition differs from the one found in many related works, i.e., [MDCGE17], where the execution time of HI tasks in LO-critical mode is defined as the conditional probability $\Pr(p_{\mathcal{C}_i}(u) = u | u \leq C_i^{\text{thr}})$, often called 'truncated' execution time. The 'trimmed' execution times, as defined here, are by definition greater or equal to the equivalent 'truncated' execution times. We use 'trimmed' execution times because they simplify the analysis of HI-criticality mode, namely by simplifying initial conditions noted by Definition 4.12. The cost of this simplification is that it introduces pessimism in the LO-criticality mode analysis, however this has been found to be numerically negligible through simulations. Nevertheless, using the 'truncated' execution times option with a more complex analysis is also possible. For more information, see the Section 4.8.1.

The response time of job $\tau_{i,j}$ is modeled with random variable $\mathcal{R}_{i,j}$. The way this variable can be obtained and upper-bounded is presented in Section 4.4. The deadline miss probability of job $\tau_{i,j}$ is the probability that this job finishes after its deadline $\text{DMP}_{i,j} = \Pr(\mathcal{R}_{i,j} > d_{i,j})$.

Schedulability. In this chapter, we consider a single-core platform. A simple execution model is used, where task preemption overhead is zero.

As in the standard MC model, the system is defined to operate in two modes of operation, LO- and HI-criticality mode. When the



(a) LO job

(b) HI job

Figure 4.1: Task execution times, with named values and trimmed execution time C_i^{LO}

system is operating in LO-criticality mode, both LO and HI jobs are released. When the system is operating in HI-criticality mode, HI jobs are released normally, while LO jobs are released in degraded mode.

The definition of schedulability is inspired by the probability-of-failure-per-hour notion. Therefore, we first define the probability of deadline miss per hour, before defining schedulability. We also define the probability of degraded job, a proportion of how many LO jobs execute in degraded mode in the long run.

Definition 4.3 *Failure Probabilities*: The *probability of deadline miss per time interval T for* HI *or* LO *jobs* is denoted as $\text{DMP}_{\text{HI}}(T)$ or $\text{DMP}_{\text{LO}}(T)$, respectively. It is the probability that at least one HI or LO job misses its deadline during a time interval of length T. Formally, we define $\text{DMP}_{\text{HI}}(T)$ and $\text{DMP}_{\text{LO}}(T)$ as:

$$\text{DMP}_{\chi}(T) = \max_{\forall t} \Pr\left(\exists \tau_{i,j} \in S_{\chi}(t) \ : \ \tau_{i,j} \text{ misses its deadline}\right) \tag{4.3}$$

where $\chi = \{\text{LO}, \text{HI}\}$, and $S_\chi(t) = \{\tau_{i,j} \,|\, \chi_i = \chi \wedge t \leq r_{i,j} < t+T\}$.

Definition 4.4 *Probability of Degraded Job*: The *probability of degraded* LO *jobs* PDJ_{deg} is the probability that any individual LO job is released in degraded mode:

$$\text{PDJ}_{\text{deg}} = \max_{\forall t} \frac{|S_{\text{LO-deg}}(t)|}{|S_{\text{LO}}(t)|} \tag{4.4}$$

where $S_{\text{LO}}(t) = \{\tau_{i,j} \,|\, \chi_i = \text{LO} \wedge t \leq r_{i,j} < t+T\}$ and $S_{\text{LO-deg}}(t) = \{\tau_{i,j} \mid \chi_i = \text{LO} \ \wedge \ t \leq r_{i,j} < t+T \ \wedge \ \tau_{i,j}$ is in degraded mode$\}$.

Definition 4.5 *Schedulability*: A mixed-criticality (MC) system is $(\sigma_{\text{HI}}, \sigma_{\text{LO}}, \sigma_{\text{deg}})$-schedulable if $\text{DMP}_{\text{HI}}(1\,\text{h}) \leq \sigma_{\text{HI}}$, $\text{DMP}_{\text{LO}}(1\,\text{h}) \leq \sigma_{\text{LO}}$, and $\text{PDJ}_{\text{deg}} \leq \sigma_{\text{deg}}$, where $1h$ denotes the duration of one hour.

The probabilistic MC scheduling scheme used in our work can now be defined:

Definition 4.6 *Probabilistic MC Scheduling*: In LO-criticality mode, all tasks are scheduled using a provided fixed-priority preemptive schedule. The system starts in LO-criticality mode, and remains in it until one of the following events causes a transition to HI-criticality mode:

1. A HI job overruns its threshold execution time C_i^{thr}.
2. A HI job misses its deadline.
3. The system-level backlog, meaning the amount of pending execution, becomes higher than a predefined threshold $B_{\max}$.

In HI-criticality mode, the same fixed-priority preemptive schedule is used, but LO jobs are released with degradation in order to free up the processor. LO jobs starting in LO-criticality mode are still continuing in their normal mode with execution time $\mathcal{C}_i$. The system remains in HI-criticality mode until it becomes idle for the first time.

4.4 Preliminaries

With tasks having probabilistic execution times, a set of computational primitives are required to perform the schedulability analysis.

A probabilistic analysis of real-time systems, on which our analysis is based, was described by Díaz et al. [DGK+02, DLG+04]. We summarize the analysis technique in this section. The analysis and its primitives are used extensively in the following sections to perform the schedulability analysis of mixed-criticality systems. Note that this analysis requires execution times to be independent from one another. More general approaches that take stochastic dependencies between tasks exist but are beyond the scope of this paper, see Ivers and Ernst [IE09].

The analysis requires computation of the *backlog*, i.e., the sum of pending execution times of all ready jobs. For each priority level i there is a backlog containing the execution times of all pending jobs with priority i or higher. When a new job with priority i arrives, all backlogs with level i or lower are increased by adding its execution time. Adding the execution time random variable to a backlog is done using *convolution*. Executing a job decreases the backlogs of all levels i that are equal or smaller than the priority of the job. Decreasing the backlog is done using *shrinking*.

Definition 4.7 *Backlog*: The i^{th} priority backlog at time t, $\mathcal{B}_i(t)$, is a random variable that describes the sum of all remaining execution times of pending jobs of priority not less than i, at time t. The backlog $\mathcal{B}_i(t-)$ is the same as $\mathcal{B}_i(t)$, except it does not take into account jobs released at time t.

Using convolution to compute backlog after arrival of a job. Suppose that a job $\tau_{i,j}$ is released at time $r_{i,j}$, and $\mathcal{B}_k(r_{i,j}-)$ is the k^{th} priority backlog at time $r_{i,j}$, but excludes the newly released job. Assuming that $i \geq k$, and that no other job is released at the same time, backlog $\mathcal{B}_k(r_{i,j})$ can be computed using the convolution operator $\otimes$:

$$p_{\mathcal{B}_k(r_{i,j})} = p_{\mathcal{B}_k(r_{i,j}-)} \otimes p_{\mathcal{C}_i} \tag{4.5}$$

Backlog reduction due to execution of highest priority job. Let us assume that in the interval $t_0 < t < t_1$ there are no job arrivals. During this interval, the backlog is decreased as the processor executes pending jobs. If $\mathcal{B}_i(t_0)$ is the i^{th} priority backlog at time t_0, the corresponding backlog at time t can be computed

using the so-called *shrinking* operation. Specifically, for computing backlog at time $t_0 < t < t_1$, the following equation can be used:

$$p_{\mathcal{B}_i(t)}(u) = \begin{cases} \sum_{j=0}^{t-t_0} p_{\mathcal{B}_i(t_0)}(j) & u = 0 \\ p_{\mathcal{B}_i(t_0)}(u + t - t_0) & u > 0 \end{cases} \tag{4.6}$$

In other words, the backlog after an execution of $t - t_0$ time units is computed by left-shifting the initial backlog by $t - t_0$, while truncating at zero since the processor is idle when no pending execution is present. For brevity, we define the corresponding shrinking function of a random variable $\mathcal{B}$:

$$\text{shrink}(\mathcal{B}, m)(u) = \begin{cases} \sum_{j=0}^{m} p_{\mathcal{B}}(j) & u = 0 \\ p_{\mathcal{B}}(u + m) & u > 0 \end{cases} \tag{4.7}$$

Backlog state space exploration. First, we define the function `bsse` for computing the backlog at some time $t+u$ given the backlog at time t.

Definition 4.8 *Backlog Computation*: $\texttt{bsse}\,(\mathcal{B}_i(t),\, \Pi,\, i,\, t,\, u)$ is a function for computing the i^{th} priority backlog at time $t + u$, i.e., $\mathcal{B}_i(t+u)$. We assume that the i^{th} priority backlog at time t is $\mathcal{B}_i(t)$, and that the task arrivals and execution times in the interval $[t, t+u)$ are in accordance with task set Π.

The computation of `bsse` can be done by applying the definition of a task set as well as the previously described operations, namely convolution and shrinking. We demonstrate this using the following example.

Example. *Task-set* Π *is given, and consists of task* $\tau_1 = (T_1 = 5,\, \phi_1 = 0,\, D_1 = 5,\, \mathcal{C}_1)\}$*, and of task* $\tau_2 = (10,\, 0,\, 10,\, \mathcal{C}_2)$*. Task* τ_2 *has a higher priority. The backlogs at time* $0-$ *at priority levels 1 and 2 are given as* $\mathcal{B}_1(0-)$ *and* $\mathcal{B}_2(0-)$*, respectively. For this set-up, find the backlog at time* $10-$ *at priority level 1, as well as the backlog at time* 7 *at priority level 2.*

Solution. The following combination of convolution and shrinking computes $\mathcal{B}_1(10-) = \texttt{bsse}\,(\mathcal{B}_1(0-), \Pi, 1, 0-, 10-)$, by taking into account the execution times of all jobs:

$$\begin{aligned} p_{\mathcal{B}_1(0)} &= p_{\mathcal{B}_1(0-)} \otimes p_{\mathcal{C}_1} \otimes p_{\mathcal{C}_2} \\ p_{\mathcal{B}_1(5-)} &= \text{shrink}(\mathcal{B}_1(0), 5) \\ p_{\mathcal{B}_1(5)} &= p_{\mathcal{B}_1(5-)} \otimes p_{\mathcal{C}_1} \\ p_{\mathcal{B}_1(10-)} &= \text{shrink}(\mathcal{B}_1(5), 5) \end{aligned}$$

For computing the highest priority backlog, task τ_1 is ignored. Using the same procedure, we obtain $\mathcal{B}_2(7)$:

$$\begin{aligned} p_{\mathcal{B}_2(0)} &= p_{\mathcal{B}_2(0-)} \otimes p_{\mathcal{C}_2} \\ p_{\mathcal{B}_2(7)} &= \text{shrink}(\mathcal{B}_2(0), 7) \end{aligned}$$

Which is equivalent to $\texttt{bsse}\,(\mathcal{B}_2(0-), \Pi, 2, 0-, 7)$.

4.4.1 Upper Bound of Backlog

In order to provide a holistic schedulability analysis, we need to determine upper bounds of the backlogs for all time instances within any future hyperperiod, i.e., we are interested in a set of random variables $\overline{\mathcal{B}}_i(t)$ such that $\mathcal{B}_i(n \cdot \text{HP} + t) \preceq \overline{\mathcal{B}}_i(t)$ for all priority levels i, future hyperperiods $n \geq 0$ and time instances within a hyperperiod $0 \leq t < \text{HP}$. We start by computing the steady-state backlog and proceed by showing that it provides the desired upper bound.

Computation of the steady state backlog. The i^{th} priority backlog at the start of the n^{th} hyperperiod is $\mathcal{B}_i(n \cdot \text{HP})$, but this backlog may be different for each n. However, the sequence of random variables $\{\mathcal{B}_i(n \cdot \text{HP})\}$ can be viewed as a Markov process as shown by Díaz et al. [DGK+02]. Specifically, they present the following theorem about the existence of a limit to the above mentioned sequence, including the corresponding proof:

Theorem 4.1 (section 4.2 of [DGK+02]): *The sequence of backlogs* $\{\mathcal{B}_i(n \cdot \text{HP})\}$ *for* $n \geq 0$*, where* i *is a priority level, has a limit if the*

average system utilization is less than one, and if the sequence of jobs remains the same each hyperperiod. If it exists, this limit is called the i^{th} priority steady state backlog at the beginning of the hyperperiod, and noted $\overline{\mathcal{B}}_i(0)$.

Proof Sketch: We provide a sketch of the proof here, while the reader can find all the details in the original work. The backlog at the start of k^{th} hyperperiod can be computed using backlog at the start of $(k-1)^{\text{th}}$ hyperperiod using

$$\texttt{bsse}\left(\mathcal{B}_i((k-1)\cdot \text{HP}-),\ \Pi,\ i,\ (k-1)\cdot \text{HP}-,\ \text{HP}\right)$$

As stated, all hyperperiods have the same sequence of jobs. Consequently, the following equation in vector form can be used for computing the backlog at the start of k^{th} hyperperiod:

$$p_{\mathcal{B}_i(k\cdot \text{HP})}{}^{\intercal} = \boldsymbol{P}\cdot p_{\mathcal{B}_i((k-1)\cdot \text{HP})}{}^{\intercal} \tag{4.8}$$

where $\boldsymbol{P}$ is a transition matrix. Note that our model does not place any upper bounds on the backlog's length, so $\boldsymbol{P}$ can have an infinite dimension. Since $\boldsymbol{P}$ is invariant in every hyperperiod, we can use known results from Markov theory: $\overline{\mathcal{B}}_i(0)$ exists and is a non-degenerate distribution (i.e., not infinite) if the average system utilization is less than one. □

For computing the steady state backlog at the start of a hyperperiod $\overline{\mathcal{B}}_i(0)$, Díaz et al. propose three methods. The first method is an exact one stated in section 4.3.2 of [DGK+02] and exploits the structure of the infinite dimension transition matrix $\boldsymbol{P}$. A second method (section 4.3.3 of [DGK+02]) finds an approximate value of $\overline{\mathcal{B}}_i(0)$ by truncating $\boldsymbol{P}$ to make its dimension finite. Finally, a third method is to iterate over hyperperiods until the following relaxed steady state condition is satisfied:

$$\max_{i,x}\left\{\left|p_{\mathcal{B}_i(k\cdot \text{HP})}(x) - p_{\mathcal{B}_i((k-1)\cdot \text{HP})}(x)\right|\right\} < \epsilon \tag{4.9}$$

This condition states that the maximum difference between all i^{th} priority backlogs must not exceed a configurable small value ϵ. This method does not require computation nor truncation of the transition matrix $\boldsymbol{P}$. For further details on choosing appropriate initial backlogs, please refer to Díaz et al. in [DLG+04].

Pessimism of the steady state backlog. Assuming that the initial backlog is zero at every priority level, and that the sequence of jobs remains the same each hyperperiod, it has been shown by Díaz and López [DL04] that the i^{th} priority steady state backlog is an upper bound to all i^{th} priority backlogs at the start of the hyperperiod. The following two lemmas can be used to show that the backlogs at the beginning of a hyperperiod are increasing from hyperperiod to hyperperiod. They state that the operations of convolution and shrinking preserve the partial ordering of random variables.

Lemma 4.1 (Property 3 in [DL04]): *Given three positive random variables $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$. If $\mathcal{A} \preceq \mathcal{B}$, then $\mathcal{A} + \mathcal{C} \preceq \mathcal{B} + \mathcal{C}$.*

Lemma 4.2 (Property 6 in [DL04]): *Given two positive random variables $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$. If $\mathcal{A} \preceq \mathcal{B}$, then* $\text{shrink}(\mathcal{A}, m) \preceq \text{shrink}(\mathcal{B}, m)$.

Now, the following theorem can be shown by means of the above considerations: we have, by definition,

$$\overline{\mathcal{B}}_i(t) = \lim_{n \to \infty} \mathcal{B}_i(t + n \cdot \text{HP})$$

for all $n \geq 0$ and $0 \leq t < \text{HP}$, and we know from Theorem 4.1 that $\mathcal{B}_i(n \cdot \text{HP}) \preceq \overline{\mathcal{B}}_i(0)$ for all $n \geq 0$.

Theorem 4.2 (Theorem 1 in [DL04]): *Assuming that the initial backlog is zero, and that the sequence of jobs remains the same each hyperperiod, the i^{th} priority backlog at time t inside every hyperperiod is upper bounded by the i^{th} priority steady state backlog at time t inside the hyperperiod:*

$$\begin{aligned} &\forall i : \; p_{\mathcal{B}_i(0)}(0) = 1 \quad \Rightarrow \\ &\forall t \in [0, \text{HP}), \; \forall n \in \mathbb{N} : \; \mathcal{B}_i(n \cdot \text{HP} + t) \preceq \overline{\mathcal{B}}_i(t) \end{aligned} \tag{4.10}$$

Proof Sketch: $\overline{\mathcal{B}}_i(t)$ can be computed using $\mathcal{B}_i(0)$ and a finite number of convolution and shrinking operations. The theorem thus follows from Lemma 4.1, Lemma 4.2, and the fact that the initial backlog is zero. □

In summary, if the initial backlog is zero, the steady-state backlog $\overline{\mathcal{B}}_i(t)$ provides an upper bound for all backlogs within any future hyperperiod. This result is used extensively in the the response time analysis described next.

Algorithm 4.1: Computing the response time of a job

```
1  procedure rta(B_i(r_{i,j}), Π, τ_{i,j})
2      R_{i,j} ← B_i(r_{i,j}) ⊗ C_i
3      t ← 0
4      while t ≤ D_{i,j} do
5          for each preempting job τ_{k,l} that arrives at r_{i,j} + t do
6              {R_l, R_u} ← split(R_{i,j}, t)
7              p_{R_{i,j}}(0, ..., t − 1) ← R_l
8              p_{R_{i,j}}(t, ...) ← R_u ⊗ C_k
9          t ← t + 1
10     return R_{i,j}
11 function split(X, m)
12     X_l ← [p_X(0), p_X(1), ... p_X(m − 1)]
13     X_u ← [p_X(m), p_X(m + 1), ... p_X(X̂)]
14     return X_l, X_u
```

4.4.2 Response Time Analysis

The response time of a job $\mathcal{R}_{i,j}$ tells us when this job will finish its execution, relative to its release time. We summarize the procedure as proposed by Díaz et al. [DGK+02]. The response time of a given job $\tau_{i,j}$ is influenced by the backlog at its release time $\mathcal{B}_i(r_{i,j})$, and the computation times of all jobs that preempt the job. Therefore we can define a function:

$$\mathcal{R}_{i,j} = \texttt{rta}\left(\mathcal{B}_i(r_{i,j}),\, \Pi,\, \tau_{i,j}\right) \tag{4.11}$$

The pseudo-code for computing response times is given in Algorithm 4.1. For a given job $\tau_{i,j}$, first $\mathcal{C}_i$ is convolved with the the current i^{th} priority backlog (line 2). This would provide us with the response time of $\tau_{i,j}$, if there were no preempting jobs. When a preempting job is released at a given point in time, then the probability function vector of $\tau_{i,j}$'s response time is split in two portions (line 6): the part before preemption (R_l), and the part after preemption (R_u). The part after preemption is convolved with the probability function vector of the preempting job's computation time, and the result is added to R_l in order to get $\tau_{i,j}$'s response time after this preemption (lines 7 and 8). The probability function of $\mathcal{R}_{i,j}$ is only computed until the job's deadline $d_{i,j}$.

Next, we present a theorem that we use to obtain the worst-case hourly deadline miss probability. Beforehand, the lemma shows that the response time function $\texttt{rta}$ is monotone in the backlog at the release time of the job.

Lemma 4.3: *(Theorem 1, property 3 of [LDEG08]) Given two random variables $\mathcal{A}$ and $\mathcal{B}$. If $\mathcal{A} \preceq \mathcal{B}$, then $\texttt{rta}\left(\mathcal{A}, \Pi, \tau_{i,j}\right) \preceq \texttt{rta}\left(\mathcal{B}, \Pi, \tau_{i,j}\right)$.*

As the steady-state backlog at any time within a hyperperiod is always greater than or equal to the backlog at the corresponding time within any hyperperiod, the following lemma can be obtained.

Lemma 4.4: *Assuming the initial backlog is zero, substituting any backlog $\mathcal{B}_i(r_{i,j})$ with the appropriate steady state backlog $\overline{\mathcal{B}}_i(r_{i,j})$ in the response time analysis, produces a value greater or equal to the response time.*

$$\begin{aligned} &\forall i: \; p_{\mathcal{B}_i(0)}(0) = 1 \quad \Rightarrow \\ &\texttt{rta}\left(\mathcal{B}_i(r_{i,j}), \Pi, \tau_{i,j}\right) \preceq \texttt{rta}\left(\overline{\mathcal{B}}_i(r_{i,j} \bmod \text{HP}), \Pi, \tau_{i,j}\right) \end{aligned} \tag{4.12}$$

Proof: This lemma is a direct consequence of Lemma 4.3 and Theorem 4.2. □

We name the value $\texttt{rta}\left(\overline{\mathcal{B}}_i(r_{i,j} \bmod \text{HP}), \Pi, \tau_{i,j}\right)$ the steady state response time, and denote it $\overline{\mathcal{R}}_{i,j}$. Note that the steady-state backlog $\overline{\mathcal{B}}_i$ was used to do this calculation of the upper bound of the response time. Based on these results, we can now determine an upper bound on the response time of each job. By focusing on one hyperperiod and obtaining steady-state (worst case) results for it, we can finally determine the worst-case deadline miss probability of a job $\tau_{i,j}$ within any hyperperiod. Instead of using the modulo operation as in Lemma 4.4 we can also just look at jobs $\tau_{i,j}$ within the single worst case hyperperiod with $0 \leq j < \text{HP}/T_i$.

Theorem 4.3: *The deadline miss probability of a job $\tau_{i,j}$ denoted as $\text{DMP}_{i,j}$ can be bounded as follows:*

$$\begin{aligned} &\forall i, 0 \leq j < \text{HP}/T_i \; : \\ &\text{DMP}_{i,j} \leq \overline{\text{DMP}}_{i,j} = \Pr\left(d_{i,j} < \texttt{rta}\left(\overline{\mathcal{B}}_i(r_{i,j}), \Pi, \tau_{i,j}\right)\right) \end{aligned} \tag{4.13}$$

Proof: The proof follows directly from the previous Lemma 4.4. □

4.5 Stochastic Analysis

In this section, we determine the $(\sigma_{\text{HI}}, \sigma_{\text{LO}}, \sigma_{\text{deg}})$-schedulability of a mixed-critical task set Π as defined in Definition 4.5. We refer to this as the probabilistic mixed-criticality (pMC) analysis. To this end, we compute upper bounds on probabilities that there is at least one deadline miss of a LO or HI job within one hour, i.e., $\text{DMP}_{\text{HI}}(T)$ or $\text{DMP}_{\text{LO}}(T)$, respectively, for a time interval of length $T = 1h$. In addition, we will compute an upper bound on the probability that a LO job operates in degraded mode PDJ_{deg}. The underlying concept of the forthcoming analysis is described next.

Let us start with the computation of the probability PDJ_{deg} that a LO job operates in degraded mode. This probability can be upper bounded by noting that LO jobs are executed only in their degraded mode if their release time $r_{i,j}$ happens during HI-criticality mode. Therefore, we will first determine the maximal length $\Delta^{\text{HI}}_{\text{max}}$ of any HI-criticality mode execution. In addition, we determine an upper bound on the probability, that there is at least one mode switch within a single hyperperiod, denoted as $\text{P}^{\text{HP}}_{\text{LO}\to\text{HI}}$. Using these two values, we can bound the relative time the system is in HI mode and therefore, the probability that a LO job operates in degraded mode.

To determine upper bounds on probabilities $\text{DMP}_{\text{HI}}(1h)$ and $\text{DMP}_{\text{LO}}(1h)$, that there is at least one deadline miss of a LO or HI job within one hour, we first look at upper bounds on the probabilities that at least one LO or HI job misses its deadline during any HI-criticality mode execution that is started within a hyperperiod, denoted as $\text{DMP}^{\text{HI}}_{\text{HI}}$ or $\text{DMP}^{\text{HI}}_{\text{LO}}$, respectively. Note that the upper index denotes the mode, whereas the lower one denotes the criticality of the jobs we are considering. In addition, we determine an upper bound on the probability that at least one LO or HI job misses its deadline during a hyperperiod under the conditions that first, no mode switches take place and second, HI jobs do not overrun their threshold C^{thr}. We denote these values as $\text{DMP}^{\text{LO}}_{\text{HI}}$ or $\text{DMP}^{\text{LO}}_{\text{LO}}$, respectively. Again, the upper index concerns the mode and the lower one the criticality of the considered jobs. Now we can determine the desired probabilities $\text{DMP}_{\text{HI}}(T)$ and $\text{DMP}_{\text{LO}}(T)$ by combining (a) the worst case probabilities $\text{DMP}^{\text{LO}}_{\text{HI}}$ and $\text{DMP}^{\text{LO}}_{\text{LO}}$ that a deadline miss happens during a hyperperiod if the system is

in LO-criticality mode, (b) the worst case probabilities $\mathrm{DMP}^{\mathrm{HI}}_{\mathrm{HI}}$ and $\mathrm{DMP}^{\mathrm{HI}}_{\mathrm{LO}}$ that at least one LO or HI job misses its deadline during any HI-criticality mode started within a hyperperiod.

We now first determine bounds $\mathrm{PDJ}_{\mathrm{deg}}$ and $\mathrm{DMP}_{\chi}(1h)$ using the above defined quantities: $\Delta^{\mathrm{HI}}_{\mathrm{max}}$, $\mathrm{P}^{\mathrm{HP}}_{\mathrm{LO}\to\mathrm{HI}}$, $\mathrm{DMP}^{\mathrm{HI}}_{\chi}$ and $\mathrm{DMP}^{\mathrm{LO}}_{\chi}$ for HI and LO jobs, i.e., for $\chi \in \{\mathrm{LO}, \mathrm{HI}\}$. Afterwards, we explain how these quantities can be determined.

4.5.1 Probability of Job Degradation

In this section, we compute an upper bound on the probability that a LO job operates in degraded mode, i.e., $\mathrm{PDJ}_{\mathrm{deg}}$. As described above, we will make use of the maximal duration of a HI-criticality mode execution and the probability that there is no mode switch within a hyperperiod.

Definition 4.9 *Maximal Duration of* HI*-Criticality Mode*: The maximal duration the system is continuously executing in HI-criticality mode is denoted $\Delta^{\mathrm{HI}}_{\mathrm{max}}$.

Definition 4.10 *Mode Switch Probability*: $\mathrm{P}^{\mathrm{HP}}_{\mathrm{LO}\to\mathrm{HI}}$ denotes an upper bound on the probability that there is at least one mode switch LO→HI within a single hyperperiod.

Using these definitions, we can determine an upper bound on the desired quantity.

Theorem 4.4: *The probability of degradation of a* LO *job can be bounded as follows:*

$$\mathrm{PDJ}_{\mathrm{deg}} \leq \left\lceil \frac{\Delta^{\mathrm{HI}}_{\mathrm{max}}}{\mathrm{HP}} + 1 \right\rceil \mathrm{P}^{\mathrm{HP}}_{\mathrm{LO}\to\mathrm{HI}} \tag{4.14}$$

Proof: We obtain this value by multiplying the probability that HI-criticality mode is entered during one hyperperiod, with the number of LO jobs that are released in degraded mode when it does.

First, note that there is some constant number K of LO jobs that are released every hyperperiod. From the moment one HI-criticality mode is entered, it executes at least partly in at most $\lceil 1 + \Delta^{\mathrm{HI}}_{\mathrm{max}}/\mathrm{HP} \rceil$ hyperperiods. Therefore, what ever the number of mode switches is

inside one hyperperiod, in the worst case, all LO jobs from this and the next $\lceil \Delta^{\text{HI}}_{\max}/\text{HP} \rceil$ hyperperiods are executed in degraded mode. In other words, $K \cdot \left\lceil \frac{\Delta^{\text{HI}}_{\max}}{\text{HP}} + 1 \right\rceil$ LO jobs are degraded.

Second, let us note that there is at least one mode switch within a hyperperiod with probability $\text{P}^{\text{HP}}_{\text{LO}\to\text{HI}}$. Combining this probability with the number of LO jobs that are degraded if a mode switch happens, we get:

$$\begin{aligned}\text{PDJ}_{\text{deg}} &\leq \left(K \cdot \left\lceil \frac{\Delta^{\text{HI}}_{\max}}{\text{HP}} + 1 \right\rceil \text{P}^{\text{HP}}_{\text{LO}\to\text{HI}} + 0 \cdot (1 - \text{P}^{\text{HP}}_{\text{LO}\to\text{HI}}) \right) K^{-1} \\ &= \left\lceil \frac{\Delta^{\text{HI}}_{\max}}{\text{HP}} + 1 \right\rceil \text{P}^{\text{HP}}_{\text{LO}\to\text{HI}}\end{aligned}$$

□

This upper bound on the probability of degradation of a LO job may be overly pessimistic in the case when the hyperperiod is much larger than the maximal duration of HI-criticality mode, $\text{HP} \gg \Delta^{\text{HI}}_{\max}$. Still, in practical scenarios, it is not considered usual practice to design a system with a very long hyperperiod. We therefore accept the upper bound as satisfactory.

The necessary quantities $\Delta^{\text{HI}}_{\max}$ and $\text{P}^{\text{HP}}_{\text{LO}\to\text{HI}}$ will be determined later as part of our analysis of the HI- and LO-criticality modes.

4.5.2 Probabilities of Deadline Misses

Let us now determine the deadline miss probabilities of $\text{DMP}_{\text{HI}}(T)$ and $\text{DMP}_{\text{LO}}(T)$, i.e., the probabilities that at least one HI criticality job or one LO criticality job misses its deadline within the time interval T. With $T = 1h$ we get the quantities as required by the schedulability test according to Definition 4.5. For the following theorem, let us suppose that $\chi \in \{\text{LO}, \text{HI}\}$ denotes the criticality of jobs in the deadline miss probabilities.

In principle, the analysis investigates two coupled systems. The first one, which is denoted as the LO-system, never does a mode switch, i.e., all mode switch events are ignored. In addition, it uses modified execution time probabilities of HI criticality jobs such that the LO-system pessimistically describes the behavior of the original

system if operating in LO-criticality mode. In particular, all execution times of HI jobs that are higher than the threshold are trimmed to it, see Definition 4.2. The worst-case steady-state probability that at least one χ job misses its deadline during a hyperperiod in the LO-system is denoted as $\mathsf{DMP}_\chi^{\text{LO}}$. This probability is determined using the worst-case steady-state backlog and response-time analysis as provided in Lemma 4.4, but using the trimmed execution times of HI jobs. The other system is denoted as the HI-system and considers the case that at least one LO→HI mode switch happened within a hyperperiod, i.e., at least one HI-criticality mode is executed.

Definition 4.11 *Deadline Miss Probabilities in Different Modes*: The worst case probability that at least one χ critical job misses its deadline during any HI-criticality mode started in a single hyperperiod is denoted as $\mathsf{DMP}_\chi^{\text{HI}}$. The worst-case steady-state probability that at least one χ critical job misses its deadline during a hyperperiod in a system where (a) all mode switch events are ignored and (b) execution times of HI jobs are trimmed to their threshold according to Definition 4.2 is denoted as $\mathsf{DMP}_\chi^{\text{LO}}$.

To reiterate, $\mathsf{DMP}_\chi^{\text{LO}}$ can be computed according to Lemma 4.4. Using these definitions, we can determine bounds on the requested deadline miss probabilities using the following result. The desired probabilities per hour can be obtained by setting $T = 1h$.

Theorem 4.5 Deadline Miss Probabilities: *The deadline miss probabilities* $\mathsf{DMP}_\chi(T)$ *for* $\chi \in \{\text{LO}, \text{HI}\}$ *can be bounded as follows:*

$$\mathsf{DMP}_\chi(T) \leq 2 - (1 - \mathsf{DMP}_\chi^{\text{LO}})^{\lceil T/_{\text{HP}} \rceil} - (1 - \mathsf{DMP}_\chi^{\text{HI}})^{\lceil T/_{\text{HP}} \rceil} \quad (4.15)$$

Proof: It needs to be proven that the probability that there is no deadline miss of any χ job within time interval T is bounded by

$$1 - \mathsf{DMP}_\chi(T) \geq (1 - \mathsf{DMP}_\chi^{\text{LO}})^{\lceil T/_{\text{HP}} \rceil} + (1 - \mathsf{DMP}_\chi^{\text{HI}})^{\lceil T/_{\text{HP}} \rceil} - 1$$

There is no deadline miss within T if there is no deadline miss when the system executes in LO-criticality mode and there is no deadline miss if it operates in HI-criticality mode. Suppose the first event is named a and the second one b, then we know that

$\Pr(a \cap b) = \Pr(a) + \Pr(b) - \Pr(a \cup b) \geq \Pr(a) + \Pr(b) - 1$ even if both events are not independent. Therefore, the theorem is true if

$$(1 - \mathsf{DMP}_{\chi}^{\text{LO}})^{\lceil T/\text{HP} \rceil}$$

lower bounds the probability that there is no deadline miss when the system is in LO-criticality mode and

$$(1 - \mathsf{DMP}_{\chi}^{\text{HI}})^{\lceil T/\text{HP} \rceil}$$

lower bounds the probability that there is no deadline miss when the system is in HI-criticality mode.

Let us first look at the LO-criticality mode. At first, note that $\lceil T/\text{HP} \rceil$ is the number of hyperperiods that completely cover an interval of length T. Therefore, we can safely assume that our interval has the length of $\lceil T/\text{HP} \rceil$ full hyperperiods. Remember that the backlogs during a steady-state computation are monotonically increasing, see Theorem 4.2. In a similar way, response times of jobs are monotonically increasing from hyperperiod to hyperperiod, see Lemma 4.4. As a result, the deadline miss probabilities of jobs are increasing from hyperperiod to hyperperiod as well and $\mathsf{DMP}_{\chi}^{\text{LO}}$ is a safe upper bound for *every* hyperperiod in our modified LO-system. We model the system as a worst-case Bernoulli process, acting from hyperperiod to hyperperiod. As a result, $\left(1 - \mathsf{DMP}_{\chi}^{\text{LO}}\right)^{\lceil T/\text{HP} \rceil}$ is a lower bound on the probability that there is no deadline miss in the LO-system, i.e., all switching events are disabled and the execution times of HI jobs are trimmed.

It remains to be shown that the response times in our LO-system are always larger or equal than those in the original system when it is in LO-criticality mode. This is certainly true as after a HI→LO mode switch, the backlogs are 0 for sure and therefore they are lower than those in the modified LO-system. Due to Lemma 4.4, the response times are larger in the modified LO-system. Moreover, trimming of execution times of HI criticality jobs has no influence on the backlogs as long as there is no HI→LO mode switch, i.e., the original system operates in LO-mode.

Now let us look at the HI-mode. Again note, that $\lceil T/\text{HP} \rceil$ is the number of hyperperiods that completely cover an interval of length T. The worst-case probability that at least one χ critical job

misses its deadline during any HI-criticality mode started in a single hyperperiod is denoted as $\mathrm{DMP}_{\chi}^{\mathrm{HI}}$, see Definition 4.11. Therefore, $\left(1 - \mathrm{DMP}_{\chi}^{\mathrm{HI}}\right)^{\lceil T/\mathrm{HP} \rceil}$ is a lower bound on the probability that there is no deadline miss caused by a LO→HI switch within a hyperperiod.

This concludes the proof as we considered the case that the systems operates in LO-criticality mode somewhere within a hyperperiod (bounded by the case that it is always in this mode during the hyperperiod) and the case that one or more HI-criticality modes are started within a hyperperiod (all corresponding deadline misses are accounted for in the hyperperiod where the HI-criticality mode was started). □

Next we determine the quantities $\Delta_{\max}^{\mathrm{HI}}$, $\mathrm{P}_{\mathrm{LO}\to\mathrm{HI}}^{\mathrm{HP}}$, $\mathrm{DMP}_{\chi}^{\mathrm{LO}}$ and $\mathrm{DMP}_{\chi}^{\mathrm{HI}}$ required to compute $\mathrm{PDJ}_{\deg}$, $\mathrm{DMP}_{\mathrm{HI}}(T)$ and $\mathrm{DMP}_{\mathrm{LO}}(T)$. We start by analyzing the behavior of the MC system in LO-criticality mode.

4.5.3 LO-criticality Mode

The analysis of the LO-criticality mode allows us to determine some of the required quantities, namely the worst case probability $\mathrm{P}_{\mathrm{LO}\to\mathrm{HI}}^{\mathrm{HP}}$ of at least one LO→HI mode switch within a hyperperiod and the worst-case probability $\mathrm{DMP}_{\chi}^{\mathrm{LO}}$ that at least one χ critical job misses its deadline within a hyperperiod if operating in the modified LO-system, see Section 4.5.2. Moreover, we determine the worst-case probability of a LO→HI mode switch at any time instance $t \in \{0, \ldots, \mathrm{HP} - 1\}$ within any hyperperiod, as this quantity allows us to analyse the HI-criticality mode later on.

Lemma 4.5: *Given a modified task system where no* LO→HI *mode switch is executed and all* HI *critical jobs are trimmed to their execution time threshold C_i^{thr}, see Definition 4.2. Then,*

$$\mathrm{DMP}_{\chi}^{\mathrm{LO}} = \left[\sum_{\tau_{i,j} \in S} \overline{\mathrm{DMP}}_{i,j} \right]^{1} \qquad (4.16)$$
$$S = \{\tau_{i,j} \mid \chi_i = \chi \ \wedge \ 0 \leq j < \mathrm{HP}/T_i\}$$

is an upper bound on the probability of at least one deadline miss of any χ job during LO*-criticality mode execution within any hyperperiod, where $\overline{\mathrm{DMP}}_{i,j}$ denotes an upper bound on the deadline miss probability of job $\tau_{i,j}$ according to Theorem 4.5. Note, $[...]^1$ indicates the expression is limited to a maximum value of* 1.

Proof: We will show that the response times in the modified system are always larger or equal than those in the original system when it is in LO-criticality mode. According to Theorem 4.3, the upper bound on the deadline miss probability $\overline{\mathrm{DMP}}_{i,j}$ holds for any hyperperiod. On the other hand, we can not assume that the deadline miss probabilities for jobs that are within one hyperperiod are independent. Therefore, we upper bound the probability of the union of events by their sum. It remains to be shown that the modified LO-system with all LO→HI mode switches disabled and the trimmed execution times of HI jobs provides upper bounds on the original system when operating in LO-criticality mode. This is certainly true as after a HI→LO mode switch in the original system, the backlogs are 0 for sure and therefore, they are lower than those in the modified LO-system. Due to Lemma 4.4, the response times are larger in the modified LO-system. Moreover, trimming of execution times of HI jobs has no influence on the backlogs as long as there is no HI→LO mode switch, i.e., the original system operates in LO-mode. The bounding of the value $\mathrm{DMP}_{\chi}^{\mathrm{LO}}$ to 1 is safe, as for any summation of events we have $\Pr(a \cup b) \leq \Pr(a) + \Pr(b)$ and $\Pr(a \cup b) \leq 1$ leading to $\Pr(a \cup b) \leq \min(1, \Pr(a) + \Pr(b))$. □

Now, we determine an upper bound on the worst-case probability $\mathrm{P}_{\mathrm{LO}\to\mathrm{HI}}(t)$ of a LO→HI mode switch at time instance $t \in \{0, ..., \mathrm{HP} - 1\}$ within any hyperperiod. Remember that there are three triggering events for a LO→HI mode switch, namely (a) a HI critical job misses its deadline (b) the system-level backlog, meaning the amount of pending executions, becomes higher than a predefined threshold $B_{\max}$ and (c) a HI critical job overruns its threshold execution time C_{thr}. We analyze the three different mechanisms one after the other and finally combine the results.

Let us start with the deadline miss probability at time instance $0 \leq t < \mathrm{HP}$, which we denote as $\mathrm{P}_{dm}(t)$.

Lemma 4.6: *Given a modified task system where no* LO→HI *mode switch is executed and all* HI *critical jobs are trimmed to their execution time threshold* C_i^{thr}*, see Definition 4.2. Then,*

$$\forall 0 \leq t < \text{HP} \; : \quad \text{P}_{dm}(t) = \left[\sum_{\tau_{i,j} \in S(t)} \overline{\text{DMP}}_{i,j} \right]^1 \tag{4.17}$$
$$S(t) = \{\tau_{i,j} \mid \chi_i = \text{HI} \; \wedge \; d_{i,j} = t\}$$

is an upper bound on the probability of at least one deadline miss of any HI *critical job during* LO*-criticality mode execution at time* t*,* $0 \leq t < \text{HP}$*, where* $\overline{\text{DMP}}_{i,j}$ *denotes an upper bound on the deadline miss probability of job* $\tau_{i,j}$ *in the modified task system according to Theorem 4.3. Note,* $[...]^1$ *indicates the expression is limited to a maximum value of* 1*.*

Proof: We can not assume that the deadline miss probabilities at time t are independent. Therefore we use as an upper bound of the union of events the sum of the individual probabilities. The bounding of the value $\text{P}_{dm}(t)$ to 1 is safe, as for any summation of events we have $\Pr(a \cup b) \leq \Pr(a) + \Pr(b)$ and $\Pr(a \cup b) \leq 1$ leading to $\Pr(a \cup b) \leq \min(1, \Pr(a) + \Pr(b))$. $S(t)$ denotes the set of all HI critical jobs with deadline at time t. □

We continue with the probability that at time instance $0 \leq t < \text{HP}$ the total backlog exceeds the upper bound $B_{\max}$ which we denote as $\text{P}_{be}(t)$.

Lemma 4.7: *Given a modified task system where no* LO→HI *mode switch is executed and all* HI *critical jobs are trimmed to their execution time threshold* C_i^{thr}*, see Definition 4.2. Then,*

$$\forall 0 \leq t < \text{HP} \; : \quad \text{P}_{be}(t) = \Pr(\overline{\mathcal{B}}_N(t) > B_{\max}) \tag{4.18}$$

is an upper bound on the probability that the total backlog at time t *exceeds* $B_{\max}$ *during* LO*-criticality mode execution within any hyper-period, where* $\overline{\mathcal{B}}_N(t)$ *denotes an upper bound on the lowest priority backlog in the modified task system according to Theorem 4.2.*

Proof: The total backlog equals $\overline{\mathcal{B}}_N(t)$ according to Definition 4.7. Then, the lemma directly follows from Theorem 4.2. □

Unfortunately, the computation of the probability $\mathrm{P}_{ov}(t)$ that at time instance $0 \leq t < \text{HP}$ at least one HI critical job overruns its threshold execution time C_i^{thr} is more involved. Whereas the overrun probability $\Pr(\mathcal{C}_i > C_i^{\text{thr}})$ can be simply calculated, it is more complex to understand at what time instance such an event happens, due to interference from other jobs. We first compute the upper bound on the backlog for our modified LO-system as usual. Based on this, we now consider each HI critical job individually and compute its response time if the job would have the execution time C_i^{thr}. If this response time plus the release time $r_{i,j}$ of the job equals t, then the job overruns at t under the condition that it overruns at all. The following lemma summarizes the corresponding result.

Lemma 4.8: *Given a modified task system where no* LO→HI *mode switch is executed and all* HI *critical jobs are trimmed to their execution time threshold* C_i^{thr}*, see Definition 4.2. Then,* $\forall 0 \leq t < \text{HP}$

$$\mathrm{P}_{ov}(t) = \left[\sum_{\tau_{i,j} \in S} \Pr(\mathcal{C}_i > C_i^{\text{thr}}) \cdot \Pr\left(\left(\texttt{rta}\left(\overline{\mathcal{B}}_i(r_{i,j}),\, \Pi,\, \tau_{i,j}^{\text{ov}} \right) + r_{i,j} \right) \mod \text{HP} = t \right) \right]^1$$

$$S = \{ \tau_{i,j} \mid \chi_i = \text{HI} \} \tag{4.19}$$

is an upper bound on the probability that at time instance $0 \leq t <$ HP *at least one* HI *critical job overruns its threshold execution time* C_i^{thr}*. Here,* $\overline{\mathcal{B}}_i(t)$ *denotes an upper bound on the level* i *backlog in the modified task system according to Theorem 4.2 and* $\tau_{i,j}^{ov}$ *denotes a modified job* $\tau_{i,j}$ *with a deterministic computation time of* C_i^{thr}*. Note,* $[...]^1$ *indicates the expression is limited to a maximum value of* 1*.*

Proof: At first note that we do not assume that the probabilities of overrunning the threshold execution time C_i^{thr} are independent. Therefore, the union of at least one overrun at time t is bounded by the sum of individual probabilities for each HI job, see the definition of S. Moreover, $\Pr(a) = \Pr(a|b) \cdot \Pr(b)$ for events a and b. In our

case, $\Pr(b) = \Pr(\mathcal{C}_i > C_i^{\text{thr}})$, i.e., the event that task $\tau_{i,j}$ has a overrun of its threshold execution time. We now need to show that the term $\Pr(\texttt{rta}\left(\overline{\mathcal{B}}_i(r_{i,j}), \Pi, \tau_{i,j}^{\text{ov}}\right) + r_{i,j}) \mod \text{HP} = t)$ denotes the probability that an overrun due to task $\tau_{i,j}$ happens at time t under condition that the overrun happens at all, i.e., it represents $\Pr(a|b)$. Note that the term $(\texttt{rta}\left(\overline{\mathcal{B}}_i(r_{i,j}), \Pi, \tau_{i,j}^{\text{ov}}\right) + r_{i,j})$ denotes the finishing time of task $\tau_{i,j}$ if using the worst-case steady-state backlogs $\overline{\mathcal{B}}$ and the execution time C_i^{thr}. Therefore, under the assumption that the task overruns, it determines the distribution of the time when the overrun actually happens. As this time may be in the next hyperperiod, we use the modulo operation. The bounding of the value P_{ov} to 1 is safe, as for any summation of events we have $\Pr(a \cup b) \leq \Pr(a) + \Pr(b)$ and $\Pr(a \cup b) \leq 1$ leading to $\Pr(a \cup b) \leq \min(1, \Pr(a) + \Pr(b))$. □

Based on the previous three lemmas, we conclude this section with the desired worst-case probability $\text{P}_{\text{LO}\to\text{HI}}(t)$ of a LO→HI mode switch at time instance $0 \leq t < \text{HP}$ within any hyperperiod.

Theorem 4.6: *$P_{\text{LO}\to\text{HI}}(t)$ is an upper bound on the worst-case probability of a* LO→HI *mode switch at time instance $0 \leq t <$ HP within any hyperperiod with*

$$\forall 0 \leq t < \text{HP} \; : \quad \text{P}_{\text{LO}\to\text{HI}}(t) = \left[\text{P}_{dm}(t) + \text{P}_{be}(t) + \text{P}_{ov}(t)\right]^1 \quad (4.20)$$

where $P_{dm}(t)$, $P_{be}(t)$ and $P_{ov}(t)$ are computed according to Lemmas 4.6 to 4.8, respectively. An upper bound on the probability of at least one LO→HI *mode switch within a hyperperiod can be determined as*

$$\text{P}_{\text{LO}\to\text{HI}}^{\text{HP}} = \left[\sum_{0 \leq t < \text{HP}} \text{P}_{\text{LO}\to\text{HI}}(t)\right]^1 \quad (4.21)$$

Note, $[\ldots]^1$ indicates the expression is limited to a maximum value of 1.

Proof: The theorem is a simple consequence of the previous lemmas, as we can not assume independence of events within a hyperperiod. □

As a simple corollary to the above theorem, one can compute a lower bound on the expected length of a single LO-criticality mode execution as

$$\Delta_{\text{exp}}^{\text{LO}} = (\lceil \frac{1}{\text{P}_{\text{LO}\to\text{HI}}^{\text{HP}}} \rceil - 1) \cdot \text{HP} \tag{4.22}$$

This results concludes the analysis of the LO-criticality mode, and we can move on to analyze the HI-criticality mode in order to determine the remaining quantities necessary for Theorems 4.4 and 4.5.

4.5.4 HI-criticality Mode

We are still missing the computation of the maximal duration of a HI-criticality mode execution quantity $\Delta_{\text{max}}^{\text{HI}}$, as well as the worst-case probability $\text{DMP}_{\chi}^{\text{HI}}$ of at least one deadline miss of any χ job during any HI-criticality mode started within a hyperperiod, where $\chi \in \{\text{LO}, \text{HI}\}$.

To this end, we will determine HP different worst-case HI-criticality mode scenarios, one for each starting time $0 \leq t < \text{HP}$ relative to the beginning of a hyperperiod. In other words, we will investigate HP different HI-criticality mode executions and then use the maximum of their durations as $\Delta_{\text{max}}^{\text{HI}}$, and the maximum of their deadline miss probabilities to determine upper bounds that at least one HI or LO task misses its deadline during a single HI-criticality mode execution. These quantities will then be combined with the probability $\text{P}_{\text{LO}\to\text{HI}}(t)$ that a LO→HI switch happens at relative starting time t in order to determine $\text{DMP}_{\chi}^{\text{HI}}$, i.e., the worst-case probability of at least one deadline miss of any χ critical job during any HI-criticality mode started within a hyperperiod.

Broadly speaking, HI-criticality mode has three differences with LO-criticality mode. First, jobs released in HI-mode have different execution times: LO jobs are released in degraded mode, and HI jobs do not have the condition that they do not overrun their C_i^{thr} execution time threshold. Second, 'carry-over' jobs, which are released in LO-criticality mode but whose deadlines are after the mode switch, are present in HI-criticality mode and they need to be accounted for. Third, the initial system-level backlog is not zero, but depends on the mode switch time trigger. To account for

these differences, we present the following worst-case HI-critical execution task-set. It is created such that it is pessimistic what ever the mode switch trigger may be, and it accounts for both carry-over jobs and jobs released during HI-mode.

The worst-case HI-mode scenario for starting at time t is defined as follows:

Definition 4.12 *Worst-Case HI-Criticality Execution*: We define HP task sets $\widehat{\Pi}(t)$, one for each starting time $0 \leq t < \text{HP}$. Each differs from the original task set Π as follows:

1. The phase offsets ϕ_i are implicitly changed such that all jobs are already available in $0 \leq t < \text{HP}$, i.e., we allow for negative job indices j.
2. We consider all jobs with starting times after t, i.e., $j \geq (t - \phi_i)/T_i + 1$. They have a known execution time $\widehat{\mathcal{C}}_i$ which is not larger than the degraded mode WCET C_i^{deg} for LO criticality jobs, and a known execution time $\widehat{\mathcal{C}}_i = \mathcal{C}_i$ for HI criticality jobs.
3. We consider jobs whose release time is smaller than t and deadline is larger than t. These included jobs $\tau_{i,j} \in \widehat{T}$ with $(t - \phi_i)/T_i + 1 < j < (t + D_i - \phi_i)/T_i + 1$ have execution times $\widehat{\mathcal{C}}_i = \mathcal{C}_i$ for both LO and HI criticality jobs; i.e., for LO jobs the execution times are not degraded, and for HI jobs they may or may not overrun their C_i^{thr} threshold.
4. For each HI-criticality mode starting time t, $0 \leq t < \text{HP}$, we introduce the initial backlog at time t and priority levels $1 \leq i \leq N$, $\widehat{\mathcal{B}}_i(t)$. If a overrun can not happen at time t, due to the fact there is no HI job released whose deadline has passed by time t, the initial backlog is as follows:

$$\Pr(\widehat{\mathcal{B}}_i(t) = u) = \begin{cases} \Pr(\overline{\mathcal{B}}_i(t) = u) & u < B_{\max} \\ \sum_{v=B_{\max}}^{\infty} \Pr(\overline{\mathcal{B}}_i(t) = v) & u = B_{\max} \\ 0 & u > B_{\max} \end{cases} \tag{4.23}$$

where $\overline{\mathcal{B}}_i(t)$ denotes an upper bound on the i^{th} priority backlog in the modified LO-criticality system according to Theorem 4.2. If an overrun can happen at time t, due to at least one

HI job having its release time before t and its deadline after, then the initial backlog at time t is the following:

$$\Pr(\widehat{\mathcal{B}}_i(t) = u) = \begin{cases} \Pr(\overline{\mathcal{B}}_i^{\mathsf{ov}}(t) = u) & u < B_{\max} \\ \sum_{v=B_{\max}}^{\infty} \Pr(\overline{\mathcal{B}}_i^{\mathsf{ov}}(t) = v) & u = B_{\max} \\ 0 & u > B_{\max} \end{cases} \tag{4.24}$$

where $\overline{\mathcal{B}}_i^{\mathsf{ov}}(t)$ denotes an upper bound on the ith priority backlog in the modified LO-criticality system according to Theorem 4.2, but with the added condition that at least one of the released HI jobs whose deadline is after time t has overrun its threshold execution time C_i^{thr}.
Let us now describe how $\overline{\mathcal{B}}_i^{\mathsf{ov}}(t)$ can be computed. To this end, we solve

$$\Pr(\overline{\mathcal{B}}_i^{\mathsf{no+ov}}(t) = u) = \\ \Pr(\mathsf{no}) \cdot \Pr(\overline{\mathcal{B}}_i(t) = u) + \Pr(\mathsf{ov}) \cdot \Pr(\overline{\mathcal{B}}_i^{\mathsf{ov}}(t) = u)$$

Here, $\overline{\mathcal{B}}_i(t)$ denotes an upper bound on the ith priority backlog in the modified LO-criticality system according to Theorem 4.2. $\overline{\mathcal{B}}_i^{\mathsf{no+ov}}(t)$ is also an upper bound on the ith priority backlog according to Theorem 4.2, but the system used for its computation is slightly modified. It is the LO-criticality system with the difference that HI jobs released before time t whose deadlines are after that time have no condition on whether they overrun their C_i^{thr} execution time or not – we use their normal execution times $\mathcal{C}_i$ in calculating the backlog. The probability that none of these HI jobs overrun their respective C_i^{thr} execution times is noted $\Pr(\mathsf{no})$, while the $\Pr(\mathsf{ov}) = 1 - \Pr(\mathsf{no})$ is the probability that at least one of these HI jobs overruns. $\Pr(\mathsf{no})$ is obtained directly from execution times of these HI jobs, $\Pr(\mathsf{no}) = \sum_{\tau_{i,j} \in S} \Pr\left(\mathcal{C}_i > C_i^{\mathrm{thr}}\right)$, where $S = \{\tau_{i,j} \mid \chi_i = \text{HI} \ \wedge \ r_{i,j} \leq t \ \wedge \ d_{i,j} > t\}$.

Condition 2 includes all tasks which are released during HI-criticality mode, noting that LO jobs are degraded and HI jobs have $\mathcal{C}_i$ execution times. The third condition deals with carry-over jobs

from LO- to HI-criticality mode, whose deadline misses have not yet been accounted for in the LO-criticality mode analysis. Note that here the worst case comes from the assumption that all HI jobs may overrun. Finally, condition 4 includes the worst-case backlog at the starting time t, as it is the backlog with the condition that an overrun of at least one HI job occurred, but also it is limited by the maximal backlog $B_{\max}$. Simpler constructions of the worst-case task-set lead to high overestimations to the length and deadline miss probabilities of HI-criticality mode.

Starting from the worst-case scenarios for the HI-mode for each time instant t, $0 \leq t < \text{HP}$, we now evaluate each scenario and determine the corresponding worst-case durations as well as the deadline miss probabilities. To do this, we apply the results from Section 4.4 and use the function $\texttt{bsse}\left(\widehat{\mathcal{B}}_i(t), \widehat{\Pi}, i, t, u\right)$ to compute all relevant backlogs for the task sets from Definition 4.12. The successive computation of the backlogs stops whenever the system gets idle for the first time: $\widehat{\mathcal{B}}_i(t_s) = 0$ for all priority levels i. This time is an upper bound on the HI→LO switching time. Using the response time analysis, see (4.11), we can finally determine all jobs that miss their deadline during the HI-mode. Additionally, for the response time analysis for calculating the deadline miss probabilities of HI carry-over jobs, we substitute the execution time of the carry-over job under analysis $\widehat{\mathcal{C}}_i$ with the conditional execution time $\Pr(\mathcal{C}_i > C_i^{\text{thr}})$, in order to get the deadline miss probability with the condition that the HI carry-over job overran its C_i^{thr} execution time threshold.

Lemma 4.9: *The first time t_{idle}, the execution of the task set $\widehat{\Pi}(t)$ from Definition 4.12 yields a system-level backlog which is zero, determines an upper bound $\Delta_{\max}^{\text{HI}}(t)$ on the duration of a HI-criticality mode starting at time t relative to the beginning of any hyperperiod of the original task system Π:*

$$\forall 0 \leq t < \text{HP} \; : \quad \Delta_{\max}^{\text{HI}}(t) = t_{\text{idle}} - t \tag{4.25}$$

Let us define the probability $p_{i,j}(t)$ that some job $\tau_{i,j}$ of task set $\widehat{\Pi}(t)$ from Definition 4.12 misses its deadline in the time interval $[t, t + \Delta_{\max}^{\text{HI}}(t)]$. Then $\text{DMP}_{\chi}^{\text{HI}}(t)$ is an upper bound on the probability

that there is at least one deadline miss of any χ critical job with $\chi \in \{\text{LO}, \text{HI}\}$ within a HI*-criticality mode execution starting at time t relative to the beginning of any hyperperiod in the original task set Π:*

$$\forall 0 \leq t < HP : \quad \mathrm{DMP}_{\chi}^{\text{HI}}(t) = \left[\sum_{\tau_{i,j} \in S(t)} p_{i,j}(t) \right]^{1} \tag{4.26}$$
$$S(t) = \{\tau_{i,j} \in \widehat{\Pi}_i(t) \mid \chi_i = \chi\}$$

Note, $[...]^1$ indicates the expression is limited to a maximum value of 1.

Proof: The main part of the proof is to show that the task set $\widehat{\Pi}(t)$ indeed defines a worst-case scenario in terms of duration and deadline miss probabilities, when the HI-criticality mode starts at time t relative to the beginning of any hyperperiod. Note that the second condition in Definition 4.12 ensures that all tasks which are released during a HI-criticality mode in the worst case, are included in the HI-criticality task set as well. Moreover, we consider the exact execution times for all of these jobs, namely the degraded execution times $\widehat{\mathcal{C}}_i$ which are not longer than C_i^{deg} for LO criticality jobs, and $\widehat{\mathcal{C}}_i = \mathcal{C}_i$ for HI criticality jobs. The third condition adds the worst-case carry-over jobs from LO- to HI-criticality mode whose deadline misses have not yet been accounted for in the LO-mode analysis. All jobs who missed their deadline before the LO→HI mode switch have been considered already in the LO-mode analysis, but their possible backlog at t will be considered. Therefore, we just need to explicitly include jobs whose release time is before and whose deadline is after the LO→HI mode switch. The corresponding execution times are taken as worst-case as well, namely for each carry-over HI job individually, for calculating its deadline miss probability we assume it overruns its execution time threshold. Finally, we look at the worst-case backlog at the starting time t. It encompasses the remaining execution times of jobs who were released before t but not yet finished. Due to the triggering condition of a mode switch, we assume the worst-case that at least one HI job has overrun its C_i^{thr} execution time. Also according to triggering conditions, the backlog is never larger then $B_{\max}$ for all priority levels. Note that the backlog also contains jobs whose deadline is within the HI-mode, i.e., the carry-over jobs who have been explicitly included

as tasks. In order to determine the upper bound on the deadline miss probability $\mathrm{DMP}^{\text{HI}}_{\chi}(t)$ of any χ-critical job we again do not assume independence of individual miss events and use the sum of the corresponding probabilities as an upper bound. □

As a result of this lemma we can determine the desired quantities, namely maximal duration and upper bound on deadline misses, for each time point t relative to the starting of a hyperperiod. The computations are based on simple simulations of HP executions of worst-case HI-criticality mode scenarios. The simulation times are finite as long as there exists a finite time in $\widehat{\Pi}(t)$ when the system gets the first time idle. The following lemma leads to a necessary and sufficient condition.

Lemma 4.10: *A set of finite bounds $\Delta^{\text{HI}}_{\max}(t)$ on the duration of HI-criticality modes exists if and only if the maximal system utilization in HI-criticality mode in the original system is less than one.*

Proof: Let us look at the modified task set $\widehat{\Pi}(t)$ starting at time t. If the maximal system utilization in HI-criticality mode is less than one, then the maximal system level backlog at time $t + (n+1) \cdot \text{HP}$ is strictly smaller than the maximal system level backlog at time $t + n \cdot \text{HP}$ for $n > 1$, because the arriving jobs in time interval $[t + n \cdot \text{HP}, t + (n+1) \cdot \text{HP})$ are identical for all $n > 1$ and there is less additional accumulated computation time from all arriving jobs than its length HP. Therefore, a time instance will exist when the maximal system level backlog is zero and the system is idle. If the maximal system utilization in HI-criticality mode is larger or equal than one, then the maximal system level backlog at time $t+(n+1) \cdot \text{HP}$ could be equal or greater than the maximal system level backlog at time $t + n \cdot \text{HP}$. Therefore, in the worst case, the system level backlog never gets to zero and the HI-criticality mode could last for ever. □

Based on these results, we can now aggregate the computed quantities in order to determine the maximal duration of a HI-criticality mode execution quantities $\Delta^{\text{HI}}_{\max}$ as well as the worst-case probability $\mathrm{DMP}^{\text{HI}}_{\chi}$ of at least one deadline miss of any χ job during any HI-criticality mode started within a hyperperiod, where $\chi \in \{\text{LO}, \text{HI}\}$.

Theorem 4.7: *$\Delta^{\text{HI}}_{\max}$ is an upper bound on the maximal duration of any* HI*-criticality mode in the original task system Π, where*

$$\Delta^{\text{HI}}_{\max} = \max_{0 \leq t < \text{HP}} \Delta^{\text{HI}}_{\max}(t) \tag{4.27}$$

$\text{DMP}^{\text{HI}}_{\chi}$ is a bound on the worst-case probability of at least one deadline miss of any χ critical job with $\chi \in \{\text{LO}, \text{HI}\}$ during any HI*-criticality mode started within a hyperperiod in the original task system, where*

$$\text{DMP}^{\text{HI}}_{\chi} = \left[\sum_{0 \leq t < \text{HP}} \text{P}_{\text{LO}\to\text{HI}}(t) \cdot \text{DMP}^{\text{HI}}_{\chi}(t) \right]^{1} \tag{4.28}$$

with $P_{\text{LO}\to\text{HI}}(t)$ as determined in Theorem 4.6. Note, $[\ldots]^1$ indicates the expression is limited to a maximum value of 1.

Proof: According to Lemma 4.9, $\Delta^{\text{HI}}_{\max}(t)$ is an upper bound on the duration of a HI-criticality mode starting at relative time t within a hyperperiod. Clearly, the maximum for all relative time instances provides the maximal duration for any time instance. The probability of a deadline miss within a HI-mode execution is the probability of the union of deadline misses at any time instance within the hyperperiod. As we cannot assume independence, we upper bound this probability by the sum of individual probabilities. The probability of a deadline miss within a HI-mode starting at relative time t is clearly the probability that a a mode switch happens, i.e., $\text{P}_{\text{LO}\to\text{HI}}(t)$, times the probability that a deadline miss happens within the HI-mode, i.e., $\text{DMP}^{\text{HI}}_{\chi}(t)$. □

This concludes the schedulability analysis of probabilistic MC systems according to Definition 4.5, as all the required quantities for Theorems 4.4 and 4.5 have been determined in Sections 4.5.3 and 4.5.4.

Of course, the tightness of the pMC analysis can be improved through various approaches. Some of them, as well as limitations of the described analysis, are noted at the end of this chapter.

4.6 Complexity of the Analysis

Here we comment on the computational complexity of our proposed pMC schedulability analysis. Algorithm 4.2 presents a high-level recapitulation of the analysis, where all pseudo-commands are as explained in Section 4.5. The computational complexity of the analysis is

$$\mathcal{O}(n^2 \cdot \text{HP} \cdot c \log c)$$

where n is the number of jobs in one hyperperiod, HP is the length of one hyperperiod, and c is the length or number of values in the execution time distributions.

In the analysis, the most complex atomic command is the convolution. When using the fast Fourier transform algorithm, one convolution has a cost of $\mathcal{O}(c \log c)$.

Let us now comment on the complexity of the analysis in detail. According to Section 4.4.1, the steady state backlog is approximated by $\mathcal{B}_i(k \cdot \text{HP})$, where k is the smallest natural number satisfying inequality (4.9). To calculate $\mathcal{B}_i(k \cdot \text{HP})$, a convolution is needed for every one of the $n \cdot k$ jobs, thus the cost of line 2 is $\mathcal{O}(n \cdot k \cdot c \log c)$. Similarly, according to point 4 of Definition 4.12, backlog $\widehat{\mathcal{B}}_i(t)$ is defined as a combination of two steady state backlogs, so the cost of line 14 is also $\mathcal{O}(n \cdot k \cdot c \log c)$. The number k depends on the required numerical precision (4.9), but we have found it to be in the same order of magnitude as n, $k \sim n$.

To compute deadline miss probabilities, i.e., lines 4 and 17, the response time analysis is used as defined by Algorithm 4.1. Line 6 is based on response time analysis as well (Lemma 4.8). To find the response time of a job, we need to do as many convolutions as there are jobs preempting the said job. Thus, the cost of these lines is $\mathcal{O}(n \cdot c \log c)$.

Finally, when analyzing HI-mode, the maximal duration of the mode $\Delta^{\text{HI}}_{\max}$ plays a role. When calculating $\Delta^{\text{HI}}_{\max}$ in line 15, and when computing deadlines miss probabilities of jobs in lines 16 and 17, we need to take into account all jobs that are released in HI-mode. Regardless on when HI-mode is entered or exited, the number of these jobs is at most $n \cdot \Delta^{\text{HI}}_{\max}/\text{HP}$. For schedulable systems, we found that $\Delta^{\text{HI}}_{\max}$ is in the same order of magnitude as HP, $\Delta^{\text{HI}}_{\max} \sim \text{HP}$ and $\Delta^{\text{HI}}_{\max}/\text{HP} \sim 1$.

Table 4.2: Probabilistic mixed-criticality (pMC) analysis run times, for a different number of jobs per hyperperiod n

# Tasks	# Jobs in HP (n)	Utilization	Average run time
13	50–125	0.5	5 s
25	150–200	1.0	19 s
60	350–475	1.2	1 min 59 s
75	450–650	1.5	4 min 21 s
100	650–825	2.0	23 min 42 s

Even though the computational complexity of this scheme is high, we find it to be acceptable. The analysis only needs to be done offline, while designing the system. Furthermore, parts of Algorithm 4.2 can be made to run in parallel. Each iteration of the for-loop in line 13 can be run independently, meaning that the analysis of HI-mode can be done in parallel on HP processes, each of complexity $\mathcal{O}(n^2 \cdot c \log c)$. Consequently, this would be the computational complexity of the whole algorithm, if we were to have unlimited resources.

Run time. We tested the analysis for various task-sets on a single core of a Dual Deca-Core Intel Xeon E5-2690 v2 processor, running at 3.00 GHz. The task-sets are generated as described in the experimental Section 4.7.2, and all have the following parameters: HP $= 1000$ and $c \sim 1000$. In Table 4.2, we noted the average analysis run times for task-sets of different number of jobs per hyperperiod n, and utilizations.

4.7 Experimental Results

In order to illustrate our pMC schedulability analysis, this section first shows one sample task-set. The sample task-set is inspired by applications from the avionics industry. Then, experiments on randomly generated task-sets are used to compare pMC scheduling

Algorithm 4.2: A high-level overview of the probabilistic mixed-criticality (pMC) schedulability analysis, with relevant references and computational complexities noted in the comments

procedure LO-criticality mode analysis ▷ $\mathcal{O}(n^2 \cdot c \log c)$
  Compute steady state backlog $\overline{\mathcal{B}}_i(0)$
    ▷ Theorem 4.1 and (4.9), $\mathcal{O}(n \cdot k \cdot c \log c)$
  for *each job* $\tau_{i,j}$ *in* HP **do** ▷ $\mathcal{O}(n^2 \cdot c \log c)$
    Compute deadline miss prob. $\overline{\mathrm{DMP}}_{i,j}$
      ▷ Theorem 4.3 and Algorithm 4.1, $\mathcal{O}(n \cdot c \log c)$
    if $\tau_{i,j}$ *is* HI *job* **then**
      Compute time of C_i^{thr} overrun detection
        ▷ Lemma 4.8 and Algorithm 4.1, $\mathcal{O}(n \cdot c \log c)$
  Get $\mathrm{DMP}_{\chi}^{\mathrm{LO}}$ ▷ Lemma 4.5, $\mathcal{O}(n)$
  Get $\mathrm{P}_{dm}(t)$ for each t in HP ▷ Lemma 4.6, $\mathcal{O}(\mathrm{HP} \cdot n)$
  Get $\mathrm{P}_{be}(t)$ for each t in HP ▷ Lemma 4.7, $\mathcal{O}(\mathrm{HP} \cdot c)$
  Get $\mathrm{P}_{ov}(t)$ for each t in HP ▷ Lemma 4.8, $\mathcal{O}(\mathrm{HP} \cdot n)$
  Get $\mathrm{P}_{\mathrm{LO}\to\mathrm{HI}}^{\mathrm{HP}}$ ▷ Theorem 4.6, $\mathcal{O}(1)$

procedure HI-criticality mode analysis ▷ $\mathcal{O}(\mathrm{HP} \cdot n^2 \cdot c \log c)$
  for *each time* t *in* HP **do** ▷ $\mathcal{O}(\mathrm{HP} \cdot n^2 \cdot c \log c)$
    Compute initial backlog $\widehat{\mathcal{B}}_i(t)$
      ▷ Definition 4.12 and (4.23) and (4.24), $\mathcal{O}(n \cdot k \cdot c \log c)$
    Get $\Delta_{\max}^{\mathrm{HI}}(t)$ ▷ Lemma 4.9, $\mathcal{O}(n \cdot \Delta_{\max}^{\mathrm{HI}}/\mathrm{HP} \cdot c \log c)$
    for *each job* $\tau_{i,j}$ *in* $[t, t + \Delta_{\max}^{\mathrm{HI}}(t))$ **do**
      ▷ $\mathcal{O}(n^2 \cdot \Delta_{\max}^{\mathrm{HI}}/\mathrm{HP} \cdot c \log c)$
      Compute deadline miss prob.
        ▷ Lemma 4.9 and Algorithm 4.1, $\mathcal{O}(n \cdot c \log c)$
    Get $\mathrm{DMP}_{\chi}^{\mathrm{HI}}(t)$ ▷ Lemma 4.9, $\mathcal{O}(n)$
  Get $\mathrm{DMP}_{\chi}^{\mathrm{HI}}$, $\Delta_{\max}^{\mathrm{HI}}$ ▷ Theorem 4.7, $\mathcal{O}(\mathrm{HP})$

Get $\mathrm{DMP}_{\mathrm{HI}}(1\,\mathrm{h})$, $\mathrm{DMP}_{\mathrm{LO}}(1\,\mathrm{h})$ ▷ Theorem 4.5, $\mathcal{O}(1)$
Get $\mathrm{PDJ}_{\deg}$ ▷ Theorem 4.4, $\mathcal{O}(1)$

Table 4.3: Scheduling schemes used for experimental evaluation

Deterministic schemes		
DMPO	Deadline monotonic priority ordering	Audsley et al. [ABRW91]
AMC	Adaptive mixed-criticality	Baruah et al. [BBD11b]
UB-HL	Upper bound on fixed priority preemptive MC schemes	Baruah et al. [BBD11b]
Probabilistic schemes		
pDMPO	Probabilistic deadline monotonic priority ordering	Díaz et al. [DGK+02]
pMC	Probabilistic mixed-criticality	*This work*

with other schemes: a probabilistic but non-mixed-critical scheme pDMPO, a deterministic adaptive mixed-criticality scheme, and a deterministic non-MC DMPO scheme. These are all listed in Table 4.3, and described in detail below. For the experiments, we generated randomized task-sets with all but one parameter the same, in order to see the effect this one parameter has.

Three experiments are conducted. The first experiment serves to show the impact of the system utilization, the second experiment varies the probability each HI task overruns its C_i^{thr} execution time threshold $\Pr(\mathcal{C}_i > C_i^{\text{thr}})$, and finally the impact of the maximal system-level backlog is visualized in the third experiment. In general, we show that pMC dominates all other schemes, except in situations when HI-criticality mode is entered too often. In these cases, we find that there is too much degradation of LO jobs, therefore scheduling using the probabilistic but non-mixed-critical pDMPO yields better results.

Baseline schemes. To evaluate pMC scheduling, we have used three deterministic and one probabilistic baseline scheme, as listed in Table 4.3. All schemes are based on fixed priority preemptive scheduling. The first deterministic scheme is a non-mixed-criticality one, deadline monotonic priority ordering (DMPO). As the name suggests, tasks are prioritized only by their deadlines, and

scheduled according to their $C_i^{\max}$ WCETs.

The next scheme is adaptive mixed-criticality (AMC), as described by Baruah et al. [BBD11b]. The scheme features two modes of operation. The system starts in LO-criticality mode where HI tasks are scheduled according to their C_i^{thr} threshold execution times. If any HI job overruns this value, a switch to HI-criticality mode happens, where all LO tasks are released in degraded mode. The scheme does not quantify the duration of these two modes, only the schedulability of them.

As a deterministic 'baseline', we introduce the upper bound on fixed priority preemptive MC schemes (UB-HL) [BBD11b]. This bound is a necessary test for all fixed priority preemptive MC schemes, and such it provides an upper bound on performance.

Finally, the probabilistic deadline monotonic priority ordering (pDMPO) scheme represents the analysis as introduced by Díaz et al. [DGK+02]. In pDMPO, tasks are given priorities based on their deadlines, they are scheduled using their complete $\mathcal{C}_i$ execution times, and there is only one mode of operation. The scheme can be viewed as a border case of pMC, where HI-criticality mode is never entered.

Task execution times. To model task execution times $\mathcal{C}_i$, Weibull distributions were used, with a condition that they do not take values greater than the task's WCET $C_i^{\max}$. These distributions have been used in related work for modeling the distribution of long but unlikely execution times [CGSH+12].

Weibull distributions are functions of two parameters, k and λ. To generate an execution time, we first choose k uniformly from $[1.5, 3]$. Then, the parameter λ is computed the following way. For LO tasks, λ was computed such that the cumulative density function at the task's WCET $C_i^{\max}$ is $1 - 10^{-8}$. Similarly, for HI tasks, we choose λ so the cumulative density function at the task's execution time threshold C_i^{thr} is $1 - 10^{-8}$, unless stated otherwise. This is the way we set the probability a HI task overruns its threshold execution time. Finally, all values of the probability density function above $C_i^{\max}$ are set to be 0, and the rest of the distribution is normalized. This way, we have a valid execution time modeled by a Weibull distribution, with the condition it never exceeds the task's WCET

$C_i^{\max}$, and for which the probability a HI task overruns its execution time threshold is C_i^{thr}.

4.7.1 Sample System

Here we introduce a task-set modelling a sample system, to which we applied our proposed schedulability analysis. We explored the task-set, first by varying execution times of all tasks, and then by varying deadlines. This was done to illustrate probabilistic mixed-criticality scheduling. We present the three schedulability values: $\text{DMP}_{\text{HI}}(1\,\text{h})$, $\text{DMP}_{\text{LO}}(1\,\text{h})$, and PDJ_{deg}, and we also show the expected duration of LO-criticality mode $\Delta_{\text{exp}}^{\text{LO}}$, and the maximal duration of HI-criticality mode $\Delta_{\max}^{\text{HI}}$.

The system's LO and HI tasks are inspired by the ROSACE [PSG+14] and FMS [DFG+14] applications, respectively. The HI tasks are inspired by an industrial implementation of the flight management system (FMS). This application consists of one task which reads sensor data, and four tasks that compute the location of the aircraft. For LO tasks, the open source avionic benchmark ROSACE was modeled. It is made up of three tasks which simulate pilot's instructions, and eight tasks implementing a controller.

Setup. Table 4.4 lists the tasks' periods and execution time values: WCETs $C_i^{\max}$, thresholds for HI tasks C_i^{thr}, and degraded WCETs C_i^{deg} for LO tasks. Execution time values are functions of the parameter f_c, which we vary from 0.05 to 7.5 in 0.05 steps. Note that for HI tasks, $C_i^{\max}$ values are 2.5 times larger than the corresponding C_i^{thr}, while for LO tasks the worst-case execution time in degraded mode is $C_i^{\text{deg}} = 0.33 \cdot C_i^{\max}$, rounded up to the nearest integer. The deadline of each task has been constrained by a factor of f_d, $D_i = T_i \cdot f_d$, where f_d is varied from 0.005 to 1 in steps of 0.005. Next, initial phases for tasks are 0, while tasks' priority assignments are given in the table. Note that we use deadline monotonic priority assignment.

We model probabilistic execution times of tasks with Weibull distributions, as described in the beginning of this section. The probability that a HI task executes for longer than its threshold

Table 4.4: The sample system's task-set

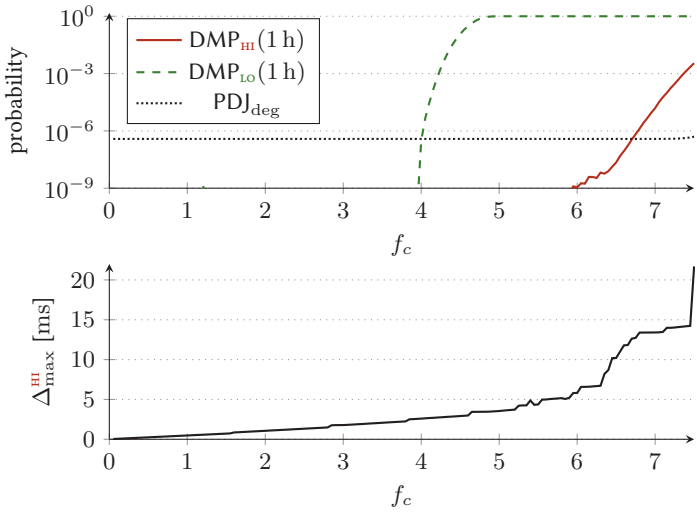
HI task τ_i	Pri.	T_i [ms]	$C_i^{\max}/f_c$ [µs]	C_i^{thr}/f_c [µs]
sens_c1	5	7.5	648	259
loc_c1	4	7.5	365	146
loc_c2	3	60	13	5
loc_c3	2	60	73	24
loc_c4	1	60	13	5
LO task τ_i	**Pri.**	**T_i [ms]**	**$C_i^{\max}/f_c$ [µs]**	**C_i^{deg}/f_c [µs]**
engine	16	0.75	23	8
elevator	15	0.75	22	8
aircraft_dynamics	14	0.75	161	54
h_filter	13	1.5	11	4
az_filter	12	1.5	12	4
Vz_filter	11	1.5	12	4
q_filter	10	1.5	11	4
Va_filter	9	1.5	12	4
altitude_hold	8	3	6	2
Vz_control	7	3	6	2
Va_control	6	3	6	2

Note that values relating to execution times are a function of parameter f_c

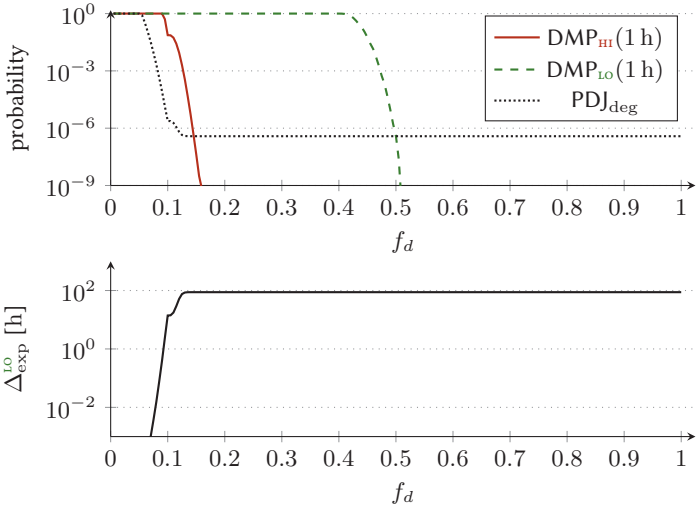
execution time C_i^{thr} is $\Pr(\mathcal{C}_i > C_i^{\text{thr}}) = 10^{-8}$, for every HI task. For the maximal system-level backlog, we used $B_{\max} = 5\,\text{ms}$. The hyperperiod lasts for $60\,\text{ms}$, and inside one there are 500 LO jobs and 19 HI jobs. Regardless of the parameter f_c, the utilization of LO tasks is 5.73 times higher than the utilization of HI tasks.

In Figure 4.2a, the plots have results when deadlines are fixed ($f_d = 1$), but execution times values from Table 4.4 are varied with $f_c \in (0, 7.5]$. In the two plots of Figure 4.2b, shown are results when deadlines are varied $f_d \in (0, 1]$, but all execution time values are fixed ($f_c = 2$).

Results. As expected, the deadline miss probability per hour for both HI and LO jobs, $\text{DMP}_{\text{HI}}(1\,\text{h})$ and $\text{DMP}_{\text{LO}}(1\,\text{h})$, increases as the



(a) With various utilization, and fixed deadlines $f_d = 1$



(b) With scaled deadlines, and fixed utilization $f_c = 2$

Figure 4.2: Metrics characterizing the sample task-set

utilization increases, or as the deadlines become more constrained. In this example, $\mathrm{DMP}_{\text{LO}}(1\,\mathrm{h})$ is larger than $\mathrm{DMP}_{\text{HI}}(1\,\mathrm{h})$, even though HI criticality tasks have the lowest priority. This is mainly because there are more LO than HI jobs, i.e., 500 versus 19 jobs per hyperperiod. As for the probability that a LO job is released in degraded mode, $\mathrm{PDJ}_{\mathrm{deg}}$, we notice it follows a similar trend. In this experiment, the value never goes to zero, because there is always a non-zero probability a LO→HI criticality mode switch occurs.

In Figure 4.2b, the expected duration of LO mode is shown to resemble the inverse of $\mathrm{PDJ}_{\mathrm{deg}}$. Except when the deadlines are very constrained ($f_d < 0.12$), LO-criticality mode lasts for an expected $\Delta^{\text{LO}}_{\text{exp}} = 88\,\mathrm{h}$ before a trigger event occurs. The maximal duration of HI-criticality mode $\Delta^{\text{HI}}_{\text{max}}$ depends only on the system utilization. This is shown in Figure 4.2a, as a function of f_c. The value is $1.1\,\mathrm{ms}$ for $f_c = 2$, and $21.7\,\mathrm{ms}$ for $f_c = 7.5$. Both values are smaller than $\Delta^{\text{LO}}_{\text{exp}}$ by orders of magnitude.

4.7.2 Randomized Systems

Now we continue, and present three further experiments. They demonstrate the impact of three design parameters on schedulability: the system utilization, the probability that a HI tasks overruns its execution time threshold C_i^{thr}, and the choice of the maximal system-level backlog.

More specifically, the first experiment shows whether task-sets of different system utilizations are $(\sigma_{\text{HI}}, \sigma_{\text{LO}}, \sigma_{\mathrm{deg}})$-schedulable using probabilistic mixed-criticality (pMC) scheduling, as well as other scheduling schemes.

The second and third experiments compare pMC with the probabilistic but non-MC scheme pDMPO. They demonstrate that pMC leads to improved schedulability, except when HI-criticality mode is entered too often, either because of the first or the third mode switch trigger, respectively.

For all three experiments, tasks were randomly generated as described below.

Task-set generation. For each of the three experiments presented, the UUnifast-Discard algorithm [DB11] was used to randomly gen-

erate task-sets, with the following parameters we found reasonable.

- First, periods and maximal execution times in LO-criticality mode (C_i^{thr} values for HI tasks and $C_i^{\max}$ for LO tasks) were generated by the UUnifast algorithm. Periods were chosen between 50 µs, 100 µs, 200 µs, 250 µs, 500 µs, and 1000 µs.
- All initial phases were set to 0, and tasks' deadlines are equal to their period.
- Then, every task's criticality is assigned to be HI with a probability of 0.5 (i.e., parameter $CP = 0.5$).
- For every HI task, the WCET $C_i^{\max}$ is a fixed multiplier of the corresponding threshold C_i^{thr}, $C_i^{\max} = 1.5 \cdot C_i^{\text{thr}}$ (i.e., parameter $CF = 1.5$). For LO tasks, their degraded WCET is set to be a third of their actual WCET, $C_i^{\text{deg}} = 0.33 \cdot C_i^{\max}$.
- To model task execution times $\mathcal{C}_i$, we have used Weibull distributions as explained at the beginning of this section. The probability each HI job $\tau_{i,j}$ overruns its execution time threshold is $\Pr(\mathcal{C}_i > C_i^{\text{thr}}) = 10^{-8}$, unless stated otherwise.
- The number of tasks per task-set is 60.
- Finally, the maximum backlog $B_{\max}$ is 500 µs, unless stated otherwise.

For the system utilization and other details, we refer the reader to the setup section of each experiment.

Priority assignment. For the probabilistic scheduling schemes pMC and pDMPO, we have used deadline monotonic priority assignment. Note that [MBS+11] shows that this assignment is in general not optimal for probabilistic systems, they suggest instead Audsley's priority assignment algorithm. For the deterministic scheduling schemes, AMC uses Audsley's priority assignment which is optimal for this scheme, while DMPO by definition uses deadline monotonic priorities.

4.7.2.1 'Utilization' Experiment

In this first experiment, we examine the schedulability of systems with various system utilizations. More precisely, we check whether randomly generated systems of utilization 0.1 through

2.0 are $(\sigma_{\text{HI}}, \sigma_{\text{LO}}, \sigma_{\text{deg}}) = (10^{-8}, 10^{-6}, 10^{-5})$-schedulable under probabilistic mixed-criticality (pMC) scheduling, under a probabilistic but non-MC scheme pDMPO, as well as under deterministic baseline schemes: DMPO, AMC, and UB-HL. We also examine the values relevant to pMC scheduling as functions of maximum system utilization: the probability of deadline miss per hour for HI or LO jobs $\text{DMP}_{\text{HI}}(1\,\text{h})$ and $\text{DMP}_{\text{LO}}(1\,\text{h})$, and the probability of degraded LO jobs PDJ_{deg}.

Setup. We ranged the system utilization from 0.1 to 2.0 with 0.1 steps, and for each step we created 1000 task-sets according to the previously given description. To reiterate, the following parameters were used: the ratio between the WCET $C_i^{\max}$ and execution time threshold C_i^{thr} for every HI task is $CF = C_i^{\max}/C_i^{\text{thr}} = 1.5$, the probability each task is assigned HI criticality is $CP = 0.5$, the probability a HI job overruns its execution time threshold $\Pr(\mathcal{C}_i > C_i^{\text{thr}}) = 10^{-8}$, the degradation of LO tasks is $C_i^{\text{deg}} = 0.33 \cdot C_i^{\max}$, there are 60 tasks in each task-set, and the maximal system-level backlog is $B_{\max} = 500\,\mu\text{s}$.

Tasks' execution times $\mathcal{C}_i$ depend on the utilization and task-set in question. We found the mean of the execution times to be between $2.84\,\mu\text{s}$ and $16.38\,\mu\text{s}$, with the maximal execution time $C_i^{\max}$ among all tasks in a task-set being between $21\,\mu\text{s}$ and $387\,\mu\text{s}$.

Results. Figure 4.3 presents the most important result of our experiments. For different system utilizations, the $(10^{-8}, 10^{-6}, 10^{-5})$-schedulability under various scheduling schemes is given in Figure 4.3 Top. To understand better how utilization impacts pMC schedulability, Figure 4.3 Middle and Bottom shows statistics on the $\text{DMP}_{\text{HI}}(1\,\text{h})$, $\text{DMP}_{\text{LO}}(1\,\text{h})$ and PDJ_{deg} metrics. The box-plots visualize the 10^{th}, 25^{th}, 50^{th}, 75^{th}, and 90^{th} percentile of each metric.

Regarding the three deterministic schemes, we see that they perform similarly as in related work, for example [BBD11b]. Remember that for deterministic schemes, a task-set is either 'completely' schedulable or it is not, as there is no notion of probabilities.

In Figure 4.3 Top, we can see that deadline monotonic priority ordering (DMPO) has the lowest schedulability among all tested

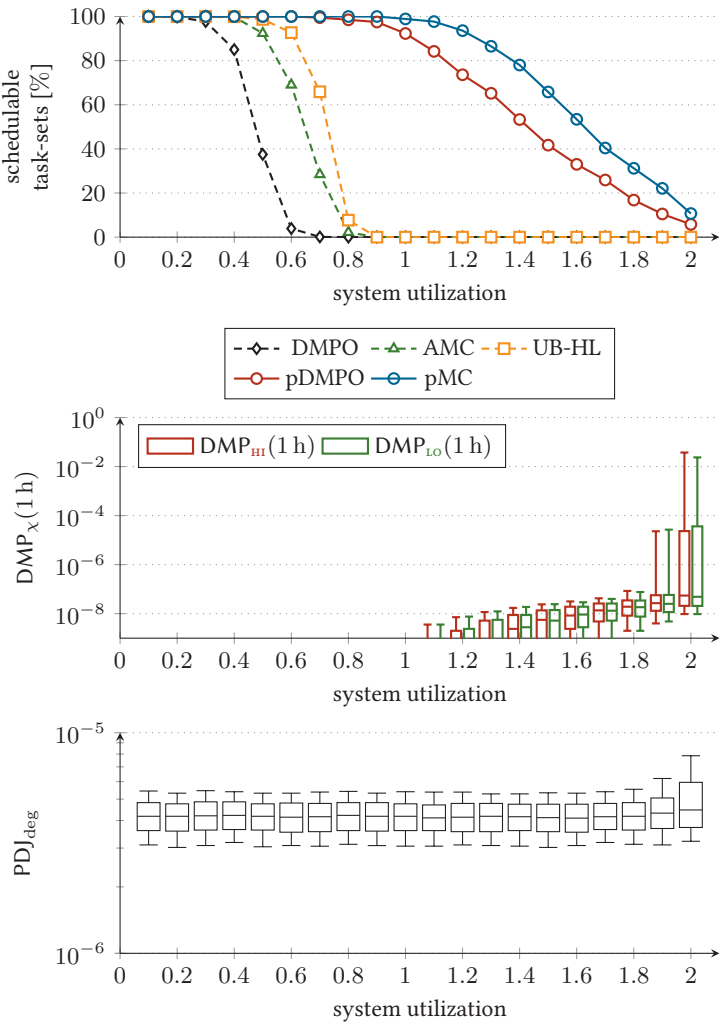


Figure 4.3: $(10^{-8}, 10^{-6}, 10^{-5})$-schedulability of task-sets as a function of utilization under pMC and other schemes (Top), and the impact utilization has on $\mathrm{DMP}_{\mathrm{LO}}(1\,\mathrm{h})$, $\mathrm{DMP}_{\mathrm{HI}}(1\,\mathrm{h})$ (Middle) and $\mathrm{PDJ}_{\mathrm{deg}}$ (Bottom)

schemes. This is because DMPO attempts to schedule a task-set using only WCET ($C_i^{\max}$) values. The adaptive mixed-criticality (AMC) scheme performs better, as it performs a LO→HI mode switch every time HI jobs need more execution time. Still, the schedulability of deterministic fixed priority preemptive schemes is upper-bounded by the UB-HL bound.

For the probabilistic schemes pDMPO and pMC, we can confirm that they outperform deterministic schemes. Probabilistic schemes allow a system with a utilization greater than one to be schedulable, because they take into account the low probability that a long execution time is observed. Let us first focus on probabilistic deadline monotonic priority ordering (pDMPO). We understand from Díaz et al. [DGK+02] that deadline misses under pDMPO happen when the backlog is large, i.e., when one or more jobs take a long time to execute. The bigger the utilization is, the likelier it is that the backlog is large.

As for probabilistic mixed-criticality (pMC), it features three LO→HI mode switch triggers. All three triggers are indicators that the backlog is large: the first trigger activates when a HI job executes for a long time, the second trigger indicates that a HI job missed its deadline due to a large backlog blocking its execution, and finally the third trigger explicitly notes that the system-level backlog is too large. After detecting these high-backlog situations, the system under pMC transitions to HI-criticality mode where LO jobs are degraded, and thus the backlog is decreased. This ensures that deadline miss probabilities of both LO and HI tasks are reduced, at the cost of having some LO jobs released in degraded mode. Most importantly, this is demonstrated in Figure 4.3 Top, where pMC outperforms pDMPO as well as all other schemes. Furthermore, in Figure 4.3 Middle, we see how both $\mathrm{DMP}_{\mathrm{HI}}(1\,\mathrm{h})$ and $\mathrm{DMP}_{\mathrm{LO}}(1\,\mathrm{h})$ increase gradually with the increase of utilization. The small difference between $\mathrm{DMP}_{\mathrm{HI}}(1\,\mathrm{h})$ and $\mathrm{DMP}_{\mathrm{LO}}(1\,\mathrm{h})$ comes from the fact that the system switches to HI-criticality mode whenever a HI jobs overruns its C_i^{thr} threshold, which helps HI jobs keep their deadline. Finally, Figure 4.3 Bottom shows the probability a LO job is released with degradation. This slight increase is a sign of being in HI-criticality mode more often, and this quantifies the cost of probabilistic mixed-criticality scheduling.

4.7.2.2 'Execution Threshold' Experiment

In this experiment, we varied a design parameter relating to tasks' execution times $\mathcal{C}_i$: the probability that a HI job overruns its execution time threshold C_i^{thr}. We then inspected how this impacts schedulability under pMC and the probabilistic non-mixed-criticality pDMPO scheme. Because we used a utilization of 1.4, deterministic schemes could not schedule any task-sets. The probability each HI job $\tau_{i,j}$ overruns its execution time threshold $\Pr(\mathcal{C}_i > C_i^{\text{thr}})$ is ranged from $5 \cdot 10^{-12}$ to 10^{-4}. Ultimately, this experiment demonstrates that it makes sense to use probabilistic mixed-criticality scheduling if HI-criticality mode is not entered too often, and the importance of the PDJ_{deg} metric is justified.

Setup. A total of 16 configurations, each with 1000 task-sets, were generated for this experiment. The configurations have the same parameters, except for the probability each HI job $\tau_{i,j}$ overruns its execution time threshold $\Pr(\mathcal{C}_i > C_i^{\text{thr}})$. The following values for $\Pr(\mathcal{C}_i > C_i^{\text{thr}})$ were used: $\{5 \cdot 10^{-12}, 10^{-11}, 5 \cdot 10^{-11}, \ldots, 10^{-4}\}$. Besides this, the system utilization for all configurations is 1.4, while all other parameters are according to the description mentioned at the beginning of Section 4.7.2.

Regardless of the fact that $\Pr(\mathcal{C}_i > C_i^{\text{thr}})$ is varied by 8 orders of magnitude, we found that the mean execution time per configuration changes little. It is between $8.69\,\mu\text{s}$ and $8.70\,\mu\text{s}$. Among all tasks in every task-set, the worst case execution time $C_i^{\max}$ is $287\,\mu\text{s}$.

Results. The results are shown in Figure 4.4. In the top figure, we present $(10^{-8}, 10^{-6}, 10^{-5})$- and $(10^{-8}, 10^{-6}, 1)$-schedulability under pMC, as well as $(10^{-8}, 10^{-6}, -)$-schedulability under pDMPO. Since by definition $\text{PDJ}_{\text{deg}} \leq 1$, we can interpret $(\sigma_{\text{HI}}, \sigma_{\text{LO}}, \sigma_{\text{deg}} = 1)$-schedulability under pMC as a schedulability test which ignores the PDJ_{deg} metric. In the middle and bottom figures, the box-plots visualize the 10^{th}, 25^{th}, 50^{th}, 75^{th}, and 90^{th} percentile of each metric.

In Figure 4.4 Top, let us first focus on comparing pDMPO and pMC when $\sigma_{\text{deg}} = 1$. In this case, when the PDJ_{deg} metric is ignored, we see that more task-sets are always schedulable under

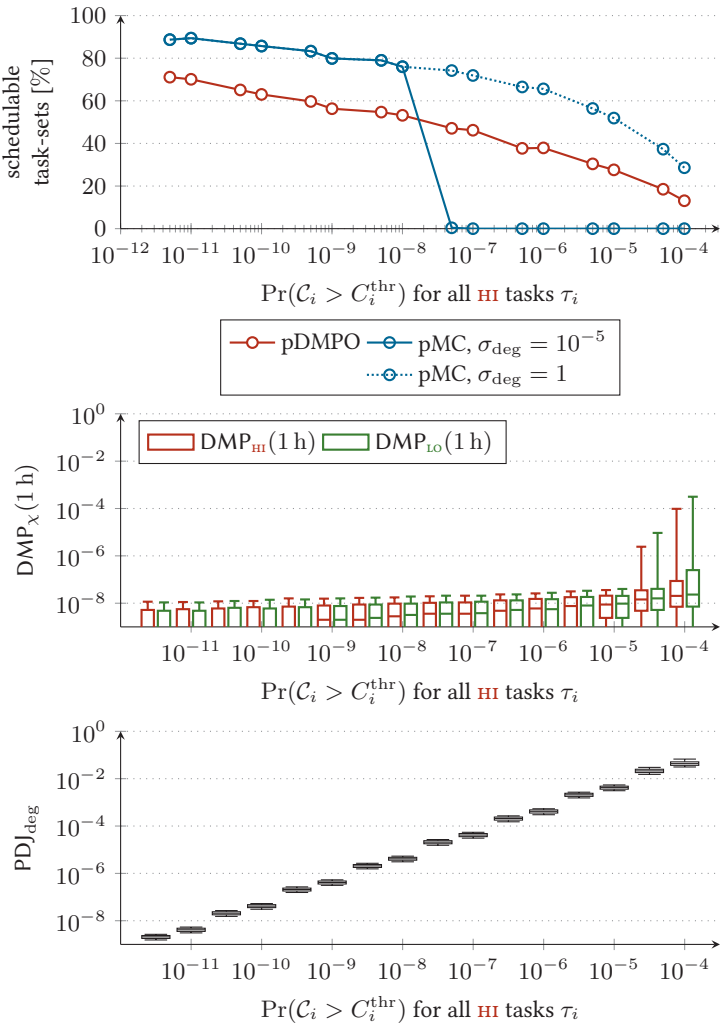


Figure 4.4: $(10^{-8}, 10^{-6}, 10^{-5})$-schedulability of task-sets under pMC and pDMPO, and $(10^{-8}, 10^{-6}, 1)$-schedulability under pMC, for various probabilities that a HI job overruns its execution time threshold $\Pr(\mathcal{C}_i > C_i^{\text{thr}})$ (Top), and the impact this value has on $\text{DMP}_{\text{LO}}(1\,\text{h})$, $\text{DMP}_{\text{HI}}(1\,\text{h})$ (Middle) and PDJ_{deg} (Bottom)

pMC than under pDMPO. The reasons pMC scheduling is better in this case are the same reasons as in the 'utilization' experiment: by switching to HI-criticality mode after certain triggering events, the system under pMC scheduling reduces the backlog in these situations, which ultimately makes deadline misses less likely.

Now, let us examine pMC with a realistic $\mathrm{PDJ}_{\mathrm{deg}}$ bound, i.e. $\sigma_{\mathrm{deg}} = 10^{-5}$. As shown in the top figure, it is clear that there exists a limit after which pMC scheduling is not useful at all, as it leads to too much degradation. This can be understood by viewing Figure 4.4 Bottom, where we see the cost of switching to HI-mode. On one extreme case, when $\Pr(\mathcal{C}_i > C_i^{\mathrm{thr}}) = 10^{-4}$, the system switches to HI-mode often, on average once every $48.93\,\mu\mathrm{s}$ (not shown in figure). Then, an average ratio of 0.046 of LO jobs are released in degraded mode. In a moderate case, for $\Pr(\mathcal{C}_i > C_i^{\mathrm{thr}}) = 10^{-8}$, HI jobs overrun their execution time threshold C_i^{thr} less often, and LO-mode lasts on average $8.34\,\mathrm{min}$. Here, an average ratio of $4.19 \cdot 10^{-6}$ of LO jobs are degraded. Finally, on the other extreme case, when $\Pr(\mathcal{C}_i > C_i^{\mathrm{thr}}) = 5 \cdot 10^{-12}$, LO-mode lasts for $278\,\mathrm{h}$ on average, and only a tiny fraction of $2.09 \cdot 10^{-9}$ LO jobs are released in degraded mode. For many realistic applications, there exists a limit on the degradation which can be tolerated, before a complete loss of function happens. Thus we argue that this experiment demonstrates why the $\mathrm{PDJ}_{\mathrm{deg}}$ metric is crucial for probabilistic mixed-criticality scheduling.

Finally, let us comment on $\mathrm{DMP}_{\mathrm{LO}}(1\,\mathrm{h})$ and $\mathrm{DMP}_{\mathrm{HI}}(1\,\mathrm{h})$, found in Figure 4.4 Middle. These are similar, except the values become larger for higher $\Pr(\mathcal{C}_i > C_i^{\mathrm{thr}})$ values. We have found that this increase in $\mathrm{DMP}_{\chi}(1\,\mathrm{h})$ appears as a result of pessimistic assumptions introduced in Definition 4.12. We comment more about this pessimism the next experiment.

4.7.2.3 'Maximal Backlog' Experiment

In the final experiment on randomized systems, the maximum system-level backlog $B_{\max}$ was varied. This affects how often HI-criticality mode is entered, while it has no effect on the LO-criticality mode. When the occurrence of HI-criticality mode is artificially increased, we can see the pessimism in the analysis of that mode

– which we found mostly to be introduced by pessimistic assumptions on the initial conditions in HI-mode, as per Definition 4.12. As in the previous experiment, we tested the $(10^{-8}, 10^{-6}, 10^{-5})$-schedulability of task-sets under pMC and pDMPO, as well as $(10^{-8}, 10^{-6}, 1)$-schedulability under pMC scheduling.

Setup. For this experiment, we first generated 1000 task-sets with a system utilization of 1.2. This high utilization guarantees that no deterministic scheme can be used to schedule task-sets. All parameters except for the maximum system-level backlog are according to the description at the beginning of this section. Then, the maximum system-level backlog $B_{\max}$ was varied from $40\,\mu s$ to $600\,\mu s$, and all of the 1000 task-sets are analyzed for every $B_{\max}$ value. Each generated task-set has 60 tasks, the mean execution time among all tasks in every task-set is $10.61\,\mu s$, while the maximum execution time overall is $255\,\mu s$.

Results. Figure 4.5 visualizes the results of this experiment. As done previously , we conducted a $(10^{-8}, 10^{-6}, 10^{-5})$-schedulability test under pMC and pDMPO, as well as a schedulability test under pMC when the $\mathrm{PDJ}_{\deg}$ metric is ignored (i.e. $\sigma_{\deg} = 1$). The box-plots visualize the 10^{th}, 25^{th}, 50^{th}, 75^{th}, and 90^{th} percentile of each evaluated metric. By definition, the maximum system-level backlog $B_{\max}$ does not impact scheduling under pDMPO at all, so the schedulability under this scheme is constant.

Regarding the impact on pMC scheduling, specifically on the two $\mathrm{DMP}_{\chi}(1\,\mathrm{h})$ values, we see two cases. On the one hand, when the maximum system-level backlog $B_{\max}$ is sufficiently large, i.e., $\geq 200\,\mu s$, we see that it has a negligible impact on $\mathrm{DMP}_{\text{HI}}(1\,\mathrm{h})$ and $\mathrm{DMP}_{\text{LO}}(1\,\mathrm{h})$ values. On the other hand, when a small $B_{\max}$ causes HI-mode to be entered often, $\mathrm{DMP}_{\text{HI}}(1\,\mathrm{h})$ and $\mathrm{DMP}_{\text{LO}}(1\,\mathrm{h})$ both increase. Ideally, how often HI-mode is entered should not impact nighter $\mathrm{DMP}_{\text{HI}}(1\,\mathrm{h})$ nor $\mathrm{DMP}_{\text{LO}}(1\,\mathrm{h})$. The increase is a result of pessimism introduced in point 4 of Definition 4.12. As the reader recalls, there we make a pessimistic assumption that all HI jobs are overrunning their execution time thresholds C_i^{thr} at the time of the mode switch. This pessimistic assumption is mainly introduced

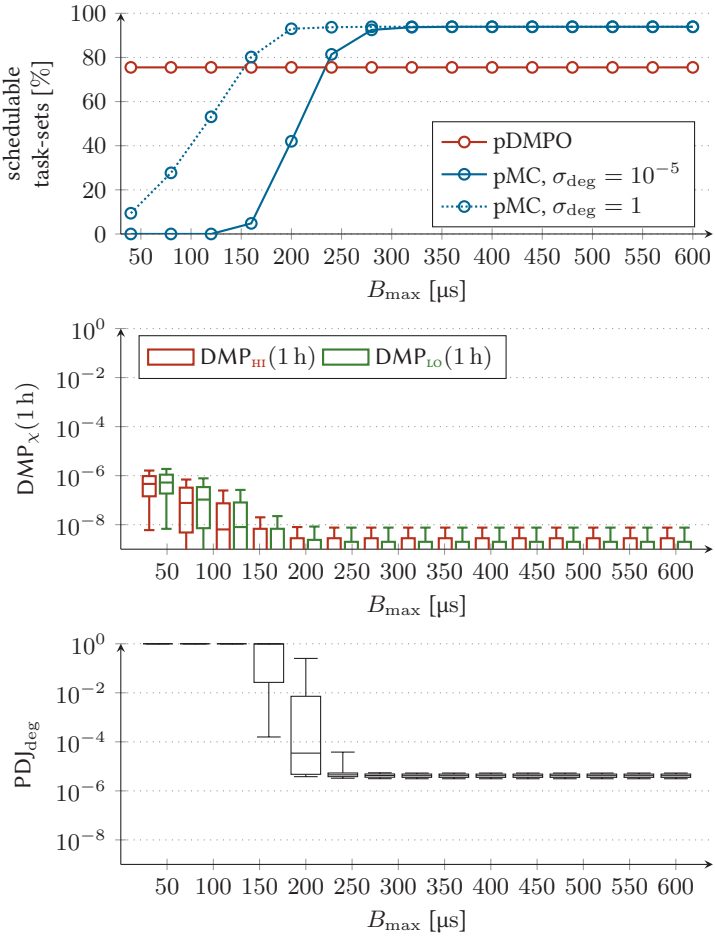


Figure 4.5: $(10^{-8}, 10^{-6}, 10^{-5})$-schedulability of task-sets under pMC and pDMPO, and $(10^{-8}, 10^{-6}, 1)$-schedulability under pMC, for various $B_{\max}$ values (Top), and the impact this value has on $\text{DMP}_{\text{LO}}(1\,\text{h})$, $\text{DMP}_{\text{HI}}(1\,\text{h})$, and PDJ_{deg} (Bottom)

to reduce the number of cases under which HI-criticality mode is analyzed.

The impact the backlog $B_{\max}$ has on PDJ_{deg} is straightforward. As HI-mode is entered more often, PDJ_{deg} increases. Because of this increase, we find that few task-sets are $(10^{-8}, 10^{-6}, 10^{-5})$-schedulable under pMC for $B_{\max}$ values less than 200 µs. We can therefore conclude thus the pessimism of HI-criticality mode analysis does not play a major role in the schedulability analysis of task-sets under realistic requirements for the maximal permitted degradation of LO jobs σ_{deg}. Finally, we observe again the main result from the 'execution threshold' experiment: probabilistic mixed-criticality (pMC) scheduling is better than the non-MC scheme pDMPO, except when HI-criticality mode is entered too often.

4.8 Extensions and Future Work

In this section, we present an extension to the system model and analysis which enables less pessimistic results for LO-criticality mode, and we discuss limitations and possible future research directions.

4.8.1 LO-criticality Mode Model Extension

Throughout this chapter, we modeled the execution time of HI tasks in LO-criticality mode with the random variable $\mathcal{C}_i^{\text{LO}}$, as introduced in Definition 4.2 and illustrated by Figure 4.1b. We used this model as it simplified finding initial conditions for the analysis of HI-criticality mode. Now, we would like to detail another way to model execution times of HI tasks in LO-criticality mode, which is often found in related work, i.e., [MDCGE17].

Let us model the execution time of HI tasks in LO-criticality mode with the conditional probability $\mathcal{C}_i^{\text{LO}\star}$, the condition being the execution time threshold C_i^{thr} is not overrun. This value is often called 'truncated' execution time, and it is defined below.

Definition 4.13 *Truncated Execution Time*: Random variable $\mathcal{C}_i^{\text{LO}\star}$ models the execution time of HI tasks τ_i, under the condition that

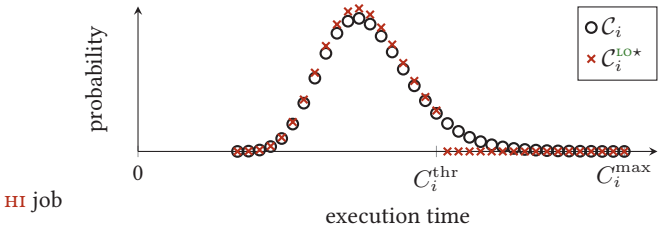


Figure 4.6: Task execution times, with named values and truncated execution time $\mathcal{C}_i^{\text{LO}\star}$

they do not execute for longer than C_i^{thr}:

$$p_{\mathcal{C}_i^{\text{LO}\star}}(u) = \Pr(p_{\mathcal{C}_i}(u) = u | u \leq C_i^{\text{thr}}) = \begin{cases} {}^{p_{\mathcal{C}_i}(u)}/_{\Pr(\mathcal{C}_i \leq C_i^{\text{thr}})} & u \leq C_i^{\text{thr}} \\ 0 & u > C_i^{\text{thr}} \end{cases} \tag{4.29}$$

By limiting the probability distribution this way, execution times above the threshold C_i^{thr} are not allowed. This can be seen in Figure 4.6, where the conditional distribution retains the same shape as the original before the threshold, and is equal to zero afterwards.

Note that the 'trimmed' execution times $\mathcal{C}_i^{\text{LO}}$ are by definition greater or equal to the equivalent 'truncated' values $\mathcal{C}_i^{\text{LO}\star}$, with regards to the $\preceq$ comparison (Definition 4.1).

Modification to the stochastic analysis. Using $\mathcal{C}_i^{\text{LO}\star}$ enables a less pessimistic analysis. Unfortunately, these variables can not be used to easily construct a safe initial backlog for HI-criticality mode, as per Definition 4.12. We are thus required to perform two LO-mode analyses: one using 'trimmed' execution time models $\mathcal{C}_i^{\text{LO}}$ for calculating initial conditions in HI-mode, and another using 'truncated' execution time models $\mathcal{C}_i^{\text{LO}\star}$ for calculating deadline miss probabilities in LO-criticality mode $\mathsf{DMP}_{\chi}(T)$.

Precisely speaking, and using Algorithm 4.2 to represent our stochastic analysis, with 'trimmed' execution times $\mathcal{C}_i^{\text{LO}}$ one needs to do the LO-criticality analysis, but calculating deadline miss probabilities of jobs can be skipped (lines 4 and 7). On the other hand,

with 'truncated' execution times $\mathcal{C}_i^{\text{LO}\star}$, one only needs to calculate the deadline miss probabilities, which is done in lines 2, 4 and 7.

For simulated experiments in Section 4.7.2, the numerical impact of this extension was found to be negligible. We have thus excluded it from the main analysis in this chapter, but kept here for completion.

4.8.2 Limitations and Future Work

Our pMC analysis applies for fixed priority preemptive scheduling, but it could be extended to dynamic scheduling schemes as well. Probabilistic response-time calculus already exists for dynamic schemes [DGK+02]. In addition, dynamic-priority mixed-criticality schemes are found to be relevant [BBD+11a, GSY15].

Regarding our proposed scheme, its main limitation is the pessimism of the analysis of HI-criticality mode. This pessimism is due to the fact that we have a single analysis whatever the reason for making the LO→HI transition was.

In a future work, it would be possible to do a less pessimistic analysis of HI-mode by deconstructing the analysis into three subclasses, one for each LO→HI mode switch reason. For example, if a mode switch was caused by a maximal system-level backlog exceedance, the initial backlog would surely be exactly $B_{\max}$. If the mode switch was not caused by an overrunning job, there would be no need to assume that carry-over jobs of HI criticality surely overrun. If the mode switch was caused by an overrunning HI job, one could introduce cases depending on which job cause the mode switch.

The pessimism of the analysis for the LO-criticality mode could be reduced as well, but arguably this would bear less fruit. One idea here is to estimate the percentage of time a system spends in LO-criticality mode. In calculating $\mathsf{DMP}_\chi(T)$ in our work, we assumed the system is in LO-mode all the time. Replacing this assumption with a better estimate would bring improvement, however only for systems which spend a non-negligible amount of time in HI-criticality mode, which is usually not assumed to be the case.

4.9 Summary

Modeling tasks' execution times with random variables in Vestal's mixed-criticality model allows for a schedulability analysis based on the 'probability of deadline miss per hour'. We presented a dual-criticality system which operates in either LO- or HI-criticality mode. In LO-criticality mode, both LO and HI jobs run normally, but certain optimism towards HI jobs exists: they are required not to overrun their C_i^{thr} execution time threshold, a value analogue to the optimistic WCET in Vestal's model. HI-criticality mode is entered when a violation of this optimistic condition is detected, or when one of the following two events happen: a HI job misses its deadline, or the system-level backlog exceeds its maximal value. In this mode, LO jobs are degraded by having a shorter time budget for execution, so HI jobs have more resources available. This mode lasts until the system becomes idle.

To characterize such a system, we first defined $(\sigma_{\text{HI}}, \sigma_{\text{LO}}, \sigma_{\text{deg}})$-schedulability, which quantifies the soft schedulability of a probabilistic mixed-criticality system. The schedulability conditions determine whether the *probability of deadline miss per hour for* HI *jobs*, the *probability of deadline miss per hour for* LO *jobs* and the *probability a* LO *job is started in its degraded mode* are less that the given $(\sigma_{\text{HI}}, \sigma_{\text{LO}}, \sigma_{\text{deg}})$ limits.

Then, we presented an analysis approach. This was done by splitting the system into two – the LO- and the HI-criticality mode system – and combining the results. On one hand, a steady state analysis was carried out for LO-criticality mode, in which the system is expected to stay for a long time. This enabled us to pessimistically bound the deadline miss probability of each job, which we then used to find the probability that any job misses its deadline while in LO-mode in a certain time period. On the other hand, a simulation of the transient HI-criticality mode was used to bound its duration, and to obtain the probability of deadline miss of jobs inside it. This, together with the probability a LO→HI mode switch happens, enabled us to find the probability any job misses its deadline while in HI-mode in a certain time period.

Finally, simulation results illustrate all of the metrics on a sample task-set, and experiments involving schedulability analysis show

how various design choices impact schedulability. Here, we show how probabilistic mixed-criticality scheduling compares to other schemes, and make a clear case that using pMC makes sense for most cases, except when the amount of LO job degradation is too high.

4.A Notations

Table 4.5: Notations

random variable	$\mathcal{A}$
probability function	$p_{\mathcal{A}}(u) = \Pr(\mathcal{A} = u)$
vector representation	$[p_{\mathcal{A}}(u_{\min}) \dots p_{\mathcal{A}}(u_{\max})]^{\intercal}$
comparison of random variables	$\mathcal{A} \succeq \mathcal{B}$
upper bound on random variable	$\overline{\mathcal{A}} \succeq \mathcal{A}$
task set, number of tasks, task i, job j	Π, N, τ_i, $\tau_{i,j}$
task period	T_i
length of a hyperperiod	HP
relative deadline of task i	D_i
absolute deadline of $\tau_{i,j}$	$d_{i,j}$
release time of $\tau_{i,j}$	$r_{i,j}$
initial phase	ϕ_i
criticality level	$\chi_i \in \{\text{LO}, \text{HI}\}$
probabilistic execution time	$\mathcal{C}_i \leq C_i^{\max}$
threshold of execution time for HI jobs	C_i^{thr}
maximal execution time for degraded LO jobs	C_i^{deg}
trimmed execution time of HI jobs	$\mathcal{C}_i^{\text{LO}}$
threshold of backlog	$B_{\max}$
response time of $\tau_{i,j}$	$\mathcal{R}_{i,j}$
deadline miss probability of $\tau_{i,j}$	$\mathsf{DMP}_{i,j}$
upper bound on deadline miss probability of $\tau_{i,j}$	$\overline{\mathsf{DMP}}_{i,j}$
deadline miss prob. within T for HI and LO jobs	$\mathsf{DMP}_{\text{HI}}(T)$, $\mathsf{DMP}_{\text{LO}}(T)$
degradation probability of LO jobs	$\mathsf{PDJ}_{\text{deg}}$
backlog at priority level i and time t	$\mathcal{B}_i(t)$
convolution operator	$\oplus$
shrinking function of $\mathcal{B}$ at value m	$\text{shrink}(\mathcal{B}, m)(u)$
backlog function	$\texttt{bsse}\,(\mathcal{B}_i(t),\, \Pi,\, i,\, t,\, u)$
response time function	$\texttt{rta}\,(\mathcal{B}_i(r_{i,j}),\, \Pi,\, \tau_{i,j})$
maximum duration of any HI-mode execution	$\Delta_{\max}^{\text{HI}}$
expected length of a LO-mode execution	$\Delta_{\text{exp}}^{\text{LO}}$
w.c. prob. of a LO→HI mode switch at time $t \in \{0, \dots \text{HP} - 1\}$ within a hyperperiod	$\text{P}_{\text{LO}\to\text{HI}}(t)$
w.c. prob. of a LO→HI mode switch at time $t \in \{0, \dots \text{HP} - 1\}$ due to deadline miss	$\text{P}_{dm}(t)$

continues on next page

Notations, continued

w.c. prob. of a LO→HI mode switch at time $t \in \{0, \dots \text{HP} - 1\}$ due to backlog exceedance	$\text{P}_{be}(t)$
w.c. prob. of a LO→HI mode switch at time $t \in \{0, \dots \text{HP} - 1\}$ due to execution time overrun	$\text{P}_{ov}(t)$
w.c. prob. of at least one LO→HI mode switch within a hyperperiod	$\text{P}^{\text{HP}}_{\text{LO}\to\text{HI}}$
w.c. prob. of at least one deadline miss of any χ job during any HI mode started within a hyperperiod	$\text{DMP}^{\text{HI}}_{\chi}$
w.c. prob. of at least one deadline miss of any χ job during a hyperperiod during LO mode execution	$\text{DMP}^{\text{LO}}_{\chi}$
w.c. task set of HI-mode if starting at t	$\widehat{\Pi}(t)$
w.c. duration of HI-mode if starting at t	$\Delta^{\text{HI}}_{\max}(t)$
w.c. prob. of at least one deadline miss of any χ job during any HI mode started at t	$\text{DMP}^{\text{HI}}_{\chi}(t)$

5

Conclusion and Outlook

Embedded systems are often designed with adaptive operation in mind. Plenty of adaption strategies exist, developed for diverse applications, and targeting different uncertainty sources. Generally speaking, a system adapts to its variable environment by changing its mode of operation online, either reactively or proactively. Doing so causes a temporary change of functionality, for example, some tasks might be degraded when resources are scarce. In this way, adaptive systems promise to avoid relevant limitations in the long term, even if a reduction in performance may occur locally.

Unfortunately, it is a challenge to guarantee such constraints. In order to understand and predict the operation of an adaptive system, one needs to take into account not only the non-determinism present in the environment but also the system's adaptive behavior.

To this end, this thesis explores two sources of uncertainties: the volatile and intermittent harvested energy, and the unpredictable execution times of tasks. Both a worst-case and a stochastic Markov analysis are formalized to tackle them, ultimately providing worst-case and probabilistic performance metrics regarding energy-neutrality and real-time constraints. We now recapitulate the main contributions of each chapter before commenting on limitations and possible future research directions.

5.1 Contributions

Worst-case analysis of energy harvesting systems (Chapter 2). We start with embedded systems harvesting energy from their environment, which are subject to energy-neutral operation constraints. We are the first to discuss the optimal use of an additional backup battery for such systems. Here, the system's goal is to realize a minimal service level with the least amount of backup energy used, while also maximizing the long-term utility. To this end, we derived an optimal though practically unfeasible solution to the problem. We next introduced a proactive adaption strategy, finite horizon control, based on model predictive control. For the solar energy harvesting case, we derived novel energy estimation schemes based on atmospheric transmittance. These methods surpass the state-of-the-art in terms of accuracy, and they have low computational and memory overhead, which allows them to be used on small energy harvesting nodes. We finally use trace-based simulations to illustrate worst-case performance metrics for an array of solar energy harvesting systems, deployed at indoor and outdoor locations, as well as a real-world system implementation to verify the accuracy of our analysis.

Stochastic analysis of energy harvesting systems (Chapter 3). We continue with a new Markov analysis of energy harvesting systems. This approach also takes into account the uncertainty of energy consumption created by variability in the execution times of tasks. Using this analysis, we prove that it is possible to safely obtain probabilistic performance metrics for a wide range of adaptivity strategies. Namely, we derive such values as the probability of failure due to energy unavailability, and the probability that the system operates in a certain mode. Experiments illustrate these performance guarantees in diverse solar energy harvesting scenarios targeting multi-year deployment, both for commonly considered outdoor locations, and for highly variable indoor ones. Finally, an example design space exploration is conducted, and the accuracy and pessimism of the analysis are assessed.

Stochastic analysis of MC scheduling (Chapter 4). Last but not least, we focus on mixed-criticality systems with real-time constraints in the probability-of-failure-per-hour format. Using the pWCET abstraction, we formulate a probabilistic mixed-criticality scheduling scheme with appropriate reactive mode switch triggers. This scheme is analyzed using Markov theory and allows us to present important guarantees: the probability of deadline miss per hour, both globally and in each mode of operation, and the probability that a task is executed in degraded mode. Our pMC scheme clearly outperforms state-of-the-art solutions.

5.2 Limitations and Open Problems

Regarding the worst-case analysis of energy harvesting systems, an interesting case is when proactive adaptivity schemes are paired with imprecise energy estimators. Such cases exist in volatile harvesting environments, where proactive schemes under-perform reactive ones. In a future work, one might investigate schemes suited for such scenarios, possibly incorporating information on the estimator's precision online in order to switch from a proactive to a reactive adaptation strategy accordingly.

In its current state, the stochastic analysis of energy harvesting systems can not consider failure recovery mechanisms nor any other kind of hysteresis behavior. Future works tackling this limitation would undoubtedly be beneficial. Furthermore, there is a requirement that random variables modeling harvested energy need to be independent of one time interval to the next. This was justified in our solar energy harvesting scenario for a one week time interval, but for many other scenarios such an assumption is not feasible, i.e., for solar energy harvesting with a daily time interval, weather patterns cause dependencies in harvested energy from one day to the next. Lifting this requirement and incorporating certain forms of dependence between random variables is a natural next step.

In the real-time domain, our own probabilistic mixed-criticality scheduling analysis applies to fixed priority preemptive scheduling, and it could also be extended to dynamic scheduling schemes. We

additionally note that a high level of pessimism in the HI-criticality mode analysis exists, which could be reduced with a richer model of initial conditions in that mode of operation.

Finally, as this thesis has a strong theoretical aspect, a natural extension would be to illustrate many aspects with practical applications. Discussed service adaption strategies for energy harvesting systems, especially when a backup battery is present, are particularly great candidates for long-term deployments with strict requirements on reliability and minimal service levels.

Bibliography

[AAA+07] C. Arnhardt, K. Asch, R. Azzam, R. Bill, T. Fernandez-Steeger, S. Homfeld, A. Kallash, F. Niemeyer, H. Ritter, M. Toloczyki, and K. Walter. Sensor based landslide early warning system – SLEWS. Development of a geoservice infrastructure as basis for early warning systems for landslides by integration of real-time sensors. *Geotechnologien science report*, 10:75–88, October 2007. URL http://media.gfz-potsdam.de/geotechnologien/doc/Science_reports/SR10.pdf.

[ABRW91] N. C. Audsley, A. Burns, M. F. Richardson, and A. J. Wellings. Hard real-time scheduling: the deadline-monotonic approach. *IFAC Proceedings Volumes*, 24(2):127–132, May 1991. doi:10.1016/S1474-6670(17)51283-5.

[AG16] B. Alahmad and S. Gopalakrishnan. A risk-constrained Markov decision process approach to scheduling mixed-criticality job sets. In *Workshop on Mixed Criticality Systems (WMC 2016)*, November 2016. URL https://hal.archives-ouvertes.fr/hal-01403223.

[AG18] B. Alahmad and S. Gopalakrishnan. Risk-aware scheduling of dual criticality job systems using demand distributions. *Leibniz Transactions on Embedded Systems (LITES)*, 5(1):1:1–1:30, October 2018. doi:10.4230/LITES-v005-i001-a001.

[AKTM16] S. Asyaban, M. Kargahi, L. Thiele, and M. Mohaqeqi. Analysis and scheduling of a battery-less mixed-criticality system with energy uncertainty. *ACM Transactions on Embedded Computing Systems*, 16(1), October 2016. doi:10.1145/2964201.

[AM17] Y. Abdeddaïm and D. Maxim. Probabilistic schedulability analysis for fixed priority mixed criticality real-time systems. In *Design, Automation & Test in Europe (DATE 2017)*, pages 596–601. IEEE, March 2017. doi:10.23919/DATE.2017.7927056.

[AOS+12] I. Atzeni, L. G. Ordóñez, G. Scutari, D. P. Palomar, and J. R. Fonollosa. Demand-side management via distributed energy generation and storage optimization. *IEEE Transactions on Smart Grid*, 4(2):866–876, September 2012. doi:10.1109/TSG.2012.2206060.

[BASM16] N. A. Bhatti, M. H. Alizai, A. A. Syed, and L. Mottola. Energy harvesting and wireless transfer in sensor network applications: concepts and experiences. *ACM Transactions on Sensor Networks (TOSN)*, 12(3):1–40, August 2016. doi:10.1145/2915918.

[BBD+11a] S. K. Baruah, V. Bonifaci, G. D'Angelo, A. Marchetti-Spaccamela, S. van der Ster, and L. Stougie. Mixed-criticality scheduling of sporadic task systems. In *19th Annual European Symposium on Algorithms (ESA 2011)*, pages 555–566. Springer, September 2011. doi:10.1007/978-3-642-23719-5_47.

[BBD11b] S. K. Baruah, A. Burns, and R. I. Davis. Response-time analysis for mixed criticality systems. In *32nd Real-Time Systems Symposium (RTSS 2011)*, pages 34–43. IEEE, November 2011. doi:10.1109/RTSS.2011.12.

[BBF+11] J. Beutel, B. Buchli, F. Ferrari, M. Keller, L. Thiele, and M. Zimmerling. X-SENSE: sensing in extreme environments. In *Design, Automation & Test in Europe (DATE 2011)*, pages 1–6. IEEE, March 2011. doi:10.1109/DATE.2011.5763236.

[BD17] A. Burns and R. I. Davis. A survey of research into mixed criticality systems. *ACM Computing Surveys (CSUR)*, 50(6):82:1–82:37, November 2017. doi:10.1145/3131347.

[BDW+16] D. Balsamo, A. Das, A. S. Weddell, D. Brunelli, B. M. Al-Hashimi, G. V. Merrett, and L. Benini. Graceful performance modulation for power-neutral transient computing systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 35(5):738–749, February 2016. doi:10.1109/TCAD.2016.2527713.

[BF11] S. Baruah and G. Fohler. Certification-cognizant time-triggered scheduling of mixed-criticality systems. In *32nd Real-Time Systems Symposium (RTSS 2011)*, pages 3–12. IEEE, November 2011. doi:10.1109/RTSS.2011.9.

[BKT15] B. Buchli, P. Kumar, and L. Thiele. Optimal power management with guaranteed minimum energy utilization for solar energy harvesting systems. In *International Conference on Distributed Computing in Sensor Systems (DCOSS 2015)*, pages 147–158. IEEE, June 2015. doi:10.1109/DCOSS.2015.9.

[BSBT14a] B. Buchli, F. Sutton, J. Beutel, and L. Thiele. Dynamic power management for long-term energy neutral operation of solar energy harvesting systems. In *12th Conference on Embedded Network Sensor Systems (SenSys 2014)*, pages 31–45. ACM, November 2014. doi:10.1145/2668332.2668333.

[BSBT14b] B. Buchli, F. Sutton, J. Beutel, and L. Thiele. Towards enabling uninterrupted long-term operation of solar energy harvesting embedded systems. In *11th European Conference on Wireless Sensor Networks (EWSN 2014)*, pages 66–83. Springer, February 2014. doi:10.1007/978-3-319-04651-8_5.

[BWL+14] Y. Bao, X. Wang, X. Liu, S. Zhou, and Z. Niu. Solar radiation prediction and energy allocation for energy harvesting base stations. In *International Conference on Communications (ICC 2014)*, pages 3487–3492. IEEE, June 2014. doi:10.1109/ICC.2014.6883861.

[BWM+14] D. Balsamo, A. S. Weddell, G. V. Merrett, B. M. Al-Hashimi, D. Brunelli, and L. Benini. Hibernus: sustaining computation during intermittent supply for energy-harvesting systems. *IEEE Embedded Systems Letters*, 7(1):15–18, November 2014. doi:10.1109/LES.2014.2371494.

[CB10] M. Chang and P. Bonnet. Meeting ecologists' requirements with adaptive data acquisition. In *8th Conference on Embedded Networked Sensor Systems (SenSys 2010)*, pages 141–154. ACM, November 2010. doi:10.1145/1869983.1869998.

[CGSH+12] L. Cucu-Grosjean, L. Santinelli, M. Houston, C. Lo, T. Vardanega, L. Kosmidis, J. Abella, E. Mezzetti, E. Quiñones, and F. J. Cazorla. Measurement-based probabilistic timing analysis for multi-path programs. In

24th Euromicro Conference on Real-Time Systems (ECRTS 2012), pages 91–101. IEEE, July 2012. doi:10.1109/ECRTS.2012.31.

[CPS12] A. Cammarano, C. Petrioli, and D. Spenza. Pro-energy: a novel energy prediction model for solar and wind energy-harvesting wireless sensor networks. In *9th International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2012)*, pages 75–83. IEEE, October 2012. doi:10.1109/MASS.2012.6502504.

[CSSJ11] S. Chen, P. Sinha, N. B. Shroff, and C. Joo. Finite-horizon energy allocation and routing scheme in rechargeable sensor networks. In *International Conference on Computer Communications (INFOCOM 2011)*, pages 2273–2281. IEEE, April 2011. doi:10.1109/INFCOM.2011.5935044.

[CVS+07] P. Corke, P. Valencia, P. Sikka, T. Wark, and L. Overs. Long-duration solar-powered wireless sensor networks. In *4th Workshop on Embedded Networked Sensors (EmNets 2007)*, pages 33–37. ACM, June 2007. doi:10.1145/1278972.1278980.

[DB11] R. I. Davis and A. Burns. Improved priority assignment for global fixed priority pre-emptive scheduling in multiprocessor real-time systems. *Real-Time Systems*, 47(1):1–40, January 2011. doi:10.1007/s11241-010-9106-5.

[DBG17] R. I. Davis, A. Burns, and D. Griffin. On the meaning of pWCET distributions and their use in schedulability analysis. In *Real-Time Scheduling Open Problems Seminar (RTSOPS 2017)*, June 2017. URL `https://www.cs.york.ac.uk/rts/publications/Davis2017.html`.

[DCD13] S. DeBruin, B. Campbell, and P. Dutta. Monjolo: an energy-harvesting energy meter architecture. In *11th Conference on Embedded Network Sensor Systems (SenSys 2013)*, pages 18:1–18:14. ACM, November 2013. doi:10.1145/2517351.2517363.

[DCG19a] R. Davis and L. Cucu-Grosjean. A survey of probabilistic schedulability analysis techniques for real-time systems. *Leibniz Transactions on Embedded Systems (LITES)*, 6(1):04:1–04:53, May 2019. doi:10.4230/LITES-v006-i001-a004.

[DCG19b] R. Davis and L. Cucu-Grosjean. A survey of probabilistic timing analysis techniques for real-time systems. *Leibniz*

Transactions on Embedded Systems (LITES), 6(1):03:1–03:60, May 2019. doi:10.4230/LITES-v006-i001-a003.

[DFG+14] G. Durrieu, M. Faugère, S. Girbal, D. G. Pérez, C. Pagetti, and W. Puffitsch. Predictable flight management system implementation on a multicore processor. In *7th Embedded Real Time Software and Systems (ERTS 2014)*, February 2014. URL https://hal.archives-ouvertes.fr/hal-01121700.

[DGK+02] J. L. Díaz, D. F. García, K. Kim, C.-G. Lee, L. Lo Bello, J. M. López, S. L. Min, and O. Mirabella. Stochastic analysis of periodic real-time systems. In *23rd Real-Time Systems Symposium (RTSS 2002)*, pages 289–300. IEEE, December 2002. doi:10.1109/REAL.2002.1181583.

[DK03] A. Devgan and C. Kashyap. Block-based static timing analysis with uncertainty. In *International Conference on Computer Aided Design (ICCAD-2003)*, pages 607–614. IEEE, November 2003. doi:10.1109/ICCAD.2003.159744.

[DL04] J. L. Díaz and J. M. López. Safe extensions to the stochastic analysis of real-time systems. Technical report, Departamento de Informatica, University of Oviedo, January 2004. http://www.atc.uniovi.es/rsa/starts/documents/tr-04-sesart.pdf.

[DLG+04] J. L. Díaz, J. M. López, M. Garcia, A. M. Campos, K. Kim, and L. Lo Bello. Pessimism in the stochastic analysis of real-time systems: concept and applications. In *25th Real-Time Systems Symposium (RTSS 2004)*, page 197–207. IEEE, December 2004. doi:10.1109/REAL.2004.41.

[Eat17] Eaton. *PHV Supercapacitors Cylindrical pack datasheet.* June 2017. URL https://www.eaton.com/us/en-us/catalog/electronic-components/phv-supercapacitor.specifications.html.

[EDN16] R. Ernst and M. Di Natale. Mixed criticality systems – a history of misconceptions? *IEEE Design & Test*, 33(5):65–74, October 2016. doi:10.1109/MDAT.2016.2594790.

[EY12] P. Ekberg and W. Yi. Outstanding paper award: bounding and shaping the demand of mixed-criticality sporadic tasks. In *24th Euromicro Conference on Real-Time Systems (ECRTS 2012)*, pages 135–144. IEEE, July 2012. doi:10.1109/ECRTS.2012.24.

[EY14] P. Ekberg and W. Yi. Bounding and shaping the demand of generalized mixed-criticality sporadic task systems. *Real-Time Systems*, 50(1):48–86, January 2014. doi:10.1007/s11241-013-9187-z.

[Fok17] T. Foken. *Micrometeorology*. Springer, February 2017. doi:10.1007/978-3-642-25440-6.

[GHL+87] L. Guibas, J. Hershberger, D. Leven, M. Sharir, and R. E. Tarjan. Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. *Algorithmica*, 2(1):209–233, November 1987. doi:10.1007/BF01840360.

[GJKZ19] K. Geissdoerfer, R. Jurdak, B. Kusy, and M. Zimmerling. Getting more out of energy-harvesting systems: Energy management under time-varying utility with PreAct. In *18th International Conference on Information Processing in Sensor Networks (IPSN '19)*, pages 109–120. ACM, April 2019. doi:10.1145/3302506.3310393.

[GSHT13] G. Giannopoulou, N. Stoimenov, P. Huang, and L. Thiele. Scheduling of mixed-criticality applications on resource-sharing multicore systems. In *International Conference on Embedded Software (EMSOFT 2013)*, pages 1–15. IEEE, September 2013. doi:10.1109/EMSOFT.2013.6658595.

[GSM+16] A. Gomez, L. Sigrist, M. Magno, L. Benini, and L. Thiele. Dynamic energy burst scaling for transiently powered systems. In *Design, Automation & Test in Europe (DATE 2016)*, pages 349–354. IEEE, March 2016. doi:10.3929/ethz-b-000116251.

[GSS+17] A. Gomez, L. Sigrist, T. Schalch, L. Benini, and L. Thiele. Efficient, long-term logging of rich data sensors using transient sensor nodes. *ACM Transactions on Embedded Computing Systems (TECS)*, 17(1):4:1–4:23, September 2017. doi:10.1145/3047499.

[GSY15] Z. Guo, L. Santinelli, and K. Yang. EDF schedulability analysis on mixed-criticality systems with permitted failure probability. In *21st International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA 2015)*, pages 187–196. IEEE, August 2015. doi:10.1109/RTCSA.2015.8.

[Hal04] W. Halang. Simplicity considered fundamental to design for predictability. In *Perspectives Workshop: Design of Systems with Predictable Behaviour*. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, September 2004. URL http://drops.dagstuhl.de/opus/volltexte/2004/4.

[HGST14] P. Huang, G. Giannopoulou, N. Stoimenov, and L. Thiele. Service adaptions for mixed-criticality systems. In *19th Asia and South Pacific Design Automation Conference (ASP-DAC 2014)*, pages 125–130. IEEE, January 2014. doi:10.1109/ASPDAC.2014.6742877.

[IE09] M. Ivers and R. Ernst. Probabilistic network loads with dependencies and the effect on queue sojourn times. In *Quality of Service in Heterogeneous Networks (QShine 2009)*, pages 280–296. Springer, November 2009. doi:10.1007/978-3-642-10625-5_18.

[Iqb83] M. Iqbal. *An introduction to solar radiation*. Elsevier, January 1983. doi:10.1016/B978-0-12-373750-2.50001-X.

[ISO18] ISO/TC 22/SC 32. *ISO 26262, Road vehicles – Functional safety*. International Organization for Standardization, December 2018.

[JAD19] N. Jackson, J. Adkins, and P. Dutta. Capacity over capacitance for reliable energy harvesting sensors. In *18th International Conference on Information Processing in Sensor Networks (IPSN 2019)*, pages 193–204. ACM, April 2019. doi:10.1145/3302506.3310400.

[JRR14] H. Jayakumar, A. Raha, and V. Raghunathan. QUICKRECALL: a low overhead HW/SW approach for enabling computations across power cycles in transiently powered computers. In *27th International Conference on VLSI Design and 13th International Conference on Embedded Systems (VLSID 2014)*, pages 330–335. IEEE, January 2014. doi:10.1109/VLSID.2014.63.

[JVT11] J. Jessen, M. Venzke, and V. Turau. Design considerations for a universal smart energy module for energy harvesting in wireless sensor networks. In *9th International Workshop on Intelligent Solutions in Embedded Systems (WISES 2011)*, pages 35–40. IEEE, July 2011. URL https://ieeexplore.ieee.org/document/6086015.

[KH06] W. H. Kwon and S. H. Han. *Receding horizon control: model predictive control for state models*. Springer, June 2006. doi:10.1007/b136204.

[KHZS07] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava. Power management in energy harvesting sensor networks. *ACM Transactions on Embedded Computing Systems (TECS)*, 6(4):32:1–32:38, September 2007. doi:10.1145/1274858.1274870.

[KRHV17] M. Küttler, M. Roitzsch, C.-J. Hamann, and M. Völp. Probabilistic analysis of low-criticality execution. In *Workshop on Mixed Criticality Systems (WMC 2017)*, December 2017. URL https://core.ac.uk/download/pdf/145232877.pdf.

[KTH+16] A. Kurth, A. Tretter, P. A. Hager, S. Sanabria, O. Goksel, L. Thiele, and L. Benin. Mobile ultrasound imaging on heterogeneous multi-core platforms. In *14th Symposium on Embedded Systems For Real-time Multimedia (ESTIMedia 2016)*, pages 9–18. IEEE, October 2016. doi:10.1145/2993452.2993565.

[LBC+17] B. Lucia, V. Balaji, A. Colin, K. Maeng, and E. Ruppel. Intermittent computing: challenges and opportunities. In *2nd Summit on Advances in Programming Languages (SNAPL 2017)*, pages 8:1–8:14. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, May 2017. doi:10.4230/LIPIcs.SNAPL.2017.8.

[LDEG08] J. M. López, J. L. Díaz, J. Entrialgo, and D. García. Stochastic analysis of real-time systems under preemptive priority-driven scheduling. *Real-Time Systems*, 40(2):180–207, April 2008. doi:10.1007/s11241-008-9053-6.

[LK07] F. Li and R. Klette. Rubberband algorithms for solving various 2d or 3d shortest path problems. In *International Conference on Computing: Theory and Applications (ICCTA 2007)*, pages 9–19. IEEE, March 2007. doi:10.1109/ICCTA.2007.113.

[LK11] F. Li and R. Klette. *Euclidean Shortest Paths: Exact or Approximate Algorithms*. Springer, November 2011. doi:10.1007/978-1-4471-2256-2.

[MAH17] G. V. Merrett and B. M. Al-Hashimi. Energy-driven computing: rethinking the design of energy harvesting systems. In *Design, Automation & Test in Europe (DATE*

2017), pages 960–965. IEEE, March 2017. doi:10.23919/DATE.2017.7927130.

[Mas16] A. Masrur. A probabilistic scheduling framework for mixed-criticality systems. In *53nd Annual Design Automation Conference (DAC 2016)*, pages 1–6. ACM, June 2016. doi:10.1145/2897937.2897971.

[MBJ+19] F. Mager, D. Baumann, R. Jacob, L. Thiele, S. Trimpe, and M. Zimmerling. Feedback control goes wireless: guaranteed stability over low-power multi-hop networks. In *10th International Conference on Cyber-Physical Systems (ICCPS 2019)*, pages 97–108, New York, NY, USA, April 2019. ACM. doi:10.1145/3302509.3311046.

[MBS+11] D. Maxim, O. Buffet, L. Santinelli, L. Cucu-Grosjean, and R. I. Davis. Optimal priority assignment algorithms for probabilistic real-time systems. In *19th International Conference on Real-Time Networks and Systems (RTNS 2011)*, pages 129–138, September 2011. URL `https://www-users.cs.york.ac.uk/~robdavis/papers/OPA-RTNS11.pdf`.

[MBTB07] C. Moser, D. Brunelli, L. Thiele, and L. Benini. Real-time scheduling for energy harvesting sensor nodes. *Real-Time Systems*, 37(3):233–260, July 2007. doi:10.1007/s11241-007-9027-0.

[MDCGE17] D. Maxim, R. I. Davis, L. Cucu-Grosjean, and A. Easwaran. Probabilistic analysis for mixed criticality systems using fixed priority preemptive scheduling. In *25th International Conference on Real-Time Networks and Systems (RTNS 2017)*, pages 237–246. ACM, October 2017. doi:10.1145/3139258.3139276.

[MFCP+19] M. Meyer, T. Farei-Campagna, A. Pasztor, R. Da Forno, T. Gsell, J. Faillettaz, A. Vieli, S. Weber, J. Beutel, and L. Thiele. Event-triggered natural hazard monitoring with convolutional neural networks on the edge. In *18th International Conference on Information Processing in Sensor Networks*, pages 73–84. ACM, April 2019. doi:10.1145/3302506.3310390.

[MTBB09] C. Moser, L. Thiele, D. Brunelli, and L. Benini. Adaptive power management for environmentally powered systems. *IEEE Transactions on Computers*, 59(4):478–491, October 2009. doi:10.1109/TC.2009.158.

[NAE+17] A. B. Noel, A. Abdaoui, T. Elfouly, M. H. Ahmed, A. Badawy, and M. S. Shehata. Structural health monitoring using wireless sensor networks: a comprehensive survey. *IEEE Communications Surveys Tutorials*, 19(3):1403–1423, April 2017. doi:10.1109/COMST.2017.2691551.

[NPK+15] S. Naderiparizi, A. N. Parks, Z. Kapetanovic, B. Ransford, and J. R. Smith. WISPCam: a battery-free RFID camera. In *2015 International Conference on RFID (RFID 2015)*, pages 166–173. IEEE, April 2015. doi:10.1109/RFID.2015.7113088.

[PK11] T. Park and S. Kim. Dynamic scheduling algorithm and its schedulability analysis for certifiable dual-criticality systems. In *9th International Conference on Embedded Software (EMSOFT 2011)*, pages 253–262. ACM, October 2011. doi:10.1145/2038642.2038681.

[PSG+14] C. Pagetti, D. Saussié, R. Gratia, E. Noulard, and P. Siron. The ROSACE case study: from Simulink specification to multi/many-core execution. In *20th Real-Time and Embedded Technology and Applications Symposium (RTAS 2014)*, pages 309–318, April 2014. doi:10.1109/RTAS.2014.6926012.

[PSW75] C. Paige, G. P. Styan, and P. G. Wachter. Computation of the stationary distribution of a Markov chain. *Journal of Statistical Computation and Simulation*, 4(3):173–186, November 1975. doi:10.1080/00949657508810122.

[RCM+15] D. Rossi, F. Conti, A. Marongiu, A. Pullini, I. Loi, M. Gautschi, G. Tagliavini, A. Capotondi, P. Flatresse, and L. Benini. PULP: a parallel ultra low power platform for next generation IoT applications. In *27th Hot Chips Symposium (HCS 2015)*, pages 1–39. IEEE, August 2015. doi:10.1109/HOTCHIPS.2015.7477325.

[RE12] RTCA SC-205 and EUROCAE WG-12. *DO-178C, Software considerations in airborne systems and equipment certification.* RTCA, Incorporated, January 2012.

[RMF19] F. Reghenzani, G. Massari, and W. Fornaciari. A probabilistic approach to energy-constrained mixed-criticality systems. In *International Symposium on Low Power Electronics and Design (ISLPED 2019)*, pages 1–6. IEEE, July 2019. doi:10.1109/ISLPED.2019.8824991.

[RPBASR09] J. Recas Piorno, C. Bergonzini, D. Atienza, and T. Simunic Rosing. Prediction and management in energy

harvested wireless sensor nodes. In *1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace and Electronic Systems Technology (Wireless VITAE 2009)*, pages 6–10, May 2009. doi:10.1109/WIRELESSVITAE.2009.5172412.

[RSF11] B. Ransford, J. Sorber, and K. Fu. Mementos: system support for long-running computation on RFID-scale devices. In *16th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS XVI)*, pages 159–170. ACM, March 2011. doi:10.1145/1961295.1950386.

[Saf19] Saft Batteries. *LS 14500 primary Li-SOCl2 cell datasheet*. March 2019. URL `https://www.saftbatteries.com/products-solutions/products/ls-lsh-lsp`.

[San08] Sanyo Semiconductor. *AM-5412 Amorphous Solar Cell datasheet*. July 2008. URL `https://docs.rs-online.com/dd86/0900766b80d10cf7.pdf`.

[SBR+12] L. Schor, I. Bacivarov, D. Rai, H. Yang, S.-H. Kang, and L. Thiele. Scenario-based design flow for mapping streaming applications onto on-chip many-core systems. In *International Conference on Compilers, Architectures and Synthesis for Embedded Systems (CASES 2012)*, pages 71–80. ACM, October 2012. doi:10.1145/2380403.2380422.

[SG15] L. Santinelli and L. George. Probabilities and mixed-criticalities: the probabilistic C-space. In *Workshop on Mixed Criticality Systems (WMC 2015)*, December 2015. URL `https://www-users.cs.york.ac.uk/~robdavis/wmc/6.pdf`.

[SG18] L. Santinelli and Z. Guo. A sensitivity analysis for mixed criticality: trading criticality with computational resource. In *23rd International Conference on Emerging Technologies and Factory Automation (ETFA 2018)*, volume 1, pages 313–320. IEEE, September 2018. doi:10.1109/ETFA.2018.8502493.

[SGG16] L. Santinelli, Z. Guo, and L. George. Fault-aware sensitivity analysis for probabilistic real-time systems. In *International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT 2016)*, pages 69–74. IEEE, September 2016. doi:10.1109/DFT.2016.7684072.

[SGIS10] N. Sharma, J. Gummeson, D. Irwin, and P. Shenoy. Cloudy computing: leveraging weather forecasts in energy harvesting sensor systems. In *7th Annual Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON 2010)*, pages 16:1–16:9. IEEE, June 2010. doi:10.1109/SECON.2010.5508260.

[SGL+17] L. Sigrist, A. Gomez, R. Lim, S. Lippuner, M. Leubin, and L. Thiele. Measurement and validation of energy harvesting IoT devices. In *Design, Automation & Test in Europe (DATE 2017)*, pages 1159–1164. IEEE, March 2017. doi:10.23919/DATE.2017.7927164.

[SGT19] L. Sigrist, A. Gomez, and L. Thiele. Dataset: tracing indoor solar harvesting. In *2nd Workshop on Data Acquisition To Analysis (DATA 2019)*, pages 47–50. ACM, November 2019. doi:10.1145/3359427.3361910.

[Sig20] L. Sigrist. *Design and Instrumentation of Environment-Powered Systems*. Doctoral thesis, ETH Zürich, February 2020. doi:10.3929/ethz-b-000403889.

[SKT17] M. Shirazi, M. Kargahi, and L. Thiele. Resilient scheduling of energy-variable weakly-hard real-time systems. In *25th International Conference on Real-Time Networks and Systems (RTNS '17)*, page 297–306. ACM, October 2017. doi:10.1145/3139258.3139282.

[SSIS11] N. Sharma, P. Sharma, D. Irwin, and P. Shenoy. Predicting solar generation from weather forecasts using machine learning. In *International Conference on Smart Grid Communications (SmartGridComm 2011)*, pages 528–533. IEEE, October 2011. doi:10.1109/SmartGridComm.2011.6102379.

[ST20] N. Stricker and L. Thiele. Analysing and improving robustness of predictive energy harvesting systems. In *8th International Workshop on Energy Harvesting and Energy-Neutral Sensing Systems (ENSsys '20)*, page 64–70. ACM, November 2020. doi:10.1145/3417308.3430264.

[SXL+18] M. Sengupta, Y. Xie, A. Lopez, A. Habte, G. Maclaurin, and J. Shelby. The national solar radiation data base (NSRDB). *Renewable and Sustainable Energy Reviews*, 89:51–60, March 2018. doi:10.1016/j.rser.2018.03.003.

[SZ15] F. K. Shaikh and S. Zeadally. Energy harvesting in wireless sensor networks: a comprehensive review. *Renewable and Sustainable Energy Reviews*, 55:1041–1054, December 2015. doi:10.1016/j.rser.2015.11.010.

[SZDF+15] F. Sutton, M. Zimmerling, R. Da Forno, R. Lim, T. Gsell, G. Giannopoulou, F. Ferrari, J. Beutel, and L. Thiele. Bolt: a stateful processor interconnect. In *13th Conference on Embedded Network Sensor Systems (SenSys 2015)*, pages 267–280. ACM, November 2015. doi:10.1145/2809695.2809706.

[Tex14] Texas Instruments. *TPS6274x 360nA IQ Step Down Converter For Low Power Applications datasheet (Rev. B)*. July 2014. URL `https://www.ti.com/lit/ds/symlink/tps62740.pdf`.

[Tex19a] Texas Instruments. *bq25505 ultra low-power boost charger with battery management and autonomous power multiplexer for primary battery in energy harvester applications datasheet (Rev. F)*. March 2019. URL `https://www.ti.com/lit/ds/symlink/bq25505.pdf`.

[Tex19b] Texas Instruments. *MSP432P401R, MSP432P401M SimpleLink Mixed-Signal Microcontrollers datasheet (Rev. H)*. June 2019. URL `https://www.ti.com/lit/ds/symlink/msp432p401r.pdf`.

[TGBT17] A. Tretter, G. Giannopoulou, M. Baer, and L. Thiele. Minimising access conflicts on shared multi-bank memory. *ACM Transactions on Embedded Computing Systems*, 16(5s):135:1–135:20, September 2017. doi:10.1145/3126535.

[TGTT17] R. Trüb, G. Giannopoulou, A. Tretter, and L. Thiele. Implementation of partitioned mixed-criticality scheduling on a multi-core platform. *ACM Transactions on Embedded Computing Systems*, 16(5s):122:1–122:21, September 2017. doi:10.1145/3126533.

[TJC08] J. Taneja, J. Jeong, and D. Culler. Design, modeling, and capacity planning for micro-solar power sensor networks. In *International Conference on Information Processing in Sensor Networks (IPSN 2008)*, pages 407–418. IEEE, April 2008. doi:10.1109/IPSN.2008.67.

[TSP15] D. Tămaş-Selicean and P. Pop. Design optimization of mixed-criticality real-time embedded systems. *ACM*

Transactions on Embedded Computing Systems (TECS), 14(3):50:1–50:29, April 2015. doi:10.1145/2700103.

[TW04] L. Thiele and R. Wilhelm. Design for timing predictability. *Real-Time Systems*, 28(2):157–177, November 2004. doi:10.1023/B:TIME.0000045316.66276.6e.

[Ves07] S. Vestal. Preemptive scheduling of multi-criticality systems with varying degrees of execution time assurance. In *28th Real-Time Systems Symposium (RTSS 2007)*, pages 239–243. IEEE, December 2007. doi:10.1109/RTSS.2007.47.

[VFPV16] P. Visconti, R. Ferri, M. Pucciarelli, and E. Venere. Development and characterization of a solar based energy harvesting and power management system for a WSN node applied to optimized goods transport and storage. *International Journal on Smart Sensing and Intelligent Systems (S2IS)*, 9(4):1637–1667, December 2016. doi:10.21307/ijssis-2017-933.

[VGB07] C. M. Vigorito, D. Ganesan, and A. G. Barto. Adaptive control of duty cycling in energy-harvesting wireless sensor networks. In *4th Annual Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON 2007)*, pages 21–30. IEEE, June 2007. doi:10.1109/SAHCN.2007.4292814.

[VPN+16] L. Vračar, A. Prijić, D. Nešić, S. Dević, and Z. Prijić. Photovoltaic energy harvesting wireless sensor node for telemetry applications optimized for low illumination levels. *Electronics*, 5(2):26:1–26:16, June 2016. doi:10.3390/electronics5020026.

[Wal13] L. Wald. HelioClim-1: 21-years of daily values in solar radiation in one-click. In *27th International Conference on Informatics for Environmental Protection*, pages 143–148. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, September 2013.

[WBDF+19] S. Weber, J. Beutel, R. Da Forno, A. Geiger, S. Gruber, T. Gsell, A. Hasler, M. Keller, R. Lim, P. Limpach, M. Meyer, I. Talzi, L. Thiele, C. Tschudin, A. Vieli, D. Vonder Mühll, and M. Yücel. A decade of detailed observations (2008–2018) in steep bedrock permafrost at the Matterhorn Hörnligrat (Zermatt, CH). *Earth System Science Data*, 11(3):1203–1237, August 2019. doi:10.5194/essd-11-1203-2019.

[WFM+18] S. Weber, J. Faillettaz, M. Meyer, J. Beutel, and A. Vieli. Acoustic and microseismic characterization in steep bedrock permafrost on Matterhorn (CH). *Journal of Geophysical Research: Earth Surface*, 123(6):1363–1385, May 2018. doi:10.1029/2018JF004615.

[ZS14] T. Zhou and W. Sun. Optimization of battery–supercapacitor hybrid energy storage station in wind/solar generation system. *IEEE Transactions on Sustainable Energy*, 5(2):408–415, January 2014. doi:10.1109/TSTE.2013.2288804.

List of Publications

The following list includes publications that form the basis of this thesis, with references to the corresponding chapter.

R. Ahmed, S. Draskovic, and L. Thiele. **Stochastic guarantees for adaptive energy harvesting systems.** *(In submission).*
. . . . . Chapter 3

S. Draskovic and L. Thiele. **Optimal power management for energy harvesting systems with a backup power source.** In *10th Mediterranean Conference on Embedded Computing (MECO),* pages 1–9. IEEE, June 2021. doi:10.1109/MECO52532.2021.9460139
. . . . . Chapter 2

S. Draskovic, R. Ahmed, P. Huang, and L. Thiele. **Schedulability of probabilistic mixed-criticality systems.** *Real-Time Systems,* 57(4):397–442, Februaruy 2021. doi:10.1007/s11241-021-09365-4
. . . . . Chapter 4

R. Ahmed, B. Buchli, S. Draskovic, L. Sigrist, P. Kumar, and L. Thiele. **Optimal power management with guaranteed minimum energy utilization for solar energy harvesting systems.** *Transactions on Embedded Computing Systems (TECS),* 18(4):30:1–26, June 2019. doi:10.1145/3317679
. . . . . Chapter 2

S. Draskovic, R. Ahmed, C. Lin, and L. Thiele. **A case for atmospheric transmittance: solar energy prediction in wireless sensor nodes.** In *Green Computing and Communications (GreenCom),* pages 427–434. IEEE, July 2018. doi:10.1109/Cybermatics_2018.2018.00097
. . . . . Chapter 2

S. Draskovic, P. Huang, and L. Thiele. **On the safety of mixed-criticality scheduling.** In *Workshop on Mixed-Criticality (WMC)*, pages 19–24, IEEE, November 2016.
https://hal.archives-ouvertes.fr/hal-01438846/
. . . . . Chapter 4

The following list includes publications that were written during the doctoral studies, yet are not part of this thesis.

D. Palossi, A. Gomez, S. Draskovic, A. Marongiu, L. Thiele, and L. Benini. **Extending the lifetime of nano-blimps via dynamic motor control.** *Journal of Signal Processing Systems*, 91(3):339–361, February 2018. doi:https://doi.org/10.1007/s11265-018-1343-1

D. Palossi, A. Gomez, S. Draskovic, K. Keller, L. Benini, and L. Thiele. **Self-sustainability in nano unmanned aerial vehicles: a blimp case study.** In *Computing Frontiers Conference (CF)*, pages 79–88, ACM, May 2017. doi:10.1145/3075564.3075580

Curriculum Vitæ

Personal Data

Name	Stefan Drašković	| Стефан Драшковић
Date of Birth	August 29, 1990	
Citizenship	Serbian	

Education

2016 – 2021 ETH Zürich, Switzerland
Computer Engineering and Networks Laboratory
Doctor of Sciences of ETH Zurich (Dr. sc. ETH Zurich)
under the supervision of Prof. Dr. Lothar Thiele

2013 – 2015 ETH Zürich, Switzerland
Electrical Engineering and Information Technology MSc (MSc ETH EEIT)
Computers and Networks specialization

2009 – 2013 University of Novi Sad, Serbia
Electrical and Computer Engineering BSc

Professional Experience

2016 – 2021 ETH Zürich, Switzerland
Computer Engineering and Networks Laboratory
Research and teaching assistant

2014 – 2015 ETH Zürich, Switzerland
Computer Engineering and Networks Laboratory
Student research assistant (part time)