



Doctoral Thesis

Visual aspects of computer aided control systems design

Author(s):

Ganz, Christopher A.

Publication Date:

1995

Permanent Link:

<https://doi.org/10.3929/ethz-a-001435858> →

Rights / License:

[In Copyright - Non-Commercial Use Permitted](#) →

This page was generated automatically upon download from the [ETH Zurich Research Collection](#). For more information please consult the [Terms of use](#).

Diss. ETH No. 10963

VISUAL ASPECTS OF COMPUTER AIDED CONTROL SYSTEMS DESIGN

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY
ZURICH

for the degree of
Doctor of Technical Sciences

presented by
Christopher Andreas Ganz

Dipl. El.-Ing. ETH
born April 25, 1963
citizen of Meggen and Luzern,
Switzerland

accepted on the recommendation of
Prof. Dr. W. Schaufelberger, examiner
Prof. Dr. J. Gutknecht, co-examiner

1995

Abstract

The use of a computer to solve control engineering problems is state of the art in today's automatic control community. Numerous software packages have been developed in recent years. A few of them are commercially available today and some are even regarded as a de-facto standard in computer aided control systems design (CACSD).

The advantage of CACSD software is the large number and the quality of the provided algorithms. However, developments concentrated on numerical rather than on user interface aspects. Until very recently, most of these packages required textually entered commands and provided rudimentary two-dimensional line graphs.

Recent developments in computer science have removed the limitations of older systems. Data and software visualization are applied to various fields of science and engineering, and most computers can be operated using a graphical user interface.

In this thesis, we explore the application of graphical methods in computer aided control systems design. With increasing emphasis we will discuss the following aspects of visual interfaces:

- Data visualization: Data resulting from either experiments or algorithm execution (*i.e.* computer simulations) are mostly displayed in two dimensional graphs. The use of more complex, higher dimensional displays is discussed. Examples of three dimensional displays, moving data visualization and interactive manipulation of algorithms at runtime are presented.
- Visual programming: The technique to show the interaction of sub-systems in block diagrams is well known in control engineering. Their similarity to visual programming languages helps us to not only display data visually, but also to design computer programs or controller implementations using visual methods. Available visual programming languages are presented in control applications.
- Graphical user interface to the control systems design cycle: As we have seen above, the emphasis of today's CACSD packages is in the availability of sophisticated algorithms. Little or no emphasis has been put on the support of the design cycle as a whole. With the 'Leporello' package we present a new software tool, which visualizes all the work done during a design session. All algorithms which were used to reach a successful controller implementation are stored and displayed in the 'action tree', together with their parameters and results. Intermediate data items are collected consistently in control data objects. The possibility to use external packages for the implementation of algorithms allows a fully graphical user interface for control systems engineering using the whole range of algorithms provided by other applications.

The methods and tools presented are used to solve one of the example plants from the student laboratory. By using the 'Leporello' tool we demonstrate its use and range of application.

Kurzfassung

Die Verwendung von Computern zur Lösung von regelungstechnischen Problemen ist heutzutage Stand der Technik. Mehrere Softwarepakete wurden in letzter Zeit entwickelt. Einige von ihnen sind heute kommerziell erhältlich, und davon haben sich einige zu einem faktischen Standard im Computer-unterstützten Reglerentwurf entwickelt.

Die Vorteile der heute verfügbaren Programmpakete liegen hauptsächlich in der grossen Vielfalt und Qualität der unterstützten Algorithmen. Bei deren Entwicklung wurde jedoch dem Entwurf der Benutzerschnittstelle wenig Beachtung geschenkt. Die Möglichkeiten dieser Pakete erschöpften sich bis vor kurzem in alphanumerischen Befehlseingaben und zweidimensionalen Liniengraphen.

Neuste Entwicklungen im Bereich der Computertechnik haben die technischen Einschränkungen früherer Systeme beseitigt. Daten- und Softwarevisualisierung hat in vielen Gebieten Einzug gehalten, die graphische Benutzerschnittstelle hat sich zur Bedienung von Computern durchgesetzt.

In dieser Arbeit werden die Möglichkeiten der Anwendung graphischer Methoden im Bereich des rechnerunterstützten Reglerentwurfes untersucht. Mit zunehmender Gewichtung werden die folgenden Bereiche diskutiert:

- Datenvisualisierung: Messwerte oder Resultate von Algorithmen (z.B. Simulationsresultate) werden heute meist in zweidimensionalen Graphen dargestellt. Durch die Verwendung höherer Dimensionen kann jedoch anfallenden Datensätzen mehr Information entnommen werden. Dreidimensionale oder animierte Darstellungen sowie die interaktive Beeinflussung von Algorithmen zur Laufzeit werden anhand von Beispielen gezeigt.
- Graphische Programmierung: Die Verwendung von Blockdiagrammen zur Darstellung der Relationen zwischen Teilsystemen ist eine dem Regelungstechniker vertraute Methode. Die Ähnlichkeit dieser Diagramme zu graphischen Programmiersprachen legt die Anwendung solcher Sprachen zum Entwurf von regelungstechnischen Algorithmen und Reglerimplementationen nahe. Verfügbare Umgebungen zur graphischen Programmierung werden jeweils mit Beispielen aus der Regelungstechnik vorgestellt.
- Graphische Benutzerschnittstelle für einen integralen Entwurfsprozess: Wie weiter oben gezeigt, liegt das Schwergewicht bei den heute erhältlichen Software-Paketen auf der Implementation ausgereifter Algorithmen. Wenig oder gar keine Beachtung wird dabei der Unterstützung des gesamten Entwurfszyklus geschenkt. Das in dieser Arbeit vorgestellte Programmpaket 'Leporello' visualisiert diesen Entwurfsprozess. Sämtliche zur Lösung eines Problems verwendeten Algorithmen werden zusammen mit den Algorithmusparametern und den resultierenden Daten gespeichert und im 'Action Tree' (Arbeitsbaum) dargestellt. Anfallende Daten werden jeweils in regelungstechnischen Datenobjekten zusammengefasst und konsis-

tent verwaltet. Die Möglichkeit, von 'Leporello' auf externe Programmpakete zuzugreifen, verwirklicht eine rein graphische Benutzeroberfläche zur Lösung regelungstechnischer Probleme unter Verwendung der Algorithmenvielfalt der externen Pakete.

Anhand eines einfachen Beispiels aus der Regelungstechnik werden die vorgestellten Methoden demonstriert und die Einsatzmöglichkeiten von 'Leporello' anschaulich dargestellt.