TU Clausthal
Clausthal University of Technology

Model-based Engineering for Energy-Efficient Operation of Factory Automation Systems within Unproductive Phases

Doctoral Thesis
(Dissertation)

to be awarded the degree
Doctor rerum naturalium (Dr. rer. nat.)

submitted by
Dipl.-Wirtsch.-Ing. Sebastian Gabriel Mechs
from Schweinfurt

approved by the Faculty of
Mathematics/Computer Science and Mechanical Engineering
Clausthal University of Technology

Date of submission: 2013/05/08

Chief Reviewer: Prof. Dr. Jörg P. Müller
Clausthal University of Technology
Department of Informatics

Reviewer: Prof. Dr.-Ing. Dr. h. c. Peter Göhner
University of Stuttgart
Institute of Automation and Software Engineering

Abstract

In the face of a future rise in energy prices, energy-efficient operation of industrial automation systems has strategic impact for manufacturing companies. The reduction of energy demand during unproductive phases helps to contribute to the overall energy efficiency of automated production systems.
Up to now, there is no general scientific concept which addresses energy-efficient operation of factory automation systems within unproductive phases technically and economically on a multi-subsystem level. However, proposing detailed instructions and strategies for multiple interacting subsystems is crucial in order to realize energy savings technically.
On this account, the proposed automaton-based system model enables the analytical description of structural and behavioral aspects of industrial automation systems. This kind of mathematical modeling serves as basis for identifying optimal strategies analytically relying on a structure-exploiting procedure which enables efficient strategy computation. Those strategies quantify the energy savings potentials and give support for technical realization.
Since the computation of optimal strategies for industrial automation systems is complex, a novel approach is developed to calculate those strategies efficiently incorporating the problem structure provided by the model. Using models of real-world automation systems, the approach of this thesis is evaluated regarding further objectives. First, the feasibility of strategy execution is ensured which enables the evaluation of design decisions. Computed strategies are verified in the target system regarding correct execution. The prediction of energy demands by strategies is sensitive to model-to-system deviations, so that tests are applied to check the system model for accuracy of predictions. Economic considerations complete the assessment of the approach.
Using the general concepts and methods of this thesis, the energy demand for industrial automation systems can be substantially reduced within unproductive phases. The chosen approach supports the model generation, the computation and evaluation of strategies, and the technical realization for industrial automation systems.

Keywords:
Energy efficiency, energy-optimal, industrial automation, unproductive phase, model-based engineering, priced timed automaton, networked automata, symbolic reachability analysis, constraint optimization problem, combinatorial optimization, optimal strategy, feasible strategy

Zusammenfassung

Zukünftig steigende Energiepreise stellen die automatisierte, industrielle Produktion vor die Herausforderung, benötigte Energie effizient einzusetzen. Die Reduzierung des Energiebedarfs in Nicht-Produktivphasen ermöglicht dabei einen wesentlichen Beitrag zur Gesamtenergieeffizienz von automatisierten Produktionssystemen zu leisten.
Forschungsansätze liefern bisher keine analytischen Ansätze zur Berechnung von detaillierten Strategien, um das Energieeinsparpotenzial von Nicht-Produktivphasen in modularen Automatisierungssystemen einzuschätzen. Es werden jedoch detaillierte Anweisungen und Strategien für interagierende Subsysteme benötigt, um Energieeinsparungen technisch realisieren zu können.
Die vorliegende Arbeit bietet daher ein automatenbasiertes Systemmodell zur Beschreibung von strukturellen und verhaltensspezifischen Aspekten von Automatisierungssystemen an. Dieses Modell dient als formale Basis zur Entwicklung eines Strategieoptimierungsmodells. Strategien liefern neben der Quantifizierung des Energieeinsparpotenzials eine Spezifikation zur Ausführung im Zielsystem.
Da die Berechnung einer optimalen Strategie für industrielle Automatisierungsanlagen komplex ist, kann auf Basis der gewählten strukturellen Problembeschreibung ein Berechnungsverfahren vorgeschlagen werden, um optimale Strategien zielgerichtet zu berechnen. Anhand von Modellen realer Fertigungsautomatisierungssysteme wird der Ansatz dieser Arbeit nach weiteren praxisrelevanten Fragestellungen evaluiert. Zum Einen muss bereits zur Designzeit des Automatisierungssystems hardwarenah die Ausführbarkeit von Strategien sichergestellt werden, um Designentscheidungen und deren Auswirkungen zu bewerten. Zum Anderen unterliegt die Vorhersage des Einsparpotenzials aufgrund von Unterschieden zwischen Modell und System einer bestimmten Abweichung. Mittels Tests wird die Auswirkung auf die Aussage des Einsparpotenzials untersucht. Eine Schlussbetrachtung zeigt das ökonomische Energieeinsparpotenzial auf, das mit dem in dieser Arbeit vorgestellten, modellbasierten Ansatz realisiert werden kann.
Der generische Ansatz dieser Arbeit erlaubt den Energiebedarf von industriellen Automatisierungssystemen in Nicht-Produktivphasen in beträchtlichem Maße zu reduzieren. Dabei wird sowohl die Modellerstellung sowie die Strategieberechnung und Strategiebewertung als auch die technische Umsetzung unterstützt.

Schlagwörter:
Energieeffizienz, energieoptimal, industrielle Automatisierung, Nicht-Produktivphase, modellbasierte Planung, zeit- und kostenattributierter Automat, Automatennetzwerk, zeitliche Erreichbarkeitsanalyse, kombinatorisches Optimierungsproblem, optimale Betriebs- und Schaltstrategien, Strategierealisierung

Acknowledgment

As in every challenging project that creates something new, this doctoral thesis would not have been possible without the inspiring advice of many parties. The outcomes of this thesis are in many ways based on interdisciplinary work during my doctoral period at Siemens Corporate Technology in Munich and Clausthal University of Technology from 2010 to 2013. The detailed work in the field of energy efficiency in industrial automation has been enabled by my former supervisor Mr. Volker Albrecht who established the contact to Mr. Frank Konopka responsible for researching energy efficiency aspects in industrial automation that time. Mr. Konopka provided the confidence on which I based my work resulting in this thesis. Additionally, Mr. Jörn Peschke, Mr. Rainer Förtsch, and Mr. Patrick Volkmann enabled the discussion of my concepts and ideas benefiting from their profound and comprehensive knowledge of energetical aspects in industrial automation systems. The discussions resulted in a detailed screening of my work and perpetually reminded me of the technical feasibility. I have obtained a deep understanding of the problem context by these lively debates over weeks and months.
Besides discussing the technical requirements, Dr. Steffen Lamparter and Dr. Stephan Grimm provided advice to improve conceptual and scientific aspects. My doctoral thesis adviser Prof. Dr. Jörg P. Müller backed my scientific approach in an open and cordial way from the computer science point of view. His perception of the problem context and his scientific experience have considerably contributed to the scientific solidity of this thesis. I also express gratitude to Prof. Dr.-Ing. Dr. h. c. Peter Göhner for reviewing the approach, the contents, and results of this doctoral thesis based on his expertise with automation systems.
To the same degree as the technical and scientific parties made possible this work, Ms. Kathrin Kiefmann has contributed essentially to this thesis with her permanent encouragement and incontrovertible affection.

Sebastian Mechs, Munich, May 2013

Contents

Acronyms

BB	Bounded investigation of the set of strategies
CE	Complete enumeration of the set of strategies
CNC	Computer Numerical Control
COP	Constraint Optimization Problem
CSP	Constraint Satisfaction Problem
CU	Control Unit
DB	Data Block
DBM	Difference Bound Matrix
DES	Discrete-Event System
DEVS	Discrete Event System Specification
DPM	Dynamic Power Management
EM	Energy Management
EMS	Energy Management System
ERP	Enterprise Resource Planning
FB	Function Block
FMS	Flexible Manufacturing System
I/O	Input/Output
ICT	Information and Communication Technology
IEC	International Electrotechnical Commission
ILP	Integer Linear Programming
IPC	Industrial Personal Computer
MES	Manufacturing Execution System

NC	Numerical Control
PCS	Process Control System
PI	Profibus International
PLC	Programmable Logic Controller
PPC	Production Planning and Control
SCL	Structured Control Language
SP	Complete enumeration of solutions without the use of strategies
SPN	Stochastic Petri Net
TA	Timed Automaton
TBA	Timed (Büchi) Automaton
TPN	Time Petri Net
UML	Unified Modeling Language

Symbols and variables

$\mathrm{Sgn}^{\mathrm{in}}$	Set of input signals
$\mathrm{Sgn}^{\mathrm{out}}$	Set of output signals
Σ	Set of timed events
V	Set of interval variables
SV	Set of shared variables
St	Set of states
St^{symb}	Set of symbolic states
T	Set of time points, time sequence
L	Set of related strategies
$L_{unrelated}$	Set of unrelated strategies
Sub	Set of subsystems
Seq^{i}	Set of alternative switching sequences of a Subsystem sub_i
Z	Set of zones
$\mathrm{Z}^{\uparrow}$	Projection
$\{\mathrm{r}\}\mathrm{Z}$	Reset
mod	Number of modes in a subsystem
tra	Number of transitions in a subsystem
$\mathrm{dev}_{\mathrm{energy}}$	Model-to-system overestimation/underestimation of energy demand [%]
$\mathrm{dev}_{\mathrm{power}}$	Average model-to-system deviation regarding input power [%]
$\mathrm{dev}_{\mathrm{delay}}$	Average model-to-system deviation regarding mode delays [%]
$\mathrm{eni}_{\mathrm{idl}}$	Energy input without strategy (system idling)

eni_{str}	Energy input with applied strategy
ens^{abs}_{pau}	Absolute energy savings within a pause interval = eni_{idl} - eni_{str}
ens^{rel}_{pau}	Relative energy savings within a pause interval [%]
pc	Input power
sv_i	Shared variable sv of subsystem sub_i
sd_{ik}	Subsystem dependency between Subsystem sub_i and Subsystem sub_k
sub_i	Subsystem i
sys	System
obj_{energy}	Objective function *energy minimizing*
obj_{time}	Objective function *time minimizing*
$v_i(seq^j_k)$	Interval variable i of a switching sequence seq_k in Subsystem sub_j
mem	Maximum computational memory consumption [MB]
run	Computational runtime [s]
ev_i	Event i
sgn^{in}	Input signal
sgn^{out}	Output signal
st_k	State k
st^{symb}_k	Symbolic state k
deg	Average number of subsystem dependencies
clu	Clustering coefficient
dep	Dependency density
m^i_k	Mode k in Subsystem sub_i
occ	Occurrence
$e(l_p)$	Minimum energy demand of a related strategy p
l_p	Related strategy p
l_{opt}	Energy-optimal, related strategy
$e(l_{p,unrel})$	Minimum energy demand of an unrelated strategy p

$l_{p,unrel}$	Unrelated strategy p
tb_M	Test bed for simulation-based experiments
tb_S	Test bed for direct experiments
d	Delay time in a mode
t_i	Time point i
α	Production mode of a subsystem
β	Ready-for-production mode of a subsystem
δ	Synchronization mode of a subsystem
ε	Off mode of a subsystem
γ_i	Standby mode i of a subsystem
$\zeta^{(n)}$	n subsystems in mode ζ
σ_i	Timed event i
$\bullet$	Planned strategy is feasible
$\circ$	Planned strategy is not feasible

List of Figures

List of Tables

List of Definitions

Part I

Energy efficiency of factory automation systems

Chapter 1

Introduction

Germany's plans for renewables and its energy transformation have been attracting worldwide attention [NYT, 2012], [Wettmann, 2012], [Buchan, 2012]. The energy policy of Germany's federal government intends to close all nuclear power plants by 2022 [FGG, 2012]. The required energy demand needs to be covered by conventional power plants respectively renewables by then [Flauger, 2012], [BMU, 2012]. This represents a challenge on future energy supply and energy demand.
The *World Energy Council* qualified accessible, industrial energy as a limited resource in 2010. The bounded availability of accessible energy is a threat to economic and wealth and growth [Gadonneix, 2010]. In fact, an increased world energy consumption and demand contributes in conjunction with limited accessible energy supply to a rise in energy prices. This development is reflected in the perpetual rise of oil prices [Conti, 2011]. The political phaseout decision intensifies the problem of affordable energy supply for industry in the future.
Germany's focus on energy-efficient industrial production [Laub, 2013] indicates the relevance of considering energy as limited resource [Schnellnhuber, 2012]. There exists a common agreement about the careful handling of limited resources. The ongoing discussion is a stimulator for rethinking human interaction with the environment. Many research projects have been started in order to investigate domestic home's and industry's energy efficiency. Starting with a first national research program for alternative energy technologies in 1977 caused by the first oil crisis in 1973 [Wagner, 2010], the 6th national program was initiated in August 2011 in Germany [BMBF, 2011]. Multiple research projects like *Green Carbody Technologies* [GCT, 2010], *Resource Conservation by Context-Activated M2M-Communication* [ResCom, 2012], and the project *FOR1088'ECOMATION'* [*FOR1088'ECOMATION'*] illustrate an increased need for giving answers to increasing energy prices. As a result, Germany is the leading nation in Europe in terms of reducing primary energy demand [Agricola, 2012, page 8].
Nevertheless, the current accomplishments give no reason to cease actions [McKinsey, 2010, page 30], [IEA, 2011], [IEA, 2012c]. In these days, industry is the player in the market with the biggest energy consumption in Germany [IEA, 2007, page 19]. It is predicted that this sig-

nificance will grow until 2030 and later on. Consequently, the integration of energy-relevant aspects in the planning and engineering aspects of automated production processes in industry can be seen as the essential competitive factor for the future [Berger, 2009], [Berger, 2010, page 9], [Agora Energiewende, 2013]. Energy efficiency gets a strategic dimension [McKinsey, 2009].

Consequently, the energetical aspects of automated production need to be considered during engineering of manufacturing processes in order to completely benefit from an energetical evaluation in early design phases of production sites [Müller, 2009]. In industry, besides "increased demand" and "replacing aging equipment", buyers and sellers of automation technology name "improving energy efficiency" as major reasons for investments [Greenfield, 2009, page 19]. Today, the energetical perspective on production sites is taken while the system is already set up [Abele, 2010]. Basic design decisions can hardly be retracted in production plants. Planners need a framework on which they can make fundamental design decisions within industrial manufacturing taking energy aspects into account.

One of those important design decisions is linked to unproductive phases of automated production. Up to now, unproductive phases of automated production systems are not adequately covered by state-of-the-art solutions in industry and science. This thesis clarifies the problem context of unproductive phases and related challenges in industrial automation and offers models, methods and evaluation to address and technically use unproductive phases.

Introducing this problem context, the political, economic and technical developments in Germany in the recent years are presented (Section 1.1). The central problem statement is exemplified by industrial examples (Subsection 1.2.1) and requirements are derived from that (Subsection 1.2.2). An overview of the scientific contribution to identified research objectives addressing unproductive phases is shown in Section 1.3. The outline of this thesis (Section 1.4) is dominated by the motivation to reduce the energy demand during unproductive phases in automated production systems.

1.1 Motivation in energy management

Energy is a key resource and quantifier for efficiency in manufacturing [IEA, 2012b, page 4]. Political and economic (Subsection 1.1.1) as well as technical (Subsection 1.1.2) challenges have an effect on decisions related to energetical considerations in industrial production.

1.1.1 Political and economic challenges

Electrical energy is, besides thermal energy, the most important and universal form of ability to perform work in industrial processes [Elefterie, 2012]. The directive of the European Parliament regarding energy efficiency [EU-Directive, 2012] passed in September 2012 requests industry to accomplish energy savings of 1,5% per year between 2014 and 2020. This affects each source of energy to the same extent and increases the pressure to reduce the energy input to industrial production. Additionally, a rise in energy prices can be experienced (Fig. 1.1). This increase [Kessler, 2008, page 7], [Kögler, 2012, page 6] influences the overall costs of companies significantly depending on the energy intensity of production ranging between 8% and 63% of the gross value added [BMU, 2011, pages 4, 5].

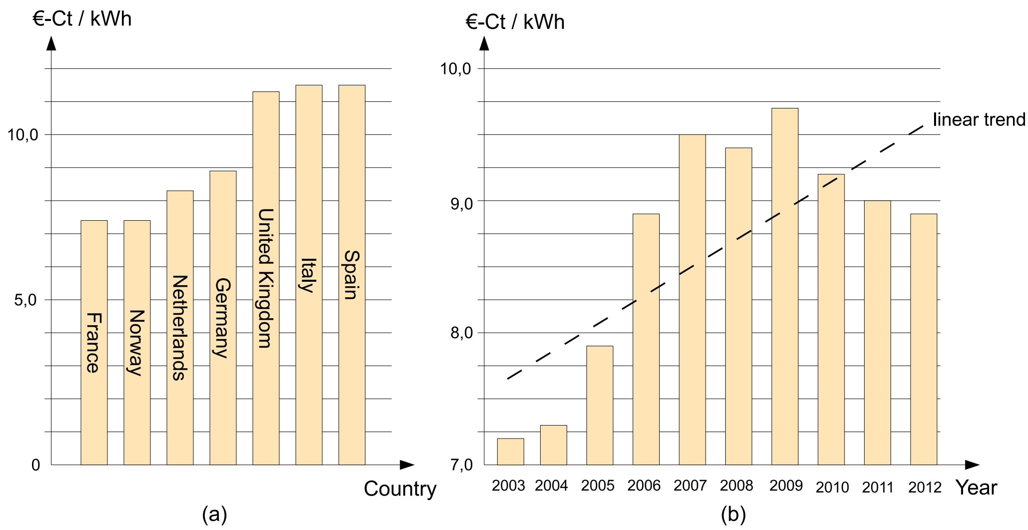


Figure 1.1: (a) National average prices without taxes for electrical energy in 2012 for middle-sized industrial companies (annual energy demand between 500 and 2.000 MWh), (b) Germany's average prices (annually) without taxes for electrical energy for middle-sized industrial companies (annual energy demand between 500 and 2.000 MWh), [*Eurostat database*]

In Figure 1.1, a snapshot of Europe's and Germany's industrial energy prices (Figure 1.1 *(a)*) are shown with a positive linear trend for middle-sized industrial companies between 2003 and 2012 (Figure 1.1 *(b)*). Although there is a slight decrease of energy prices between 2009 and 2012, among experts there is a common understanding about the influence of Germany's transformation in energy supply from nuclear power to renewable energy sources on energy prices. If the nuclear phaseout is realized in Germany, *McKinsey* company has predicted costs of 21,5 Billion Euros per year at the start of 2020 [Krägenow, 2012, page 5] which accelerates the rise in energy prices. Law and politics strongly influence the national energy prices [Brost, 2012, pages 17-19]. These developments force industrial companies to increase their energy efficiency and to partially decouple energy demand and production. Manufacturing is a crucial lever being applied for energy reductions. However, organizational and technical challenges have to be faced.

1.1.2 Organizational and technical challenges

In technical processes, electrical units provide access to evaluate the energy demand of manufacturing systems. Unless otherwise indicated, electrical energy is in the focus of this thesis. By means of electrical units, the demand for operating resources like compressed air, vapor pressure and laser energy can be expressed [IEA, 2004]. The complement of productive operation in industrial settings has been seen as necessary evil for long while: *unproductive operation and machine idling*. This idling operation is caused by change over times of machines, partial breakdowns, planned production breaks, and short-term machine idling. The analysis of idling times and the deduction of action plans bear a significant energy savings potential. In the past years, it has been realized that idling of machines and production equipment causes economic costs. This has resulted in a continuing progress in technical systems for energy management making energy flows transparent [Duflou, 2012] whilst the aspect of idling is not addressed adequately in present Energy Management (EM) implementations [Niemann, 2012].
The definition of EM (Def. 1) and Energy Management System (EMS) (Def. 2) give no indication how to deal with unproductive phases in industrial manufacturing.

Definition 1 (Energy management)
EM comprises the total of planed and executed actions in order to ensure a minimum of energy input for a predefined performance [DIN-EN-ISO-50001, 2011] (prior: DIN EN 16001).

This definition is interpreted differently on distinct levels of an organization. Since *DIN EN ISO 50001* resides on a conceptual level, it does not provide advice for implementations. The guideline [VDI-4602, 2007] gives an additional definition of energy management systems (Def. 2).

Definition 2 (Energy management system)
An EMS is a "set of interrelated or interacting elements to establish an energy policy and energy objectives, and processes and procedures to achieve those objectives" [DIN-EN-ISO-50001, 2011, page 2]. The task of an energy management system "is to arrange the production factor of energy independently alongside the production factors of capital, work and material to be able to monitor the use and the costs (efficiency of using energy) of energy in the company" [VDI-4602, 2007].

These definitions have to be interpreted and realized on an organizational and technical level. It has to be stated that both, energy management and energy management systems, need to be substantiated [IEA, 2012a].
Research tries to concretize the conceptual Definitions 1 and 2. A study has been commissioned by ZVEI[1] in order to render today's EMS implementations. The study comprises the investigation of capabilities and functions of EMS implementations [Niemann, 2012]. Today's EM concepts strongly focus on accurate assignment of energy costs to entities in a company and the

[1]Zentralverband Elektrotechnik- und Elektroindustrie e. V.

transparent visualization of energy flows on production level. Among others, the most important characteristics comprise data acquisition, data visualization, archiving and the prognosis of future energy demand [Kahlenborn, 2012]. The acquisition is realized in real-time while the aggregation of data is supported. EMS provides data acquisition and data analysis functions covering energy demand and prognosis. If energy demand is recorded, then companies often monitor the demand for compressed air and other energy sources. After having collected the energy data, the analysis of the data is often conducted manually [Knafla, 2010]. The study also reveals that plant operators request propositions for shutdowns. Automated shutdown is not supported by EMS whereas the requirement for shutdowns is recognized by the users. Furthermore, users of EMS request tools for modeling energy-related information during the engineering of a plant. In summary, the study expresses two facts: energy data acquisition, visualization, and analysis have reached maturity whereas support for evaluating and reducing energy demands within unproductive phases is missing.
Concentrating on unproductive phases must be of substantial benefit. Therefore, the central problem of this thesis is put in concrete terms next.

1.2 Problem statement

Energy efficiency is expected to contribute essentially to affordable energy supply and security of energy supply in the context of depletion of resources [BMWi, 2011]. Energy efficiency is a major focus of automation technology as a survey of VDI/VDE reveals [Westerkamp, 2009, page 17]. Since energy efficiency is a central issue, the understanding of this term has to be clarified. Efficiency is the extent of effort dedicated for a specific purpose. Depending on the field of application, it is the ratio of utility to input [Müller, 2009]. In the context of energy efficiency, the definition of [DIN-EN-ISO-50001, 2011] is adopted (Def. 3).

Definition 3 (Energy efficiency)
Energy efficiency is the ratio or other quantitative relationship between an output of performance, service, goods or energy, and an input of energy.

In this thesis, energy efficiency is discussed on subsystem level focusing on the idling phases of discrete manufacturing systems. Subsystems can consist of machines, components of machines, and automation infrastructure. Today, unproductive phases in manufacturing result in subsystem idling which is inefficient, in general. Inefficient means that there exists no measurable gain in performance while subsystems are idle and require input of energy. To clarify the extent of idling, Figure 1.2 shows the accumulated idling times during five-day car production [Hübner, 2011]. The study reveals the idling phases of production lines at *Daimler AG* and *Volkswagen AG*, two of the largest car manufacturers worldwide. 64 percent of the pauses during production for the examined production lines are at least three minutes. A 33% energy savings potential through complete production shutdown strategies is identified assuming two shifts and five working days per week. Today, short-term and mid-term idling phases as shown in Figure 1.2 are not subject to energy reducing actions. Automated manufacturing systems are left in idling which requires a considerable amount of energy.

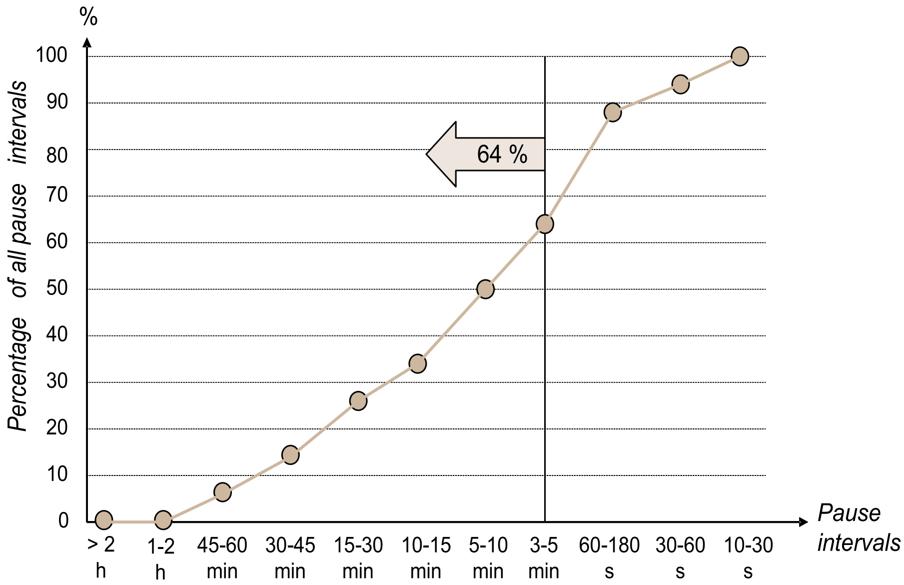


Figure 1.2: Accumulated idling times during five-day production in [Hübner, 2011, page 2]

On machine tool level, [Dietmair, 2009b, page 63] analyzes the transformation of input power. The operating modes of a machine tool and the attached power input for each mode are identified. It is shown that a 20% share of input power is used for idling. A 20% energy savings potential within unproductive phases is confirmed by [Kulus, 2011, page 588]. Noting that the power input strongly differs depending on the manufacturing process, [Neugebauer, 2008] identifies the basic input power of inactive machine tools at about 30% of the full load input power.

Today, there are technical means to shut down individual, electrical energy consumers with input power up to 100 Mega Watt. Entities with high input power appear especially in metal working, chemical, paper and cement industries. The extent of industry's overall input power that can be used for energy demand reductions is numbered at about 6,5 Giga Watt in Germany [Schermer, 2012]. For instance, energy input can be reduced by using unproductive phases of heating processes [Müller, 2012, page 18].

In order to give an impression of the energy savings potential within unproductive phases in industrial manufacturing, two common examples are presented. The first application presents a subsystem of mixed process and factory automation in the field of synthetics industry. The second example shows the facet of machine tool idling.

1.2.1 Industrial examples

Example 1: Idling in injection die casting

Measurements have revealed that injection die casting machines still have a 30% input power compared to the input power during full load operation [Hornberger, 2009]. In a production setup at *Kärcher* company, which develops and produces cleaning systems like high-pressure cleaners, machines require approximately 20% percent of electrical energy an 10% of thermal energy in unproductive phases compared to productive phases (Fig. 1.3) [Neher, 2009].

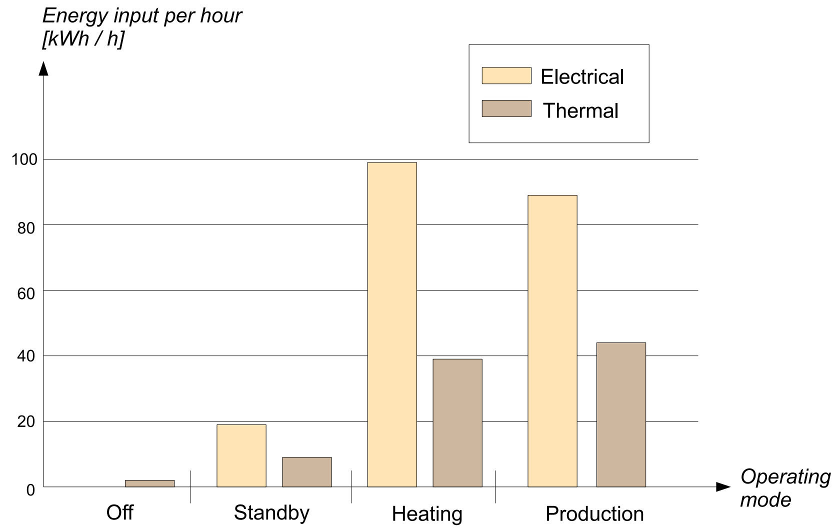


Figure 1.3: Injection die casting, energy input per hour [kWh/h] per mode in [Neher, 2009, page 87]

Figure 1.4 illustrates that even during standby (idling) of the system, the energy input increases substantially.

Example 2: Idling of machine tools

Similar to the example mentioned before, energy savings potentials in factory automation exist especially on machine level. The productive time of machine tools is estimated at 15% of the overall process time [Neugebauer, 2008]. The rest of the time is spent for setup times and other unproductive times. Even within large-scale production, machine tools have a productive time about 40%. Energy savings potentials are estimated at 10% to 15%.
In [Santos, 2011], the share of energy input to different stages of the life cycle of a machine tool is analyzed. It is stated that the investigated types of machine tools still have a standby input power between 7% and 30% of the full load input power. Machines with high standby input power are turned off manually after a fixed time interval of ten minutes in the examined setting

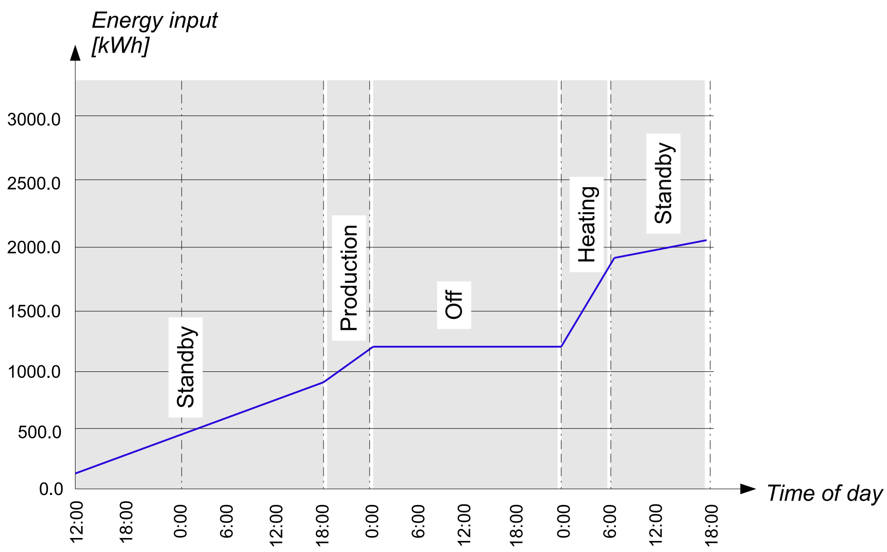


Figure 1.4: Injection die casting, accumulated energy input [kWh] in [Neher, 2009, page 87]

[Santos, 2011, page 5]. In this context, control interfaces to single machine tools can support the exploitation of energy savings potentials within unproductive phases [SiemensAG, 2011]. These introductory examples exemplify today's necessity for regarding unproductive phases as important lever to reduce energy demand. The exploitation of energy savings for single machines is manageable with state-of-the-art solutions. However, in industrial settings, multiple machines interact with each other in a complex production process. Managing automation systems with many interacting subsystems needs to respect interdependencies. This challenge and other industry's requirements to be met for using unproductive phases are presented next.

1.2.2 Requirements for energetically exploiting unproductive phases

To use unproductive phases of automated production systems for energy reductions, several requirements have to be met.

Need for modeling systems regarding energetical aspects

Systems in industrial automation show a high process and system complexity arising from subsystem interactions. This system complexity has to be taken into account in concepts addressing unproductive phases. Actions taken at one component can have severe implications for the complete system. Consequently, relations and dependencies between subsystems need to be respected proposing an intelligent usage of pause intervals. A plant operator needs analytical support to assess on unproductive phases and to be able to handle the complexity of an

automation system consisting of many subsystems. “Concrete models for energy consumption prediction at each layer e.g. device level, location, process level etc. are needed“ [EC, 2009, page 37].
The internal behavior of subsystems can be complex as well. For instance, machine tools are highly integrated subsystems which can exhibit various temporal and energetical properties. The insights of a machine need to be specified and explicitly modeled to know the way a machine can be used in order to increase energy efficiency for unproductive phases.

Need for energy-centric implementations

Frameworks on conceptual and implementation level are required which enable to evaluate the energy saving potentials for subsystem idling. A general research objective is identified: “An effective energy control system has to be developed, using the information of sensors and in-process measurement and a suitable energy efficiency performance measuring system. This control system focuses on concepts that facilitates the evaluation, control, and improvement of energy efficiency in manufacturing processes“ [Bunse, 2010, page 674].

Need for intelligent usage of pause intervals

Fraunhofer and the *Federal Ministry of Education and Research* in Germany state that further research is required towards using energy information on a subsystem level in manufacturing. The structured access of information regarding pause intervals is essential to drive components into power saving modes. For this purpose an intelligent control application taking care of pause intervals, starts and ends of those intervals and the execution of switching commands is required [Neugebauer, 2008, page 8]. The *European Commission* states that intelligent control in manufacturing can effectively increase energy efficiency. Selective switch off allows energy reductions to be realized. It is stated that the “detection of beginning and end of down times, intelligent monitoring, system diagnosis and auto-correction should be implemented“ [EC, 2009, page 20]. A tool support that can evaluate idling periods and can provide information about actions to be taken to reduce energy input within unproductive phases, can make a substantial contribution to reductions of energy demand.

Need for a quantification of achievable energy savings

The question to be answered for achieving energy demand reductions within unproductive phases is the following. When does it pay off to guide the system to energy-efficient modes and in which way should it be technically realized? Assuming different pause intervals, it is interesting to know which energy savings potential is linked to the pause intervals. To make statements about expected energy inputs and savings of the system, the internal behavior of subsystems and subsystem dependencies must be noted and taken into account. Evaluation of energy savings requires to compare different alternative measures and to identify the best alternative. This thesis addresses these requirements in the scope of its scientific contribution.

1.3 Research objectives and scientific contribution

From the requirements, research objectives are derived. The scientific contribution of this thesis is clarified with reference to these objectives (Fig. 1.5).

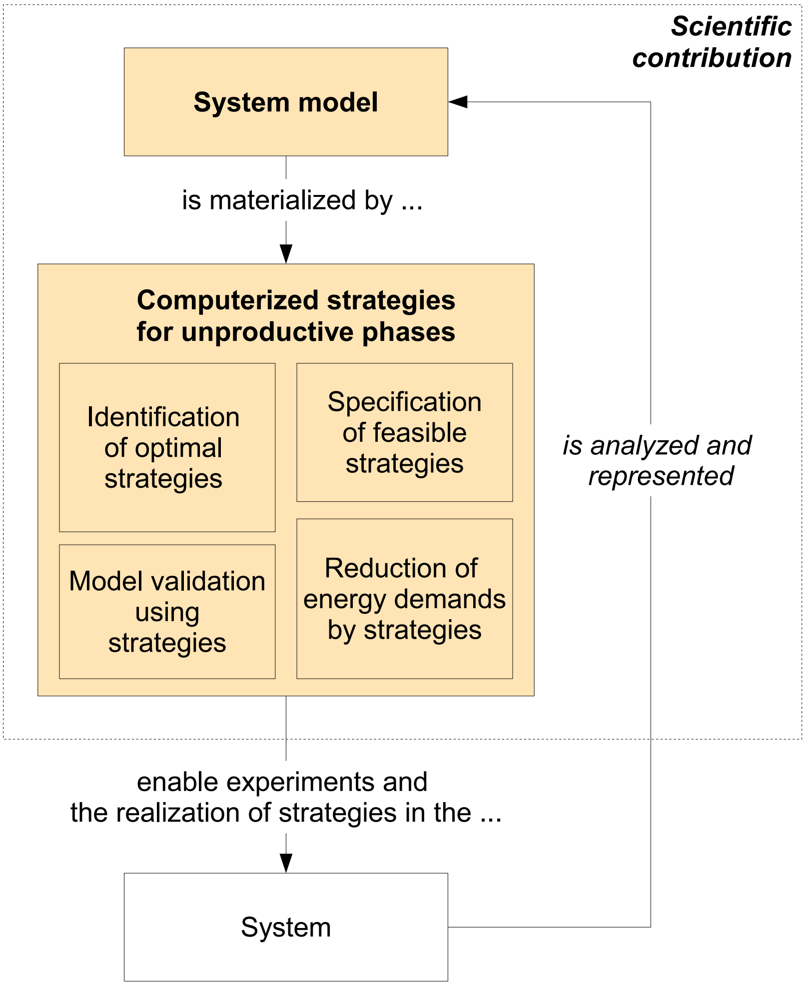


Figure 1.5: Scientific contribution of the approach of the thesis

1.3.1 Analytical system model

In order to use energy savings potentials within unproductive phases, a model incorporating relevant information needs to be set up as basis for energetical considerations and predictions. Model-based engineering of manufacturing subsystems provides means to do so [BMWi, 2010, page 19]. The structural and behavioral aspects of automated production systems need to be reflected in a system model for analytical purposes.

> ***Contribution 1***
> *An analytical system model for describing the structural and behavioral aspects of automated production systems is introduced in Chapter 4. This generic model serves as information basis for evaluating the energy savings potentials in the system.*

1.3.2 Computerized strategies for unproductive phases

To achieve the objective for lower energy input during unproductive phases, strategies must be specified which enable the quantification of energy savings potentials. In this thesis, the term strategy refers to the application of an executable plan in order to reduce the energy input during unproductiveness. In [Mittelbach, 2010, page 47], it is stated that strategies are effective means to face inefficient idling phases of industrial production.

> ***Contribution 2***
> *Based on the structural and behavioral information of systems contained in the analytical system model, strategies are derived for unproductive phases in Chapter 5.*

Identification of optimal strategies

The computation of the optimal strategy based on an analytical model is a major obstacle because of computational runtime performance. Therefore, a tractable procedure needs to be proposed which supports the identification of optimal strategies for industrial systems.

> ***Contribution 3***
> *A tractable procedure is proposed which helps to find optimal strategies for practical problems using the structure of the system model in Chapter 6. The efficiency of the procedure is evaluated in Section 9.1.*

Specification of feasible strategies

Credibility regarding the feasibility of strategies has to be provided. It needs to be checked if the strategy specification is correct and the system model does not contain errors preventing the execution of strategies. Expectable deviations between system model and system influence the feasibility of strategies. Therefore, support has to be given to check the execution of strategies towards model-to-system deviations.

Contribution 4
A framework presented in Chapter 7 provides means to check the feasibility of strategies. The correct representation of the structure of the system and the internal behavior of the system is analyzed based on strategies. The results of verification tests regarding correct implementation of strategies are presented in Section 9.2.

Model validation using strategies

Computed strategies need to be realized in the system in order to benefit from energy savings. The strategy realization should be used to evaluate the accuracy of the system model. This gives indication about the validity and the need for calibration regarding the system model. Model-to-system deviations determine the validity of the strategy-based prediction of the energy demand. A decision regarding tolerable deviations can be made based on this information. The measuring effort as well as the model details can be managed based on these considerations.

Contribution 5
A framework is provided in Chapter 7 to realize computed strategies in the system. In form of validation tests, the validity of a system model is checked in Section 9.3. The influencing factors on the sensitivity of the strategy-based prediction regarding the energy demand are analyzed.

Reduction of energy demands by strategies

The economic benefit of strategies for unproductive phases needs to be quantified. The objective is to minimize the energy demand (input) for a given unproductive period. Utilizable pause intervals need to be identified and economically evaluated (Fig. 1.2). This gives decision support for the selection of strategies in certain pause intervals.

Based on Definition 3, the economic objective of this thesis is detailed. The energy efficiency of an automation system during unproductiveness without strategy is denoted by $\mathrm{eff}_{\mathrm{idl}}$ (Equ. 1.1). The efficiency is determined by the ratio of *output* to eni_{idl}. The variable *output* can be the availability of the automation system, for instance. In a second case, the same automation system with $\mathrm{eff}_{\mathrm{str}}$ is assumed to use a strategy which reduces the energy input to eni_{str} while the *output* stays the same (Equ. 1.2).

$$\mathrm{eff}_{\mathrm{idl}} = \frac{\mathrm{output}}{\mathrm{eni}_{\mathrm{idl}}} \tag{1.1}$$

$$\mathrm{eff}_{\mathrm{str}} = \frac{\mathrm{output}}{\mathrm{eni}_{\mathrm{str}}} \tag{1.2}$$

Consequently, if $\mathrm{eni}_{\mathrm{str}}$ within unproductive phases is smaller than $\mathrm{eni}_{\mathrm{idl}}$, it can be stated that:

$$\mathrm{eff}_{\mathrm{str}} > \mathrm{eff}_{\mathrm{idl}} \tag{1.3}$$

On this account, it has to be shown that strategies comply with Equation 1.3. This means that it must be argued that using strategies during unproductive phases reduces the energy input of an automation system.

> ***Contribution 6***
> *The economic benefit of using strategies for unproductive phases is quantified in Section 9.4.*

1.4 Outline

The outline of this thesis is graphically illustrated in Figure 1.6. Part I comprises the motivating and introductory chapters of this work with focus on energy demand reductions within unproductive phases for factory automation systems. Part II contains the approach and scientific contribution of this thesis. The contribution of the approach to the research objectives is evaluated and discussed in Part III.

PART I is the point of origin for the formulation of the energy efficiency problem in unproductive stages and comprises the Chapter 1 (Introduction), the Chapter 2 (State of the art), and the Chapter 3 (Theoretical background).

PART II introduces the model-based engineering approach for energy-efficient operation of automation systems required to tackle unproductiveness. A conceptual and formal problem

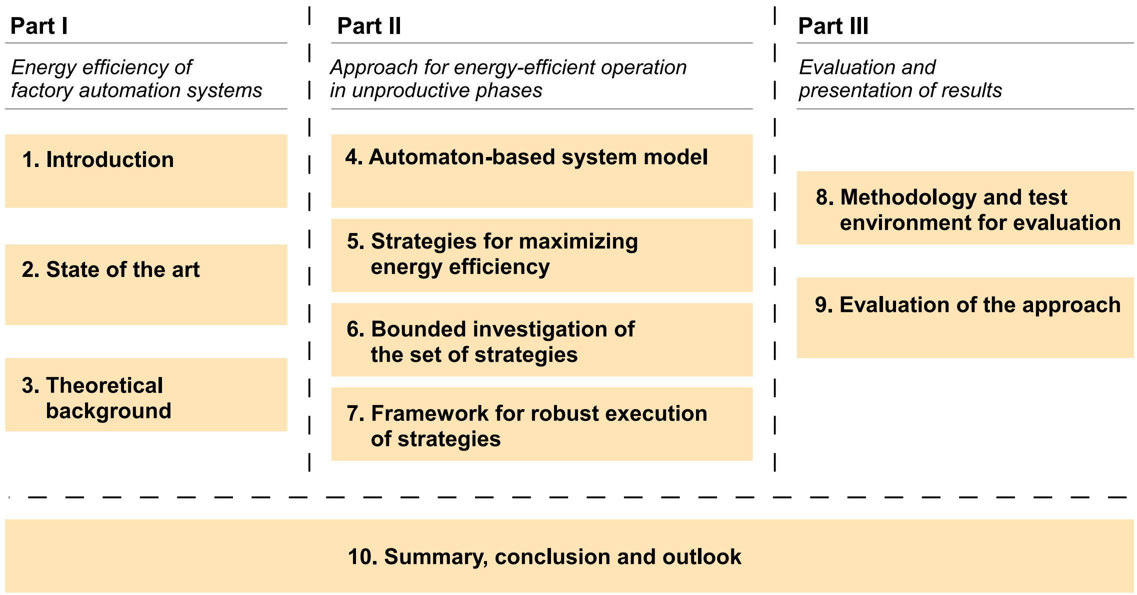


Figure 1.6: Thesis structure

description is given in Chapter 4. The interdisciplinary capability of being understood by engineers and computer scientists is respected by the proposed automaton-based system description which is based on the well-founded formal model of networked timed automata. The model enables the description of structural and behavioral aspects of the automation system while accepted Unified Modeling Language (UML) standards have been considered. Thus, the effective modeling support of energy-centric engineering of an automation system is ensured.
The formal model serves as analytical basis for the identification of strategies in Chapter 5. Alternative strategies are formulated as constraint optimization problems in order to identify the optimal strategy. The challenge in determination of the optimal strategy is rooted in the combinatorics of alternatives. On this account, a procedure using decomposing techniques is introduced to accelerate the identification of the optimal strategy in Chapter 6. Having found a strategy, the strategy needs to be checked for feasibility. A framework which supports the modeling of automation systems for unproductive phases as well as the generation of robustly executable strategies is focus of Chapter 7 required for evaluation purposes.
PART III evaluates the proposed approach regarding the identified research objectives. First, an introduction to the applied methodology and the test environment is given in Chapter 8. The method perspectives and evaluation objectives reflect the research objectives of this chapter (Section 1.3). The research objectives and the capability of the approach to contribute to these objectives is evaluated in Chapter 9.
A final review completes Part III in form of a summary, conclusion and outlook to future work (Chapter 10).

The economic pressure and the awareness of energy issues affecting the competitiveness of companies has resulted in intensive research trying to give an answer on the challenges mentioned in this chapter. The state of the art in the problem context of this thesis is reviewed in Chapter 2. Research already completed for energy-efficient manufacturing provides an impression of the importance and the attention to energy-related issues in today's production.

Chapter 2

State of the art

Planning, monitoring and controlling the energy demand in manufacturing are intensively discussed aspects in research and industry. Questions regarding energy efficiency and energy management are interpreted differently in diverse research domains and on different abstraction levels in factory automation. Since decisions concerning energy-related issues need to be made on different planning and control levels, a classification of state-of-the-art approaches according to abstraction level and system scope is chosen (Section 2.1 and Section 2.2). In Section 2.1, systems and subsystems are understood as a black boxes. In these approaches, it is abstained from the internal behavior of subsystems. In contrast, Section 2.2 presents white box approaches which allow to evaluate the internals of subsystems.
Each state-of-the-art approach is screened for its interpretation and use of energy demand as planning, monitoring and control quantity. A summary and identification of the research gap based on the state-of-the-art classification is given in Section 2.3.

2.1 Energy planning, monitoring and control using black boxes

2.1.1 Business planning

On business planning level, factories are regarded as black boxes since only aggregated energy demand and supply are relevant. The multi-site or multi-factory view is taken and the impacts of the factory network or the supply chain on energy demand, supply and prices is analyzed. For instance, peak load reduction of energy input is focused because it directly affects costs and profits on a business level. Company-wide demand-response systems for energy management aim at the integration of business and production level in terms of energy aspects [Müller, 2012]. The objective is to balance the energy supply and demand globally.

FLOREA ET AL. discuss a distributed energy management approach based on service-oriented architectures [Florea, 2012a]. This approach aims at ensuring cross-layer interoperability of system components as well as intraoperability of companies. On this account, it is focused on the interaction of Enterprise Resource Planning (ERP) and Manufacturing Execution System (MES) as well as the cooperation of different ERP systems. It is distinguished between strategic aspects of energy demand and supply on ERP level and operational energetical aspects on MES level. The proposed architectural design emphasizes the importance on MES level to integrate building and factory automation. The integrated view on energy management of building and factory automation is required in order to comprehensively manage energetical aspects [Florea, 2012b]. For this purpose, the structural and behavioral aspects of subsystems of manufacturing systems need to be analyzed in further research.

CASTRO ET AL. consider the planning stage of agreeing contracts between a plant operator and an energy supplier [Castro, 2009]. In these contracts, maximum input powers are limited within a specific time period. Exceeding a previously agreed threshold results in penalties. Using discrete-time and continuous-time optimization models help to take energy constraints into account. Therefore, these optimization models are compared in terms of computational efficiency in the scope of production scheduling represented as Resource-Task-Network [Pantelides, 1994]. Evaluation shows that continuous-time formulations are restricted to small problems. Discrete-time representation provides an efficient way to determine the optimal schedule in this problem class [Castro, 2011]. Focusing on productive phases of the plant, potential cost savings up to 20% can be realized by considering variable energy costs.

2.1.2 Multiple subsystems

Production Planning and Control (PPC) gives support for production planning, evaluation, and analytics. The mindset is process-oriented targeting on machine scheduling, capacity planing, and job order planing [Chiotellis, 2010], [Baehre, 2011], [Reinhart, 2011]. Though, the focus is on describing process-related effects rather than internal machine effects in production. Often, simulation-based approaches are applied in order to evaluate specific production designs in the context of several influencing parameters. Considering energy demand, the parameters of interest are the assignment of production orders to machines, the sequence of production orders, variation of batches, and backward/forward termination.

GUTOWSKI proposes an energy framework accounting for the trend towards energy-intensive processes [Gutowski, 2006]. The approach considers energy inputs and outputs of a manufacturing system. Electric energy per unit is evaluated. The energy demand of different subsystems (oil pressure pump, machining, etc.) is quantified for a production line in car manufacturing giving an impression of the influence on energy input. The approach enables to evaluate different manufacturing designs (for instance electrical vs. hydraulic drives).

RAGER develops an energy-centric machine scheduling formalism on identical parallel machines [Rager, 2006]. The author proposes the formulation of the optimization problem with energy as additional decision variable. The complexity of finding optimal solutions to the optimization problem is recognized recommending search methods based on heuristics. Consequently, the guarantee for optimality is abandoned for better computational performance.

JUNGE analyzes material flows and energy flows using a simulation system in order to implement a resource optimized process control [Junge, 2007]. The objective is to relate the production level to environmental and energetical aspects. For this purpose, the simulation of the material flow is linked to a thermal building simulation and a simulation of energy flows. The challenge consists in coupling the discrete event simulation with the simulation using continuous signals (thermal building simulation). The approach comprises multiple simulation runs while the discrete event simulation steps precede the continuing signal simulation [Junge, 2007, page 82]. This simulation setup has been realized distributively. The simulation components communicate using a specifically implemented middleware based on TCP/IP. The integration of the simulation system into the production control is provided [Junge, 2007, page 93]. In this work, the analysis is based on key performance indicators relying on a specific plant. The energy demand of the plant is checked in different scenarios, for instance with regard to energy demand with shortest operation time. Stated by the author, the realization of the simulation system in manufacturing systems seems to be difficult, since most of the information necessary for simulation setup might be not accessible.

HEILALA ET AL. propose a discrete event simulation within a virtual factory for maximizing production efficiency with respect to environmental impacts. The approach is used to analyze the manufacturing process in early stages of the engineering process towards energy efficiency and CO_2 emissions [Heilala, 2008]. The approach integrates environmental, ergonomic, and production aspects. In this way, the proposed tool provides decision support for engineering sustainable manufacturing systems. In the energy efficiency analysis [Heilala, 2008, page 1927], the energy demand calculation of a production is based on production modes. Energy information of a subsystem is assigned to modes. Slipping energy charges into energetical behavior of subsystems, energy savings potentials are calculated for production. The authors state that a significant amount of energy is demanded for unproductive periods in an automobile production line. The proposed tool covers aspects of the factory planning process. However, it abstain form detailed internals of individual machines.

HERRMANN AND THIEDE analyze a holistic description of production process chains towards energy input [Herrmann, 2009]. It is stressed that energetical aspects need to be integrated into production management. The holistic view on factories shows the complex interaction of different system components like technical building services and production machines [Herrmann, 2011a]. The approach is motivated by the energy demand of a grinding machine in different modes. The objective is to evaluate different production strategies towards energy efficiency. Four different layers are distinguished. The *input*, *logic*, and *user* layer serve as means to set up the simulation, the *evaluation* layer provides means to investigate the setting. The use of reusable modules make possible a quick setup of the simulation. It is explicitly stated that the production chain simulation does not reflect the internal behavior of single machines in detail. It is abstracted from the specific operating behavior of subsystems [Herrmann, 2007], [Herrmann, 2011b]. A typical question to be answered by the approach is related to the energy input per manufactured piece. Combined with energy load profiles and peak load analysis, the simulation can give hints on the configuration of energy contracts.

MÜLLER ET AL. focus on the energy-related decision support during engineering phases of a plant [Müller, 2009]. The approach described in [Müller, 2010] includes energetical information in the planning process of a factory. Additional decision variables are the productivity, quality, and flexibility of manufacturing. Constraints like equipment capacities, energy and compressed air supply are identified. The conceptual approach consists of a basic planning model comprising analysis, basic engineering, detailed engineering, and implementation within factory planning. A system view of the manufacturing system with its inputs and outputs completes the considerations on a high level.

NOLDE AND MORARI propose a scheduling approach to track the electric load of a steel plant [Nolde, 2010]. The objective is to schedule production tasks respecting a maximum total electricity consumption of all machines in the production. The problem is represented as mixed integer linear programming problem with continuous time. The approach enables the prediction of energy consumption in a steel plant to plan energy demand on factory level.

RAHIMIFARD AND SEOW use a holistic approach providing a model which relates energy to production output [Rahimifard, 2010], [Seow, 2011]. This approach provides additional insights into the energy consumption during the life-cycle of a product. For this purpose, the authors distinguish between different forms of energy required to produce a unit. The proposed simulation model abstains from details of single machines, but gives indication about the energy input of subprocesses. The simulation results can give decision support for designing manufacturing processes with respect to energy efficiency.

HAAG ET AL. integrates peripheral subsystems like heating, ventilation, and air conditioning into a material flow simulation [Haag, 2012]. Based on a production simulation, the objective is to find the energy-optimal working point of a factory integrating peripheral subsystems. It is explicitly distinguished between different unproductive and productive machine states like *waiting*, *blocked*, *broken down* on the one hand and *busy* or *work* on the other hand. An energy saving mode is mentioned which is meant for reducing energy demand in unproductive phases. It is stated that an overall management system is required to guide machines of the factory to this energy saving mode. The simulation model is calibrated with field level data (power measuring at each machine and peripheral system) which allows more precise energetical predictions. The proposed approach enables the support of systems engineering to consider energetical interactions and consumption in factories.

PUTZ ET AL. emphasize that the engineering of production lines in car manufacturing in early phases determines the running manufacturing system [Putz, 2011]. Design decisions during engineering have strong implications for the system runtime. In the context of the digital factory, the authors propose to couple the material flow simulation and energy efficiency considerations. The objectives of the energy-focused simulation are the dimensioning of manufacturing systems and the optimization of production processes. It is argued that an analytical model is not applicable because of complexity issues. The approach is motivated by energy demand reduction for manufacturing car bodies including manufacturing techniques like laser welding, spot welding, and bonding. Material flow is realized by robots. The production line comprises 30 industrial robots in 5 different stations. The energy-oriented simulation distinguishes production states and assigns load profiles (characteristics of input powers for a given interval) to these states. The model is set up with the help of the commercial simulation tool *Plant simulation* which enables to analyze models of manufacturing systems based on discrete event simulation. In this way, production processes can be evaluated and redesigned according to energetical guidelines.

DIETMAIR ET AL. introduce a conceptual framework for modeling and predicting energy demand in manufacturing systems [Dietmair, 2011]. The model-based optimization of energy demand is emphasized as important aspect besides due times and product quality. Guidelines for model-based evaluation of energy consumption are provided in the context of the framework. The objective is to propose a composite multi-machine model to describe and analyze the energy input of the subsystems of a manufacturing system. Finding an acceptable operating point of the manufacturing system is supported by discrete-event simulation.
Considering a multi-machine network view with special respect to supply units (for example pressured air) is seen as essential for a comprehensive framework. The proposed model is based on the Petri net formalism explicitly respecting functional dependencies between different machines. Synchronization transitions and additional places are used to model component dependencies. Places are annotated with input power accounting for energetical costs in a machine state. The input power of machine states is derived by kinematic models representing

the input power of manufacturing processes (for example milling processes).
In addition, the authors discuss the necessities and possibilities of energetical optimization from different points of view. First, the difficulty of optimal machine parameter selection is presented. In a second step, logic and sequential program optimization is illustrated for a single machine. A state graph includes machine states, transitions between states, as well as transitional times and input power of states derived by a discrete-event simulation. Temporal constraints are given with minimum and maximum times provided for a state as a result of the simulation. A third aspect is the factory-level planning and control optimization which combines a stochastic material flow simulation and the energy consumption model. This view links energetical aspects to scheduling strategies. The last facet is the optimization of the Numerical Control (NC) programs of machine tools which requires a detailed look inside the physical and chemical aspects of the manufacturing process.
The composite machine model uses the graphical power of Petri nets. Although modularity is emphasized in the model for dealing with complexity arising from multiple subsystems, it is abstained from difficulties of handling large Petri nets. From simulation runs, temporal information for the state-based model (state graph) is derived. It is stated, that a method for identifying and evaluating alternative state trajectories is required to find the optimal trajectory.

YANG ET AL. emphasize the importance of considering energy aspects in early design stages of a factory [Yang, 2011]. A facility planning approach is enriched by heat exchange networks integrating the energetical view on production. The authors focus the energy recovery in those networks accounting for the regain of energy. Three cost categories are considered: material transport, manpower, and energy costs. Costs are optimized using a linear program. In a case study, two layouts are compared to each other in a production process with regard to heating and annealing workpieces. Layout 1 considers material transport as optimization angle whereas for layout 2, a multi-criteria optimization based on the same cost quantifier is applied. Concluding, the authors show that layout 2 incorporating energy aspects reduces costs compared to layout 1. The approach abstracts from inner machine behavior.

ARTIGUES AND HAÏT discuss energy-related scheduling of jobs in a foundry in order to minimize the overall energy demand [Artigues, 2011]. The authors propose a hybrid heuristic based on a constraint programming approach. The problem is formulated as mixed integer linear programming problem with continuous variables. A heuristic is applied for computational reasons. Within a steel plant's heating and melting processes, the energy demand is mathematically optimized by this approach.

WOLFF, KULUS, AND BERGER link a material flow simulation with energetical considerations [Wolff, 2012]. Energy efficiency as an additional strategic and planning factor besides produc-

tion time or invest is analyzed. The authors center the energetical view which influences the alternatives in factory design. Therefore, the objective is to augment digital factory tools with energy flow simulations in order to give decision support for energetical design of manufacturing processes. A proof of concept is illustrated integrating the energy-efficiency view into an off-the-shelf factory simulation tool. The discrete-event energy simulation is exemplified by a production line of a car manufacturer [Kulus, 2011].
The first part of the simulation approach is the identification of operational states and the mapping of these states to so called energy states. Operational states for subsystems are derived by a material flow simulation. For different operational states, energy states are defined which link information of machine operation to input power. Typical manufacturing subsystems are included in the model implementation: transfer machines, machining centers, portals, inspection stations, and buffers. The execution phase supports the scenario-based analysis of influencing parameters and the subsystem behavior. The simulation software supports a decision maker in evaluating the aggregated energy consumption of the system as well as of single subsystems [Berger, 2012]. Mentioned by the authors, evaluating the energy states of subsystems requires detailed models of the internal behavior.

2.1.3 Summary

Approaches for energy planning, monitoring and control taking a black-box view are presented in this section. Energy aspects regarding business planning and (multi)-factory levels focus on energy demand and supply of complete factories and corresponding inter-factory aspects.
A more detailed view is taken by PPC approaches which give support to design and dimensioning of manufacturing systems with regard to energetical objectives [Reinhart, 2011, page 598]. On the abstraction level of PPC, detailed internal aspects of individual subsystems is out of scope. Approaches include behavioral aspects of subsystems by describing the subsystem states. The focus of PPC approaches is on energetical aspects of productive phases. Many presented concepts in planning of the manufacturing processes are formed by the thinking in terms of discrete event systems [Shanon, 1938]. Simulation-based approaches are frequently used in order to make statements about future or expected energy demand of a production system. Discrete-event simulation techniques are suitable for dimensioning and coupling material flow considerations with energy efficiency aspects [Berger, 2012]. However, special weaknesses can be identified in terms of optimization cycles and proposals [Thiede, 2012, page 83]. Expertise is needed for tuning the simulation model towards optimality of a decision variable. Simulation-based optimization methods rely often on heuristics (genetic algorithms, tabu search, simulated annealing) to cope with complexity issues unable to guarantee optimal solutions. Simulation-based optimization [Andradóttir, 1998], [Fu, 2002], [Deng, 2007] integrating optimization methods into simulation analysis with multiple variables suffers from inherent complexity.

2.2 Energy planning, monitoring and control using white boxes

In contrast to the black-box view taken in Section 2.1, this section presents energy planning, monitoring and control approaches considering detailed internal behavior of subsystems. In these approaches, energy information is closely related to specific subsystems. It is distinguished between approaches for multiple subsystems (Subsection 2.2.1) and approaches for single subsystems (Subsection 2.2.2).

2.2.1 Multiple subsystems

VIJAYARAGHAVAN ET AL. investigate machine tools with regard to energy monitoring. The manufacturing process is related to the energy demand of machines in complex manufacturing settings. The objective is to correlate energy usage with operation of the manufacturing system. Therefore, event stream techniques are applied in order to monitor energy input patterns in large systems [Vijayaraghavan, 2010]. The software-based approach for automated energy reasoning enables the concurrent monitoring of energy demand related to process data for different system scales. Reasoning is based on a rule engine respectively complex event processing engine implementing the Rete algorithm. Linked to event stream techniques, reasoning over time of energy demand can be realized. Different modes with related energy demand of a machine (start up and shut down, idling, and processing modes) can be distinguished in this way. Since this is a monitoring approach, the control of the production process is not addressed.

WEINERT separates the production process into segments with specific input powers [Weinert, 2010]. With the help of a mathematical description of the input power of production modes, the input power of subsystems within a production process can be represented. Since this analytical approach focuses on the production planning process of factories, this approach set ups individual, prescriptive models of the energy consumption of individual subsystems. This enables to calculate energy efficiency measures for different operating modes. The proposed approach derives energetical modes or states by observing the operating behavior of subsystems. This monitoring technique is limited to the runtime of the manufacturing system.

BECK AND GÖHNER introduce a model-based approach for optimizing energy demand in automation systems [Beck, 2011]. It is distinguished between situation-dependent and situation-independent knowledge affecting energy-centric decisions. Production processes are screened for energetical influences with the help of knowledge modeling. In [Beck, 2012], the authors investigate a system consisting of two subsystems. Additionally, a human modeler and installer is considered in a user-centric approach identifying an energetically tuned operating point. The operation of two related automation subsystems is identified in a four-levels analysis. The four-levels analysis is an iterative and user-interactive approach to increase the energy-efficiency of the two subsystems. The model-based approach providing structured knowledge requires detailed information about the production process. Moreover, in complex industrial settings, the challenge of this approach lies in separation and identification of the functional causality between process parameters (cause) and energy demand (effect). Furthermore, the user-centric approach might become complex if a modeler has to optimize several interrelated or interacting subsystems.

SCHLECHTENDAHL ET AL. describe a framework for monitoring energy consumption of machine tools [Schlechtendahl, 2013]. The framework mainly consists of providing an information model of relevant energetical aspects. In the approach, subsystems are classified according to the capability to be guided to different machine states. The term strategy is used to denote monitoring techniques which record the energy demand of machines. The importance of knowledge-based information modeling is recognized by the use of an energy information description language (EIDL). This description language enables to represent structural as well as behavioral information about modeled subsystems. So called live values denote measured information that is available during runtime of a subsystem. An interface using subsystem communication completes the proposed framework. PROFIenergy, a profile in industrial communication, is considered as interface to subsystems. Approaches to use communication technologies to address device energy saving modes are developed by organizations like CAN in Automation and Profibus International (PI). A representative approach (PROFIenergy) is introduced by PI integrating powering commands into communication technology which enables the access of operating modes of devices on field level [PNO, 2010]. The potential of switching off devices is shown in [PNO, 2011]. In a study [Hübner, 2011], an energy savings potential of one third is identified by switching off subsystems partially. Currently, the solution provides only two distinct operating modes of a PROFIenergy-enabled device: a sleep mode and a productive mode. A complex internal behavior of a subsystem is not representable in this way. Intermediate operating modes like distinguishable stand-by modes need to be included additionally. The changeover between the two possible modes is initiated via sleep commands defining the period for a device being in the sleep mode. Since the profile addresses single devices, the structure of a set of devices is not considered. An information model how devices interact needs to be setup beyond this profile.

2.2.2 Single subsystems

Elements of the multiple machine approaches are machine tools and smaller automation subsystems. The approaches for single subsystems are presented here. It is distinguished between machine-tool approaches and approaches addressing embedded devices.

Machine tool approaches

DEVOLDERE ET AL. analyze the unproductive phases of a press brake and a milling machine. The total share of modes representing unproductiveness in the overall energy consumption of the press brake is 65% including idling phases [Devoldere, 2007]. In case of a milling machine, the energy input for idling phases is at the amount of 13 percent of the overall energy input of the machine. The energy consumption during production of an investigated laser cutting process shows that the energy input of idling aggregates to 5% of the overall energy input of the machine [Devoldere, 2008]. Nevertheless, the studies show that a considerable amount of energy has to be invested in order to last the machine tools in idling modes. Although the studies do not reveal the absolute time intervals of the different unproductive phase of the machines, the studies give indication to energy demands of industrial machine tools.

MOUZON AND YILDIRIM analyze the scheduling of jobs and power consumption based on a greedy and randomized adaptive search heuristic [Mouzon, 2008]. In [Yildirim, 2012], the approach is developed further and provides the estimation of the Pareto front in the evaluated multi-objective optimization problem. Finding Pareto-optimal solutions consists in a construction phase and a local search phase. The multi-objective optimization problem considering the two parameters, energy input and tardiness, for a single machine in a manufacturing environment is presented and the computational complexity in finding optimal solutions is stated. The authors provide an integrative view on the energy demand of productive and unproductive phases. The approach provides optimal schedules with respect to energy demand and tardiness of a single machine. Since the focus lies on a single machine, no system view introducing process-related dependencies between machines is provided.

DIETMAIR AND VERL propose a method to model the energetical behavior of subsystems. The approach is used to forecast the power drain profile and to optimize machine operation [Dietmair, 2009b]. By measuring the input power of different states in a machine, the energy input for specific manufacturing processes can be precalculated. A composite model based on a Petri net formalism that takes process-related and functional dependencies into account is introduced [Dietmair, 2009c, pg. 230]. The objective of the approach is to give decision support for machine operation by a state-based model. The proposed model is intended to predict the energy input during production as a mathematical sequence of operating modes [Dietmair, 2009a]. Alternatives in operation within unproductive phases are mentioned but not analyzed [Dietmair, 2009c, pg. 232].

NEWMAN ET AL. investigates the energy input of a production process on machine level. In [Newman, 2012], a framework for the planning process in order to analyze the power consumption for a milling process is introduced. Manufacturing processes are modeled as a set of operations. Assigning energy demands by an energy usage function, different manufacturing processes can be compared based on a parameter variation. The cutting process is investigated in detail.

Automaton-based approaches

BOUYER studies the problem of infinite schedules for one-clock, weighted timed automata. This problem derived from automata theory has benefits on modeling embedded devices with limited energy resources. The objective of the approach is the generation of energy optimal schedules for infinite, safe operations in the context of resource optimal scheduling [Bouyer, 2008]. These schedules need to be synthesized for embedded devices where resources are consumed (electrical energy is transformed into mechanical energy) and are gained (electrical energy of the sunlight is transformed into electrical energy of a robot equipped by solar cells). This approach can be applied to individual embedded devices (single, autonomous controllers in factory automation) with energy constraints. In another work, the cost-optimal reachability problem is studied from the computational point of view [Bouyer, 2007]. In an earlier work, [Bouyer, 2004] proposes a dual-priced timed automata with two price parameters, costs and rewards. An optimization approach of a cost-reward ratio is proposed.

RASMUSSEN evaluates minimum-cost reachability analysis in the context of energy-optimal task graph scheduling [Rasmussen, 2004]. Task graph scheduling consists in distributing interdependent tasks on multiple processors. Temporal constraints between tasks need to be respected. Precedence constraints are formulated as task graphs. Tasks in task automata must be expressed with predefined time bounds. The optimization problem is solved relying on priced timed automata compared to a mixed-integer linear programming approach.

Petri net approach

SHORIN AND ZIMMERMANN develop a simulation model and tool using a stochastic Petri net formalism (TimeNET) [Zimmermann, 2007] in order to describe the energetical aspects of automation controllers. For specification of the controller behavior, extended MARTE (Modeling and Analysis of Real-Time and Embedded Systems) (see [Arpinen, 2012]), an extended version of the specific UML profile [OMG, 2010] is deployed. The objective is to provide model-based engineering support with focus on energy information of controllers [Shorin, 2010]. The evaluation is based on an implementation for embedded devices. Energy-tuned operation of the micro controller is identified by simulation runs. Another contribution is dedicated to the transformation of UML state charts to stochastic Petri nets with special regard to energy aspects of embedded devices [Shorin, 2012]. While energetical behavior of the embedded device is analyzed, the energetical facets of a physical production process is disregarded.

Dynamic power management

Dynamic Power Management (DPM) is a methodology for designing electronic devices and provides services to access the operating state of the components of a device (CPU, RAM, display, etc.). An open standard for power management has emerged from this consideration called *Advanced Configuration and Power Interface (ACPI)* [ACPI, 2011]. The objective is to guide devices and its components to low power states while idling [Benini, 2000]. Stochastic models are applied describing the use of components [Šimunić, 2001], [Rong, 2006]. DPM assumes that the components of the device work at different workloads during system operation. The view of DPM is restricted to single electronic and computing devices.

SRIVASTAVA show three different approaches for shutting down embedded devices. An approach for predictive shutdown in personal digital assistants (PDA) taking the cost of the shutdown into account is provided by [Srivastava, 1996].

SWAMINATHAN provides an approach for handling embedded devices with multiple power states [Swaminathan, 2003]. The discussion is centered around device-scheduling algorithms for hard real-time systems reducing the energy input of Input/Output (I/O) devices. The algorithms guarantee that task deadlines are not missed if shutdown functions are implemented. Interdependencies of tasks are considered for single I/O devices.

ARPINEN proposes a profile enriching DPM approaches with power information by means of UML modeling [Arpinen, 2012]. The introduced profile associates power information to states of hardware components. The energetical information of hardware components is modeled as state machines with time annotations. This approach focuses on power states of the individual hardware component and does not account for a controlled physical process.

IRANI AND PRUHS review the state of the art in algorithmic problems in power management [Irani, 2005]. The focus is on single embedded devices with limited energy supply. The authors give an overview of the approaches proposed in processor speed scaling, and power-down strategies.

2.2.3 Summary

Energy planning, monitoring and control taking a white-box view regarding subsystems is in the focus of this section. The presented approaches center on internal behavior of subsystems. On machine-tool level, the analysis of specific manufacturing processes becomes the focus of attention (the energy demand linked to specific production parameters [Draganescu, 2003]). Related work with regard to embedded devices is separated into automaton-based approaches, Petri net approaches, and dynamic power management. Power management approaches exist already for a while in Information and Communication Technology (ICT) [Rinaudo, 2011] and for embedded devices with limited resources [Chen, 2012]. The energetical relevance of unproductive phases of single subsystems is acknowledged by some approaches. Naturally, power management for embedded devices takes a single-subsystem view without considering physical production processes.

2.3 Summary

The amount of research on all abstraction levels illustrates the importance of considering energetical aspects. The approaches materialize the abstract definition of energy management (Def. 1) and energy management systems (Def. 2). State-of-the-art approaches have been reviewed with regard to energy planning, monitoring, and control. It has been distinguished between approaches which take a black-box view and those which take a white-box view.

Black-box approaches abstain from internal subsystem details, although some approaches consider distinguishable subsystem states. While different approaches recognize the relevance of energetical consideration of unproductive phases, research presented in Section 2.1 focuses on productive phases. In this context, energy demand of manufacturing is considered as one aspect among others like material flow and due times. Operational instructions to guide subsystems to specific energy saving modes can not be derived based on this abstracting view. Furthermore, discrete event simulation "assess the performance of previously identified solutions of a decision maker" and can not guarantee to "produce an optimum answer to decision under study" [Bradley, 1977, page 4].

In contrast, white-box approaches, explicitly consider the internals of subsystems. The integration of unproductive stages is considered in many approaches on this abstraction level. Approaches assume machines and subsystems as independent form each other. Especially, approaches for single subsystems ignore the inter-component dependencies. This is especially insufficient if trying to propose instructions for unproductive phases how interdependent subsystems should be guided to energy-efficient modes. Proposing strategies for energy demand reductions within unproductive phases requires to explicitly describe subsystem dependencies. Moreover, alternatives for operating subsystems in unproductive phases are not evaluated and compared to each other in an analytical way.

The problem of complex interaction of subsystems while trying to apply strategies within unproductive phases for energy savings is not addressed adequately by state-of-the-art approaches. Consequently, a research gap concerning energy planning and control can be stated which is graphically illustrated in Figure 2.1.

Because of missing model and evaluation support of state-of-the-art approaches for deriving strategies to address unproductive phases, this thesis provides a white-box approach for mul-

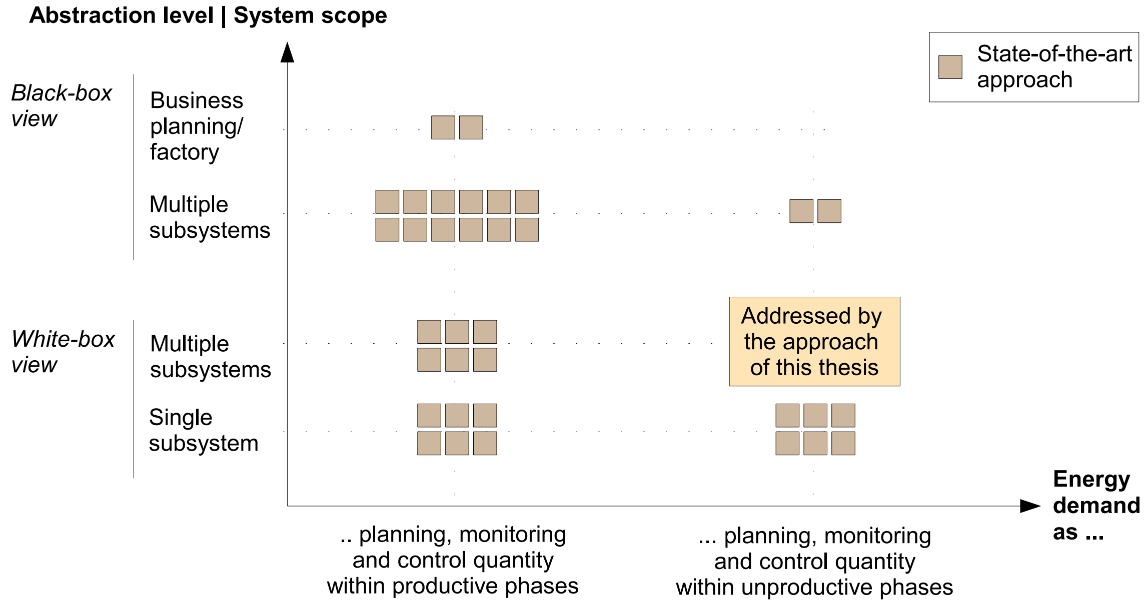


Figure 2.1: Approach classification according to addressed abstraction level, system scope and energy planning, monitoring and control

tiple subsystems. A formal model describing the internals of subsystems and the subsystem dependencies is proposed. A derived strategy optimization model makes the computation of energy-optimal strategies for unproductive phases possible. The importance of model support enabling the description of modes of automation subsystems is emphasized by the International Electrotechnical Commission (IEC). Energy-related aspects need to be incorporated into factory automation design in order to benefit from energy savings potentials [IEC, 2011, pages 19, 21]. The methods applied in this thesis allow to link the detailed analysis of the energy savings potentials for interacting subsystems to verification and validity considerations. The identified research gap is picked out as a central theme in Part II. Before presenting the approach of this thesis, the theoretical background is given in Chapter 3.

Chapter 3

Theoretical background

In the introductory chapter, research objectives are identified which are important to realize strategies for reduced energy demand within unproductive phases. These objectives require an adequate formalization to analyze and contribute to energy efficiency.
Today, automation systems, which are the addressed technical systems of this thesis, tend to be modularized and divided into subsystems [Lunze, 2006]. To give an overview of the structure and subsystems, a classification according to function and control is used for factory automation systems (Section 3.1). Systems theory and models for Discrete-Event System (DES) are applied in order to map the structural and behavioral aspects of factory automation systems to an analyzable formalism. Systems theory (Section 3.2) serves as the fundamental basis for the proposed model of this thesis (Section 3.3). Discrete event approaches are determined to provide a basis to describe strategies formally. Constraint optimization techniques provide analytical means to identify optimal strategies and to compare alternative strategies energetically.

3.1 Automated manufacturing systems

Automated manufacturing systems are technical systems which comprise machines, transfer systems as well as communication and control infrastructures. By various (rigid or flexible) interactions of machines regarding communication and material flow, a complex technical system arises. First, an overview of machine tools is given to illustrate the nature of technical systems and subsystems addressed in this thesis (Subsection 3.1.1). This is followed by a presentation of the functional (Subsection 3.1.2) and control (Subsection 3.1.3) structure of modern automation systems.

3.1.1 Machine tool classification

Automated manufacturing systems consist of multiple machines and other subsystems with specialized responsibility in the production process. The machines work cooperatively in order to fulfill an automation task or a manufacturing objective. As it is indicated in Section 1.2, machine tools are subsystems relevant to be considered regarding their energy demand within unproductive phases. A classification of machine tools can be found in the DIN[1] 69651 [Weck, 2005]. The functional classification illustrated in Figure 3.1 divides automation subsystems with reference to productivity and flexibility. Machine tools can be classified according to productivity and flexibility and based on the number of machines in a system: individual machine tools and multiple machine tools.

Individual machine tools

In general, general purpose machines do not integrate own control systems. A manually operated Lathe can serve as an example. Machines with integrated NC or Computer Numerical Control (CNC) are capable to execute a predefined (programmed) manufacturing process automatically (a lathe with integrated NC for instance). Processing centers enlarge the capabilities of NC machines by integrated, automatic tool changing as well as a tool magazine. Milling machines in manufacturing can be constructed as processing centers with automatic tool change and tool magazines.

Multiple machine tools

Flexible machine cells are able to process a higher amount of manufacturing parts than processing centers. A machine that combines the functions of CNC milling and labeling may serve as an example. A Flexible Manufacturing System (FMS) consists of several machines which are linked via transfer systems. FMS exhibit higher flexibility compared to (flexible) assembly lines because of high flexibility in routing workpieces. An example for FMS is the electronics assembly equipped with flexible routing of parts. Flexible assembly lines are linked linearly so that they produce higher outputs (higher productivity) than FMS. The assembly of different printed circuit boards is often realized via flexible assembly lines. Generating the highest output, the assembly of car models is implemented in assembly lines. These lines show the highest productivity linked to inferior flexibility.
Subsystems addressed by this thesis can also be classified regarding functional properties.

3.1.2 Functional structure

Manufacturing systems are often modular and hierarchical structured. Since the interpretation of these terms strongly depend on the manufacturing domain, a generic taxonomy cannot be given. In general, functional properties of manufacturing entities can be used to give a hierarchical order of subsystems. The division of subsystems into assembly components and man-

[1] DIN: germ. Deutsche Industrie Norm, German industrial standard

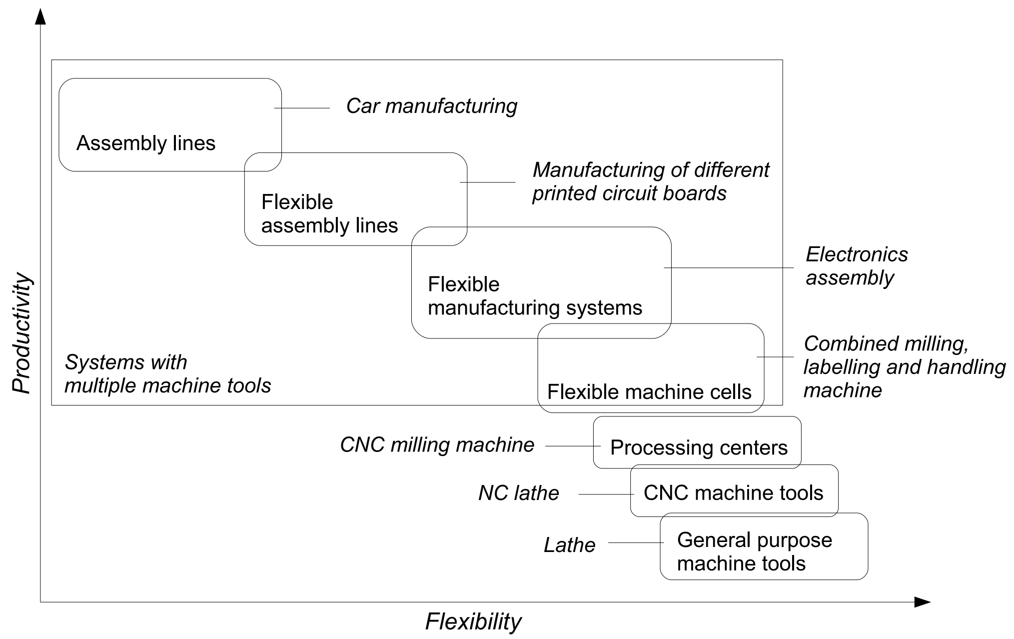


Figure 3.1: Machine tool classification in [Weck, 2005, page 411]

ufacturing components is one classification approach [Pahl, 2007]. Nevertheless, functional classification provides an appropriate abstraction level to handle the complexity of different interacting subsystems.

The functional view is clarified by the following example illustrated in Figure 3.2. A facility is an exclusive entity in which an *automation task* is realized. A typical automation task of factory automation is *the transportation and change of the properties of discrete elements*. This automation task is implemented by machine functions on machine level. For instance, the filling of bottles is one subtask implemented by the filling machine. This subsystem consists of components supporting the automation subtask called *functional groups*. The filling machine hosts an entity which stores and separates discrete parts. Moreover, the functional group comprises few *functional units*. The actuation function (electrical motor of the parts storage) may serve as an example. A *function unit* divides into *functional elements*. A *functional element* of the electrical motor might be a strap for power transmission.

3.1.3 Control structure

Apart from the manufacturing functions of subsystems, automation systems posses a control infrastructure for automatic control. A common approach to structure manufacturing systems with regard to control aspects is the four levels classification in Figure 3.3. The most abstract level is the *management level* of the factory with software systems that are called ERP software systems. Mid- and long-term decisions concerning the production are made on this level. By

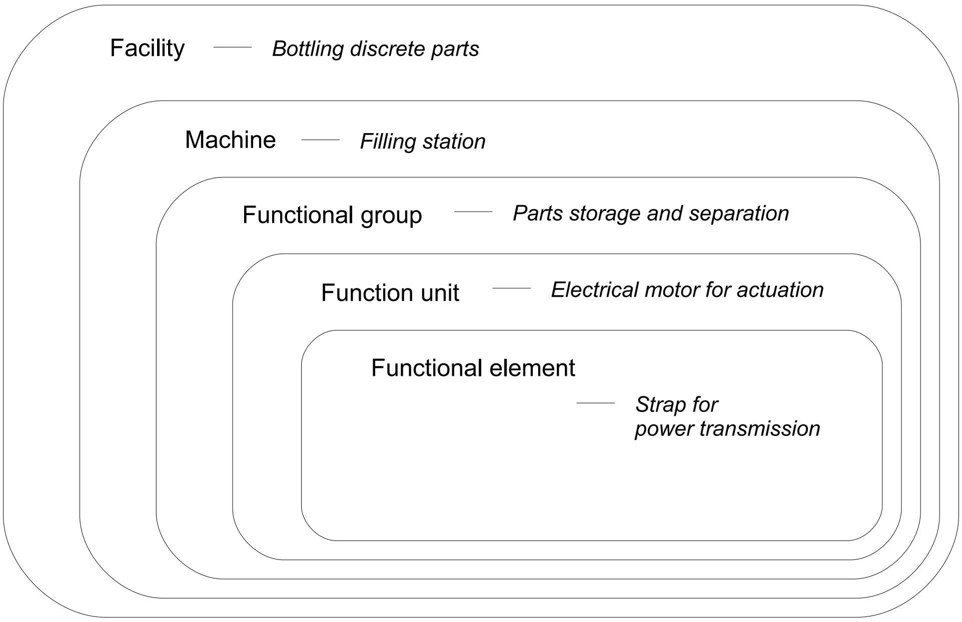


Figure 3.2: Functional view for hierarchical structuring of manufacturing systems

using PPC, production schedules can be identified meeting specific objectives like due times or machine capacities. In several steps, customer orders of the ERP and PPC are transformed into detailed production orders on *process control level*. The plan for specific production steps and production scheduling is made by MES. The execution of the production plan in a *production line* or in a *production cell* is supervised by *line supervisors* reporting to a Process Control System (PCS) providing to human supervisors a current image of the production process. *Line supervisors* are control units with specific monitoring responsibility. On *field level*, Control Unit (CU) (Programmable Logic Controller (PLC), Industrial Personal Computer (IPC)) can be found which actively control and interact with the physical production process specified by a PCS. The control is often realized using industrial communication between CU and I/O devices. *Actuators* (A) and *sensors* (S) (on *actuator/sensor level*) serve as interaction components between control units and production process. Moreover, CUs may communicate on field level with other CUs like those of machines (M). Machines can be interpreted as physical compact systems with integrated CU as well as actuators and sensors.

Both, the functional classification and the control structure illustrate the modular setup of today's automation systems. Modularity requires subsystems to communicate information in order to fulfill the overall automation task. In general, the message exchange between modular automation subsystems is carried out asynchronously. The asynchronous message exchange demands to have a closer look at systems theory and DES. The following concepts and models enable the representation of asynchronous information exchange based on messages or events.

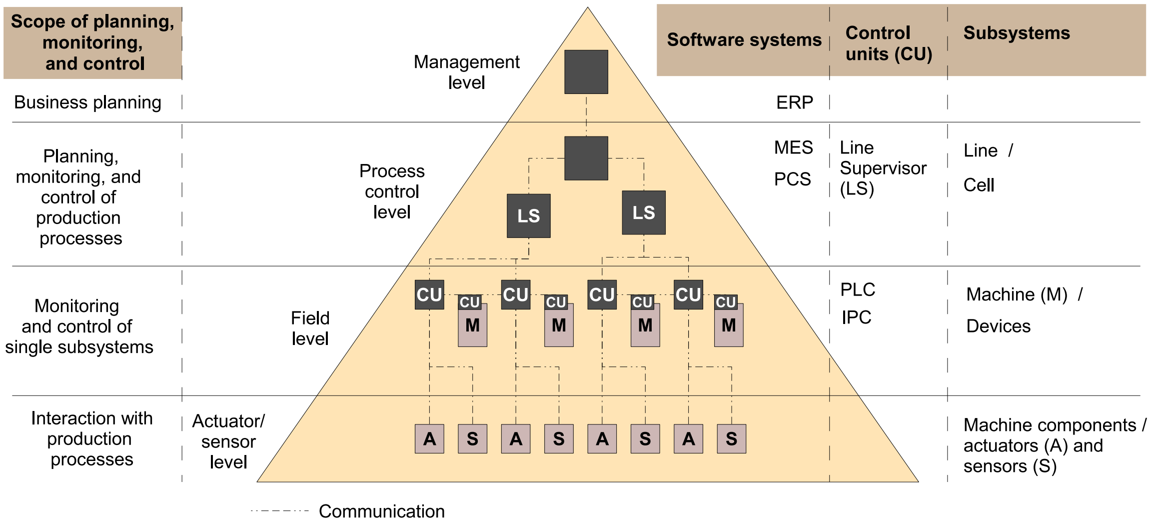


Figure 3.3: Canonical structure of today's industrial automation systems

3.2 Systems theory

Systems theory as meta theory, provides an abstraction regarding the dynamics of technical systems. The introduced technical terms enable the representation of the structure and the dynamics of manufacturing systems [Lunze, 2006, page 25].
Within a system, a physical process is a course of events in which energy, material or information is transformed, transported or stored (Fig. 3.4).

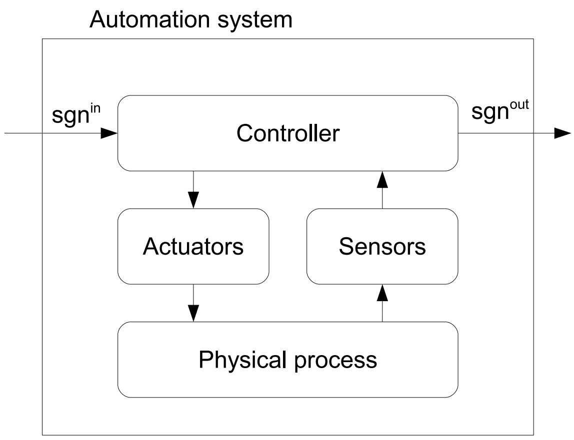


Figure 3.4: Control and process of an automation system

Systems in industrial automation are entities in which the controller and the physical production process are coupled via sensors and actuators. A system is demarcated by system boundaries distinguishing the system from its environment [DIN-IEC-60050, 2012]. In general, a system is an entity with a specific structure as well as a certain behavior [Forrester, 1968], [Zeigler, 2005]. The controller can communicate over these system boundaries via specific interfaces that are called signals *Sgn*. The signal sgn^{in} denotes an input signal, sgn^{out} denotes an output signal respectively. The general characteristic of a system sys is being composed of subsystems sub_i (Fig. 3.5). The action of stripping the system into spare parts is called *decomposition*. These subsystems are entities of the system interacting with each other so that a complex relationship inside the system arises. In the manufacturing domain, systems are often modularized as it is illustrated with sub_1, sub_2, and sub_3.

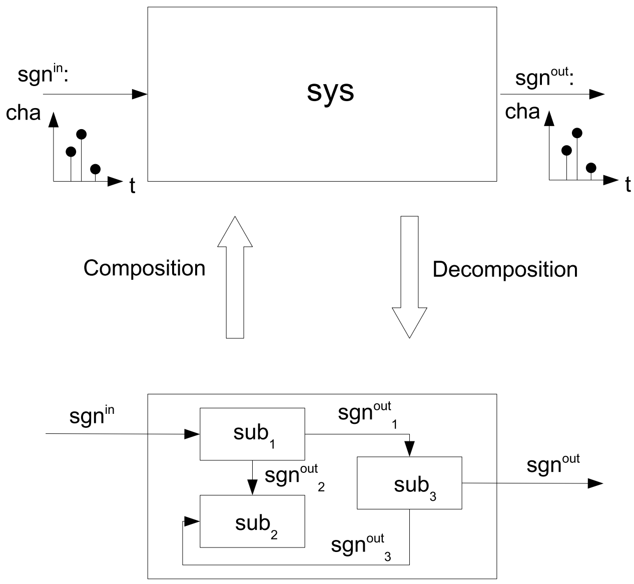


Figure 3.5: Hierarchical and modular systems model

The counterpart of decomposition of a system is called *composition*. This compositional aspect allows building up modular and hierarchical system structures with interacting relations presented in the decomposed view of system sys. The composed system view is also called *black-box view* whereas the decomposed system view is named *white-box view*.
Central terms used are *time*, *signal*, and *state*. Time is represented as time sequence T (Def. 4) and can be regarded as medium to sort past, current and future values of characteristics of processes.

Definition 4 (Time sequence)
A time sequence $T = t_0, t_1, ...,$ is an infinite sequence of values $t_i \in \mathbb{R}_0^+$ as following:

- *t increases with strict monotonicity: $t_i < t_{i+1} \; \forall \; i \geq 0$*

- *for any $t \in \mathbb{R}_0^+$ there is progress, so that $\exists\ t_{i+1} > t_i$ with $i \geq 0$*

Signals Sgn are abstract descriptions of characteristics in processes. A signal *sgn* is a function which maps the set of exact points of time t_i to the set of values of this characteristic *cha*:

$$\mathrm{sgn} : \mathrm{t}_i \mapsto \mathrm{cha} \tag{3.1}$$

With the concepts of time and signals, the system ability to transform input signals $\mathrm{sgn}^{\mathrm{in}}$ to output signals $\mathrm{sgn}^{\mathrm{out}}$ can be introduced. The system maps

- the set of input signals $\mathrm{Sgn}^{\mathrm{in}} = \{\mathrm{sgn}^{\mathrm{in}}(\mathrm{t}_1), \mathrm{sgn}^{\mathrm{in}}(\mathrm{t}_2), ..., \mathrm{sgn}^{\mathrm{in}}(\mathrm{t}_n)\}$
- to the set of output signals $\mathrm{Sgn}^{\mathrm{out}} = \{\mathrm{sgn}^{\mathrm{out}}(\mathrm{t}_1), \mathrm{sgn}^{\mathrm{out}}(\mathrm{t}_2), ..., \mathrm{sgn}^{\mathrm{out}}(\mathrm{t}_n)\}$
  and $\mathrm{t}_1, ..., \mathrm{t}_n \in \mathrm{T}$
- by function $\mathrm{s} : \mathrm{Sgn}^{\mathrm{in}} \rightarrow \mathrm{Sgn}^{\mathrm{out}}$

Signals are linked to states (Equ. 3.2 and Def. 5). In DES, the future state $\mathrm{st}_k(\mathrm{t}_i + \mathrm{t}_{i+1})$ of a system can be derived by the current state st_k of the system, its input signal $\mathrm{sgn}^{\mathrm{i}}(\mathrm{t}_i)$ and a function *b*. Function *b* represents the behavior of the system and is called the system function.

$$\mathrm{st}_j(\mathrm{t}_i + \mathrm{t}_{i+1}) = \mathrm{b}(\mathrm{st}_k, \mathrm{sgn}^{\mathrm{i}}(\mathrm{t}_i)) \tag{3.2}$$

Definition 5 (State)
A state st_k is a distinct and distinguishable status of a system or a set of subsystems with specific properties at point in time t_i.

Equation 3.2 implies that state changes are deterministic, so that successor states can be mathematically identified based on the current state of the system. This is in contrast to classical systems theory where state changes happen in continuous time. The discrete behavioral aspect *b* of a system can be interpreted formally as a temporal relation between states as part of a *timed transition system* (Def. 6) [Laroussinie, 1995, page 4]. This enables to define a structure characterizing states and state changes.

Definition 6 (Timed transition system)
A timed transition system is a tuple (St, st_0, $\rightarrow$):

- *A set of states St*
- *An initial state st_0 with $st_0 \in St$*
- *A timed transition relation: $\rightarrow \subseteq St \times \mathbb{R}_0^+ \times St$*

There are two main implications of signals mentioned here: *time* and *structure*. Signals occur at specific points in time which serve as information carrier between the system on the one hand and the environment on the other hand. Since a system can also be hierarchically embedded in another system or being orthogonal to another system, signals can be used to exchange information between systems as well. This structural and temporal quality of signals need to be adequately represented by a model. On that account, models and concepts for representing temporal and structural aspects are introduced next.

3.3 Timed discrete event systems and the reachability problem

DES [Shanon, 1938] describe the sequence of events (state changes from st_k to st_{k+1}) evoked by triggering signals in controlled systems and belong to the set of logical models [Inan, 1988]. A DES is described by the following formalism (Def. 7).

Definition 7 (Discrete event system)
A discrete event system DES is a tuple
(St, st_0, In, Out, tr_{int}, tr_{ext}, f, τ) [Zeigler, 1976], [Zeigler, 2000]:

- *A set of states St, and initial state st_0*
- *A set of inputs In*
- *A set of outputs Out*
- *An internal transition relation: $tr_{int} :\subseteq St \rightarrow St$*
- *An external transition relation: $tr_{ext} :\subseteq St \times In \rightarrow St$*
- *An output function f: $St \times In \rightarrow Out$*
- *A state delay function τ: $st_k \rightarrow st_k(t_{i+1})$*

[Zeigler, 1976] introduces a formal description using the Discrete Event System Specification (DEVS). It allows the formal modeling of system behavior using inputs and outputs. Inputs and outputs allow for modeling communicating subsystems. This concept has been extended in many ways resulting in the introduction of several models like *I/O automata* [Tuttle, 1984] and *interface automata* [Alfaro, 2001]. These modeling approaches address the modularity of a system.
Moreover, DES can be distinguished according to their time and signal interpretation (Fig. 3.6). Time-driven and event-driven systems can be distinguished. In *time-driven* event systems, the clock is incremented uniformly for Δ t time units. Updates to states (state changes) are made after the increment of a clock. The synchronization of events is realized by the clock ticks.

Event-driven approaches focus on the fact that the clock is initialized to zero and the times of occurrence of future events are determined. The clock is incremented to the time of the occurrence of the first events. The state of the system is updated to account for the fact that an event has occurred. In this way, times of inactivity can be skipped.

Definition 8 (Event)
An event ev_i is an action or spontaneous occurrence that is linked to a state change.

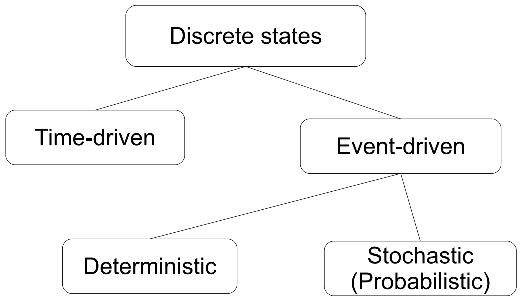


Figure 3.6: Classification of discrete states

DES can be defined being deterministic or stochastic [Lunze, 2006]. Events occur deterministically if successor states are determined based on the current state. In stochastic settings, the selection of the successor state responds to a probability distribution. Stochastic models include the randomness of real world phenomena. Those models incorporate a specific degree of uncertainty in the state change. Events occur within a specific degree of probability. Deterministic models can be interpreted as instances of stochastic (probabilistic) models with a probability of 100% passing over to a specific state.
In order to distinguish the time and signal interpretation, Figure 3.7 illustrates four cases. The first one (a) is the dense or continuous time and dense signal view in which signals occur continuously over time. Discretized time leads to case (b). In case (c), signals and time are discrete. In case (d), discrete events ev_i occur at specific time points t_i. This is called an event sequence Ev (Def. 9). The occurrence of an event is linked to a state change (transition) from st_k to st_{k+1} of the set St.

Definition 9 (Event sequence)
An event sequence $Ev = [ev_0, ev_1, ..., ev_6]$ is the succession of events ev_i.

There exist several timed models of timed and deterministic DES. The most established are presented here: *timed automata* and *timed Petri nets*.

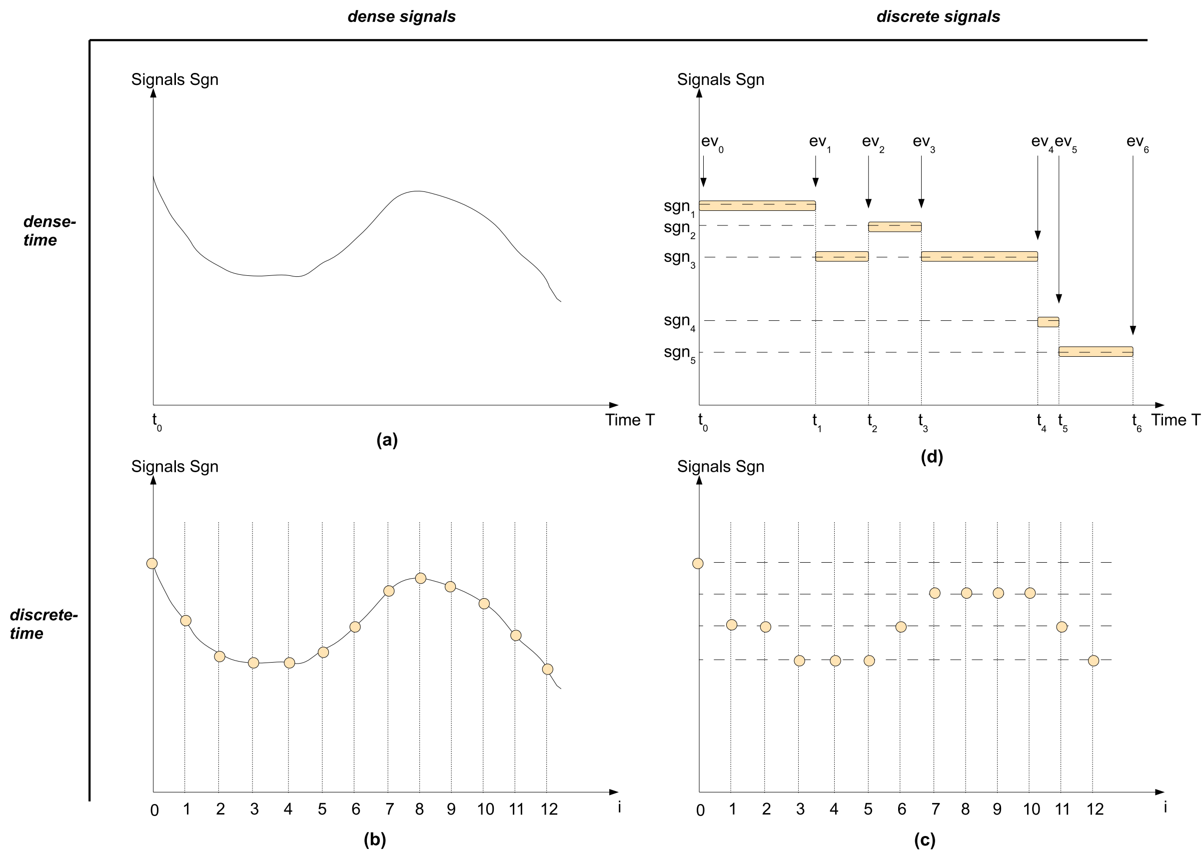


Figure 3.7: The concept of discrete events

3.3.1 Timed models

An extension of basic models [Mealy, 1955], [Moore, 1956] are timed models. In the basic models only the sequence of events is important. Additionally, timed models incorporate the information when events occur temporally. State changes occur in a discrete manner. Here, two well-established timed models with deterministic delays are considered: *timed automata* and *timed Petri nets*.

Timed automata

A Timed Automaton (TA) [Alur, 1994] enables to model and analyze the timed behavior of technical systems. The formalism of a TA has been extended in many ways [Waez, 2011]. The TA formalism incorporates *clocks* allowing descriptions of real-time behavior.

Since the formalism of TA is applied in this thesis (Chapter 4), several terms are presented. Events ev_i occur at specific points in time t_i. This fact is called a timed event σ_i (Def. 10). If no time information is assigned to an event, the event is called to be *untimed*.

Definition 10 (Timed event)
A timed event σ_i is a tuple (ev_i, t_i) where ev_i is an event occurring at point in time t_i.

The sequence of timed events is called a *timed word* (Def. 11).

Definition 11 (Timed word)
A timed word over a finite alphabet Σ is an infinite sequence of timed events (ev_1, t_1), (ev_2, t_2), (ev_3, t_3), ... where $ev_i \in Ev$ and $t_i \in T$.

The frequency enables the frequency of state changes in a timed word. The function occ maps an event ev_i to its occurrence in a timed word: occ: $\mathrm{Ev} \mapsto \mathbb{N}_0^+$. The set of timed words form a timed language (Def. 12).

Definition 12 (Timed language)
A timed language is a set of timed words over an alphabet Σ.

A type of automata that can interpret a timed language is a Timed (Büchi) Automaton (TBA). A TBA is a *Büchi automaton* [Büchi, 1962] accepting an infinite input alphabet and defines a set of real-valued variables called *clocks* [Alur, 1990] (Def. 13).

Definition 13 (Timed Büchi automaton)
A Timed (Büchi) automaton is a tuple (St, st_0, Σ, C, E, F):

- *St is a set of states with initial state $st_0 \in St$*
- *Σ is an alphabet*
- *C is a set of variables representing real-valued clocks*
- *$E \subseteq St \times \Sigma \times 2^C \times Cond(C) \times R(C) \times St$ is a set of edges with an edge e = (st, σ_i, $cond$, r, st'): $st, st' \in St$, $\sigma_i \in \Sigma$, cond as clock guards and r as the set of clocks to be reset*
- *$F \subseteq St$ is a set of accepting states*

In TA, two types of transitions may occur: *discrete transitions* (Def. 14) and *delay transitions* (Def. 15). A transition is a changeover from a state st to a state st′.

Definition 14 (Discrete transition)
A discrete transition is a transition $st = st_k(t_i) \xrightarrow[r]{\sigma_i,\, cond} st' = st_k(t_i')$, so that the guard cond *evaluates to true in st, $t_i' = t_i$ where all clocks in the set r are reset.*

Delay transitions result in delaying in st_k (Def. 15).

Definition 15 (Delay transition)
A delay transition is represented by $st = st_k(t_i) \xrightarrow{d} st' = st_k(t_i')$ where all clocks are incremented by d, *so that $t_i' = t_i + d$ holds.*

Time Petri nets

Classical Petri nets have been introduced by [Petri, 1962] and since then extended in many ways. Petri nets serve as formal basis for modeling simultaneous processes. The two main time extensions of Petri nets are Time Petri Nets (TPNs) [Merlin, 1974] and Timed Petri Nets [Ramchandani, 1974]. A Time Petri Net is a bipartite graph which associates a time interval to each transition whereas a Timed Petri Net uses durations as timing constraints.

Definition 16 (Time Petri Net)
A Time Petri Net is a tuple (Pla, Trans, •(.), (.)•, $mark_0$, I):

- *Pla is a finite set of places*
- *Trans is a finite set of transitions with Pla ∩ Trans = ∅*
- *•(.) ∈ $(\mathbb{N}^{Pla})^{Trans}$ is the backward incidence function*
- *(.)• ∈ $(\mathbb{N}^{Pla})^{Trans}$ is the forward incidence function*
- *$mark_0 \in \mathbb{N}^{Pla}$ is the initial marking*
- *I maps to each transition a time interval I: Trans $\mapsto U(\mathbb{Q}^+)$ with $u := [a, b]$ with $a < b \in \mathbb{Q}^+$*

Right as in timed automata, two different types of transitions may occur: *discrete* and *delay* transitions. The set of input places of a transition is denoted by •trans = {pla ∈ Pla | •trans(pla) > 0}, the set of output places is represented by trans• = {pla ∈ Pla | trans(pla)• > 0}. A transition is *enabled* in marking mark if mark ≥ •trans which is denoted by *trans* ∈ en(mark) which means that a transition is enabled if the number of tokens on each input place of transition *trans* is greater or equal than the valuation on the arc from this input place to the transition *trans*. An enabled transition is ready to fire if for time valuation v(*trans*) (as the time elapsed since the transition *trans* was enabled) holds that v(*trans*) ∈ $u := [a, b]$. The consequence of the firing of a transition is mark' = mark - •trans + trans•. For a detailed description of the semantics of Time Petri Nets, it is referred to [Bérard, 2005].

3.3.2 Stochastic timed models

Stochastic models in form of Probabilistic Timed Automata [Segala, 1995] or the Stochastic Petri Net (SPN) formalism [Natkin, 1980], [Molloy, 1981] enable the description of stochastics of processes as probabilistic occurrence of events. This is in accordance to the experience that in real-world technical systems, the changeover between two states is not always determined, but obeys environmental deviations. In probabilistic timed automata, the firing of a transition and the selection of a transition follows a probability distribution. The system state is mathematically interpreted as a random variable. In the probabilistic context, the probability of delays of transitions is not uniformly distributed, but obeys a probability distribution. For an introduction to Stochastic Petri Nets, it is referred to [Marsan, 1990].

3.3.3 Reachability and optimal reachability

The reachability of states is an essential property to be analyzed in TA [Alpern, 1985]. The reachability problem in timed transition systems can be formulated as symbolic reachability problem. In this context, the term *symbolic* means that the infinite state space is represented symbolically as so called zones [Bellman, 1957] (Def. 17).

Definition 17 (Zone)
A zone Z is a compact and symbolic representation of temporal constraints.

Zones can be interpreted as continuous-time symbolic abstractions with discrete-time constraints. The symbolic technique [Bellman, 1958], [Dill, 1990], [Henzinger, 1994] is computed by using a zone representation of temporal constraints. By this approach, the infinite domain of real-valued clocks can be represented as a finite data structure. An overview of the available data structures representing temporal constraints is given in [Behrmann, 2003, pages 19–26]. In the following, important terms for analyzing reachability problems with a symbolic representation are introduced for the application in this thesis (Section 5.1). A symbolic state $\mathrm{st}_k^{\mathrm{symb}}$ represents a state with respect to temporal constraints (Def. 18).

Definition 18 (Symbolic state)
A symbolic state st_k^{symb} is a tuple (h_k, Z_i) with vertex h_k and zone Z_i. Z is a conjunction of clock constraints and represents symbolically the set of all valid assignments t_i to a clock c with $t_i \in Z$.

The zone abstraction provides a mathematical basis for the calculation of successor symbolic states with respect to temporal constraints. In order to calculate the successor zone, two operations [Larsen, 2001] are relevant in this thesis: *projection* and *reset*. The projection is the state St being obtained if a delay occurs in a node (Def. 19).

Definition 19 (Projection)
A projection is represented by $Z^{\uparrow} := \{t_i{}' = t_i + d \mid d \in \mathbb{R}^{+}\}$ and $t_i, t_i' \in Z$.

A reset is given in Definition 20.

Definition 20 (Reset)
A reset is denoted by $\{r\}Z := \{\ t_i\ [r \rightarrow 0] \mid t_i \in Z_i\ \}$, r is the set of clocks to be reset.

Zones and the operations *projection* and *reset* can be efficiently implemented using Difference Bound Matrix (DBM) representation [Berthomieu, 1983]. Based on the precedent definitions, the reachability problem using this symbolic representation can be stated as follows (Def. 21).

Definition 21 (Reachability problem)
The reachability problem for:

- *a given timed automaton,*
- *the set of target symbolic states St_{tar}^{symb},*
- *and the set of reachable symbolic states St_{reach}^{symb}*

consists of deciding: $St_{tar}^{symb} \cap St_{reach}^{symb} \neq \emptyset$.

Deciding the reachability problem can be computed by Algorithm 1. The reachability problem in the context of timed automata is proved to be decidable by [Alur, 1990], [Alur, 2004]. It is formulated as emptiness problem deciding if the language accepted by the timed automaton is empty.

Data: Timed automaton ta(St^{symb}, Σ, $\text{st}_0^{\text{symb}}$, C, E, F), $\text{St}_{\text{reach}}^{\text{symb}} = (\text{h}_0, \text{t}_0 = 0)$, $\text{St}_{\text{tar}}^{\text{symb}} \subseteq St^{\text{symb}}$
Result: true if $\text{St}_{\text{tar}}^{\text{symb}} \cap \text{St}_{\text{reach}}^{\text{symb}} \neq \emptyset$

while $St_{tar}^{symb} \cap St_{reach}^{symb} = \emptyset$ **do**
  $\text{St}_{\text{reach}}^{\text{symb}} = \text{St}_{\text{reach}}^{\text{symb}} \cup \{(\text{h}', \text{t}_i') \in St^{\text{symb}\prime} \mid \exists(\text{h}, \text{t}_i)\text{: h} \wedge (\text{h}, \text{t}_i) \rightarrow (\text{h}', \text{t}_i) \vee (\text{h}, \text{t}_i) \rightarrow (\text{h}, \text{t}_i') \}$
  if $St_{tar}^{symb} \cap St_{reach}^{symb} \neq \emptyset$ **then**
    return true
  end
end
return false

Algorithm 1: Computing the reachability problem

In this context, an *accepting run* is a succession of symbolic states accepted by the timed automaton (Def. 22).

Definition 22 (Accepting run)
An (accepting) run is a finite succession of symbolic states in the form of acc = $st_0^{symb} \xrightarrow{\sigma_1} st_1^{symb} \xrightarrow{\sigma_2} \ldots \xrightarrow{\sigma_n} \ldots st_n^{symb}$
iff a state $st_k^{symb} \models cond$ and $t_i = (t_i')[r_i = 0]$ resulting in $inf(acc) \cap F \neq \emptyset$.

Considering optimality, the optimality of an accepting run with regard to a defined criterion must be shown. Two important classes of optimal reachability are known to be decidable: the *minimum-time reachability* problem [Niebert, 2000] and the *minimum-cost reachability* [Behrmann, 2001b] problem. In the context of minimum-cost reachability, the concept of *linearly priced timed automata* as an extension to ordinary timed automata was proposed. For so called priced clock regions, operations to compute clock resets and projections are defined. A branch-and-bound algorithm is provided in order to compute the cost-minimal reachability [Behrmann, 2001b,

page 10]. The problem of symbolic optimal reachability is solved using a zone-based approach, the computation is limited to certain problem sizes [Larsen, 2001], [Beyer, 2002, page 17]. For instance, a simple reachability analysis for the modular Test Bed tb_M presented in Subsection 8.3.2 results in a state-space explosion. The effort for representing all states in memory rises exponentially with the number of automation subsystems [Beyer, 2002, page 13]. Tests have shown that reachability analysis for medium size automation systems cannot be efficiently computed based exclusively on techniques introduced by [Larsen, 2001], [Behrmann, 2001b].
Using TAa, the optimization problem consists in finding the minimal-cost run reaching a specific target state. In [Alur, 2001], it is stated that the optimal run using only one clock can be computed in exponential time. Furthermore, since the available algorithms search in the symbolic state space of the product automaton, the proposed algorithms suffer from the state space explosion problem which leads to insufficient performance measures [Behrmann, 2001b].
The efficiency of an algorithm in solving the optimal reachability problems strongly depends on applied data structures for computation and the way models are constructed [Behrmann, 2001a]. Especially the number of clocks used in the model influences exponentially the computation time. Structure exploiting (hierarchy and modularity) and abstraction techniques [Beyer, 2002], [Dierks, 2006] need to be applied for efficient computation of optimal solutions as it is applied in this thesis (Chapter 6).
Since optimal reachability is formulated as constraint optimization problem in this thesis, an introduction is given next.

3.4 Constraint optimization

Selecting the best alternative among a set of possibilities belongs to the problem context of optimization. Optimization techniques and methods give analytical support for choosing the best alternative based on a given criterion. Solving optimization problems analytically requires the generation of optimization models. For an introduction to optimization see [Chong, 2013]. In general, an optimization problem has to respect constraints. A problem subject to constraints is therefore called a constraint optimization problem (Def. 23).

Definition 23 (Constraint optimization problem)
A Constraint Optimization Problem (COP) is a tuple COP = (Var, Dom, Cst(Var), obj)*:*

- *Var is a set of variables var $\in$ Var with a configuration denoted as Conf(Var)*
- *Dom is the set of domains with dom(var) $\in \mathbb{R}$ as the domain of a variable var*
- *Cst(Var) is the set of constraints over a set of variables Var*
- *obj is an objective function which assigns configurations of Conf(Var) to real values with obj: Conf(Var) $\mapsto \mathbb{R}$*

The problem is typically formalized in the following form requiring the objective function being minimized or maximized (Equ. 3.3).

$$\text{Minimize } obj(Conf(Var)) \tag{3.3}$$

subject to:
Cst(Var) $\leq a$ with $a \in \mathbb{R}$

The problem can be stated with $\text{conf}_{\text{opt}}(\text{Var}) \in \text{Conf(Var)}$ as global optimum [Papadimitriou, 1982, page 4] (Equ. 3.4):

$$\text{obj}(\text{conf}_{\text{opt}}(\text{Var})) \;\leq\; \text{obj}(\text{conf}_{\text{i}}(\text{Var}))\; \forall\; \text{obj}(\text{conf}_{\text{i}}(\text{Var})) \;\in\; \text{Conf}(\text{Var}) \tag{3.4}$$

If just Equation 3.5 holds, $\text{obj}(\text{conf}_{\text{opt}}(\text{Var}))$ is called a *local* optimum.

$$\text{obj}(\text{conf}_{\text{opt}}(\text{Var})) \;\leq\; \text{obj}(\text{conf}_{\text{i}}(\text{Var}))\; \forall\; \text{obj}(\text{conf}_{\text{i}}(\text{Var})) \;\in\; \text{Conf}_{\text{subset}}(\text{Var}) \;\subset\; \text{Conf}(\text{Var}) \tag{3.5}$$

In this thesis, *obj(Conf(Var))* is considered as a *linear function*. If Var $\in \mathbb{N}$ exclusively, then *Var* is the set of *discrete variables*. The optimization problem is then formulated over a finite set of discrete variables and the optimization problem is called a problem of Integer Linear Programming (ILP). The arising problem is denoted by the term *combinatorial problem* respectively *combinatorial optimization*.

3.4.1 Combinatorial optimization

Since the problem of this thesis is formulated as combinatorial optimization problem, an introduction to this special kind of optimization problems shall be given. Technical problems like single and multi-machine sequencing discussed in operations research literature [Graham, 1979] can be classified as combinatorial optimization problems. Combinatorial optimization comprises the optimization over discrete structures [Cook, 1998]. This requires other solution methods than optimizing over continuous variables. The combinatorial optimization asks "for an object from a finite, or possibly countably infinite set – typically an integer, set, permutation, or graph" [Papadimitriou, 1982, page 3].
An important class of combinatorial optimization is optimal scheduling of jobs on machines which consists of an assignment problem and a sequencing problem, in general. These problems belong to the class of scheduling problems since a set of tasks (jobs) have to be allocated to resources (machines). There exists many facets of machine scheduling problems, for a classification see [Pinto, 1998, page 435]. Scheduling problems have in common the assignment of tasks to machines, the sequencing of activities and the timing of tasks on resources [Reklaitis, 2000]. Time representation and the combinatorics of tasks are a major challenge in this field and in the scope of this thesis. Since the problem in this thesis is similar to multiple machine

sequencing, this well-studied problem class should serve as theoretical basis. However, there exist some differences. Classical machine scheduling consists of assigning and sequencing of jobs on machines. Both aspects are represented as constraints of the problem. Energy-efficient operation within unproductive phases comprises only the sequencing problem. There are no assignment issues to deal with. Nevertheless, the common ground for machine scheduling and the focus of this thesis is rooted in the objective to find optimal schedules in the set of multiple alternatives. Both problems are inherently exponential in complexity because of the combinatorics in scheduling alternatives.

Single machine scheduling

The single machine scheduling has been studied already in the 1950ies [Jackson, 1955] and consists of sequencing of weighted jobs on a single machine. The machine can process one job at a point in time t_i. Each job job_i has a length $length_i$ and a weight w_i and is composed of one or more operations. Jobs are subject to constraints. An important constraint is the precedence order of jobs that requires job job_i completed before job_k: $job_i \rightarrow job_k$. These precedence constraints can be graphically represented in acyclic directed precedence graphs. The schedule of the constrained jobs is optimal if a given criterion is minimized. An example is the minimization of the overall weighted completion time of all jobs on a single machine:

$$\text{Minimize } f(\tau) = \sum_{i=1}^{n} w_i \cdot \tau_i \tag{3.6}$$

with τ_i as completion time of job job_i.
The general problem has been shown to be NP-hard [Lawler, 1978]. For this kind of problem, extensive research has been accomplished providing procedures to solve or approximate single machine scheduling problems [Ambühl, 2011].

Multiple machine sequencing and assignment

Besides single machine scheduling, parallel machine scheduling has been widely studied [Chen, 1990]. In [Lawler, 1989, page 6], a machine scheduling problem as task that m machines have to process n jobs is defined. Machines compete for processing different jobs. If the relationship between jobs is unconstrained, there exist $(n!)^m$ schedules for n jobs and m machines. Even if the jobs are constrained by precedence constraints, [Ullman, 1975] shows that the scheduling of tasks on two processors subject to precedence constraints is NP-complete. In deterministic scheduling with multiple machines minimizing the make span, Johnson's rule is applicable [Johnson, 1954]. This is a method for finding the precedence of jobs on multiple machines in the optimal sequence.
Problems of combinatorial optimization have naturally an exponential complexity, so that several methods have been introduced to attenuate complexity issues.

3.4.2 Time representation in optimization models

As it was shown in Section 3.3, there exist two time representations: *continuous-time* and *discrete-time* interpretations. The time representation influences the formulation of the optimization model. Therefore models can be classified according to their time representation [Méndez, 2006]. Concerning the time interpretation, it depends on the specific problem to be modeled [Mouret, 2011, page 1038].

DISCRETE-TIME models use a fixed time grid for the ease of modeling. The time grid is a succession of elementary time intervals. Events can only occur at the boundaries of these elementary intervals and therefore, the time grid provides a natural way for synchronization points. The advantage of using a time grid is the support for reference of positioning of different operations and tasks [Kondili, 1993, page 216]. In this way, temporal constraints regarding interrelated tasks and operations can be easily formulated (precedence constraints). This facilitates setting up the optimization model if many temporal constraints are involved. "Discrete formulations have proved to be very efficient, adaptable and convenient for a wide variety of industrial applications, especially in those cases where a reasonable number of intervals is sufficient to obtain the desired problem representation" [Méndez, 2006, page 919]. Moreover "scheduling constraints have only to be monitored at specific and known time points, which reduces the problem complexity" [Méndez, 2006, page 918]. Care has to be exercised when choosing the number and granularity of time intervals since this has strong impacts on computational efficiency.

CONTINUOUS-TIME approaches rely on a dense-time model. Without a time grid, the formulation of constraints is more complicated than in the discrete-time approach. This significantly can increase the modeling effort. Furthermore, the problem size which "can be solved to optimality is usually smaller" than for discrete-time models in specific problem instances [Castro, 2005], [Floudas, 2004].

3.4.3 Selected solution procedures for optimization problems

Presented solution procedures are methods in order to solve optimization problems. The main solution methods for tagging optimal solutions are presented here. Complete enumeration and branch-and-bound methods belong to the decision tree methods. The brunch-and-cut method is a combination of a cutting planes algorithm and the branch-and-bound method. All three techniques are complete solution methods which guarantee to find the optimal solution if one exists.

Complete enumeration of solutions

Complete enumeration means to go through all possible solutions in order to identify the best and optimal alternative with regard to a predefined objective. In many real-world cases an exact algorithm cannot provide the optimal solution in reasonable time. Therefore, a lot of approaches have been proposed in order to attenuate the complexity problem. For an overview of state-of-the-art machine scheduling and energy-related optimization approaches, see [Artigues, 2011, page 3].

Branch-and-bound methods

In contrast to a complete enumeration of solutions, branch-and-bound methods are counted among the bounded enumeration methods. This meta method takes advantage of transforming the basic problem into a problem which is easier to solve [Mitten, 1970], [Bradley, 1977, page 289]. Branch-and-bound procedures strongly rely on decomposition. The ability to decompose a problem allows to benefit from parallel processing. This is reflected in significant reduction of computational time. It has been evaluated that biggest savings in computational runtime can be achieved by decomposition techniques [Harjunkoski, 2002]. Decomposition of discrete structures that is classically a graph-theoretical problem is described in [Möhring, 1984]. Decomposition means to split up a problem into smaller independent subproblems. This method is oriented at the divide and conquer paradigm introduced by nested dissection of [George, 1973] and follows the application of a specific search technique. The decomposition and search technique is exemplified using the following setting. Assuming the objective function *obj* for three given subsystems (Equ. 3.7).

$$
\begin{aligned}
\text{Minimize obj}(\text{var}) = & \\
& \sum_{p=1}^{k} \mathrm{w_p} \cdot \mathrm{var_p} + \\
& \sum_{p=k+1}^{m} \mathrm{w_p} \cdot \mathrm{var_p} + \\
& \sum_{p=m+1}^{n} \mathrm{w_p} \cdot \mathrm{var_p} \qquad (3.7) \\
\text{subject to} & \\
& \mathrm{a}_p \leq \mathrm{var}_p \leq \mathrm{b}_p (p = 0, 1, 2, ..., k) \\
& \mathrm{a}_p \leq \mathrm{var}_p \leq \mathrm{b}_p (p = k+1, 2, ..., m) \\
& \mathrm{a}_p \leq \mathrm{var}_p \leq \mathrm{b}_p (p = m+1, 2, ..., n)
\end{aligned}
$$

with $\mathrm{w_p}$ as weight of variable $\mathrm{var_p}$.

The objective function obj(var) for the system can be split up into three separate subproblems, since the variables $\mathrm{var}_0, \mathrm{var}_1, ..., \mathrm{var}_k$, the variables $\mathrm{var}_{k+1}, \mathrm{var}_{k+2}, ..., \mathrm{var}_m$, and the variables $\mathrm{var}_{m+1}, \mathrm{var}_{m+2}, ..., \mathrm{var}_n$ do not appear in common constraints. This fact has implications on computational savings, since linear programs are sensitive to the number of constraints *constr*, growing proportionally to $constr^3$. Given the number of subproblems s and each subproblem contains a set of $\frac{1}{s} \cdot n$ constraints, leads to a computational effort of $(\frac{\text{constr}}{s})^3$ per subproblem. If the subproblems can be solved independently, the overall effort is reduced to $s \cdot (\frac{\text{constr}}{s})^3 = \frac{1}{s^2} \cdot \text{constr}^3$ [Bradley, 1977].

The decomposition of an optimization problem into smaller subproblems is then used for branching. This term expresses the procedure to decompose the underlying problem into several subproblems:

$$\text{Sol}(\text{prob}_0) = \bigcup_{i=1}^{n} \text{Sol}(\text{prob}_i) \tag{3.8}$$

with $\text{Sol}(\text{prob}_i)$ as the set of valid solutions of problem prob_i. The intersection of a solution pair of problems is meant to be the empty set: $\text{Sol}(\text{prob}_i) \cap \text{Sol}(\text{prob}_j) = \emptyset \; \forall \; i \neq j$. The decomposition of the problem space into smaller subproblems can be represented as decision tree which allows for branching.

The second part of the branch-and-bound method serves as bounding in the decision tree. Bounds are used for deciding if a subproblem must be investigated or not. In the underlying problem, a lower bound of the objective function can be given. In the course of the branch-and-bound procedure, the best known valid solution is the actual lower bound of the objective function value. This is called the *global lower bound* $\underline{lb}$. Moreover, there exists a *local upper bound* $\overline{lb}_i$ of the subproblems. The local upper bound can be efficiently identified by relaxation which means to omit constraints so that holds $\text{Sol}(\text{prob}_i) \subseteq \text{Sol}(\text{prob}_i^{relax})$.
Three cases can be distinguished [Domschke, 2011, page 134]:

- Case 1 ($\overline{lb}_i \geq \underline{lb}$):
  The optimal solution of the subproblem $\overline{lb}_i$ is not better than the already known actual optimal solution $\underline{lb}$.

- Case 2 ($\underline{lb}_i < \underline{lb}$):
  A new valid solution $\text{sol}(\text{prob}_i^{relax})$ for $\text{sol}(\text{prob}_i)$ is found, so that the actual (global) lower bound is replaced by $\underline{lb} = \overline{lb}_i$.

- Case 3 ($\text{Sol}(\text{prob}_i^{relax}) = \emptyset$):
  No solution $\text{sol}(\text{prob}_i^{relax})$ can be found which results in $\text{Sol}(\text{prob}_i) = \emptyset$.

This method can be applied to job shop scheduling on parallel machines as shown in [Chen, 1999]. [Salem, 2000] presents the efficient computation of the minimum of the maximum completion time based on a branch-and-bound algorithm. Although this procedure is often used in operations research, this meta method can be applied whenever decomposition is applicable and lower bounds for subproblems can be specified.

Branch-and-cut methods

Combining pure branch-and-bound procedures with cutting plane approaches [Gomory, 1958], [Benders, 1962] results in *branch-and-cut* methods. For integer programming, in a first step, cutting planes are derived for being complemented in a second step by a branch-and-bound method. For further details, see [Müller-Merbach, 1973].

Heuristics

Since complexity is a major challenge in scheduling problems, heuristics are often applied. Especially when multiple objectives are addressed, heuristics provide a way to find efficiently suboptimal solutions. The optimality criterion is abandoned for better computational performance [Harjunkoski, 2001]. The challenge consists in developing heuristics which provide high quality solutions by simultaneously speeding up computational performance. For more details, see [Domschke, 2011, pages 129–133].

3.5 Summary

In this chapter, an overview of technical systems which are in the focus of this thesis has been given. Those manufacturing systems can be classified according to the installed machine tools or machines, the functional structure of the system or the control structure of the system (Section 3.1). The different ways to classify automation systems provides alternatives for modeling the subsystems described in Part II of this thesis.
Systems theory (Section 3.2) provides the conceptual basis and elements for describing the structure and behavior of manufacturing systems in an abstract way. Events that denote state changes play an important role in DES (Section 3.3). Two main models for representing the structural and behavioral aspects of DES have been introduced for dealing with temporal characteristics of technical systems. The analytical model of timed automata is adopted in this thesis (Chapter 4). In this context, stochastic aspects are omitted since the system properties, the initial states and the sequence of events are assumed to be determined in the problem context of this thesis.
Providing the theoretical background for identification of optimal strategies for automation systems within unproductive phases, the theory of constraint optimization (Section 3.4) has been introduced. Combinatorial optimization serves as theoretical basis for setting up the optimization problem in Part II. Identifying the optimal strategy for unproductive phases is stated as a precedence-constrained sequencing problem in Section 5.2. The procedures provided for

constraint optimization (Subsection 3.4.3) serve basis for the proposition of a procedure that helps to identify the optimal strategy (Chapter 6).

Part II

Approach for energy-efficient operation within unproductive phases

Chapter 4

Automaton-based system model

The structural and behavioral description of an automation system and of its subsystems is the central aspect of this chapter. An analytical model for automation systems based on automata theory (Section 3.3) is proposed which can serve as basis for the computation of strategies.
This chapter is structured as follows. The analytical model is derived from the conceptional elements of Section 4.1. Starting at conceptual considerations, the modularity paradigm of automation systems (Subsection 4.1.1) and the behavioral perspective of automation systems (Subsection 4.1.2) is presented. A generic model of automation systems is derived from those conceptional elements (Section 4.2). In Subsection 4.2.1, the representation of structural information in form of subsystem interdependencies is delineated. The temporal and energetical perspective regarding automation subsystems is introduced in Subsection 4.2.2. The structural and behavioral description is used when passing to the computation of strategies in Chapter 5. In addition, the aggregation of subsystem models is illustrated in Subsection 4.2.3.

4.1 Conceptional elements

The system model to be introduced addresses the structural and behavioral aspects of industrial automation systems. According to systems theory (Section 3.2), automation systems are divided into subsystems expressed by the structure of today's automation systems (Section 3.1). The control and physical structure of an automation system determines its degree of modularity and hierarchy.

4.1.1 Structural view: Modular structure of automation systems

Modularity arises from decomposing a system into modules or subsystems (Section 3.2). Each subsystem is interpreted as being equipped with a dedicated control unit, automates a part of the production process, and interacts with other subsystems via specific interfaces (Fig. 4.1). In this way, the automation task is subdivided into subtasks according to the existing subsystems.

In modular automation systems, subsystems can be linked on control level communicating dependencies on physical production level. Consequently, the control level maps the physical production level.

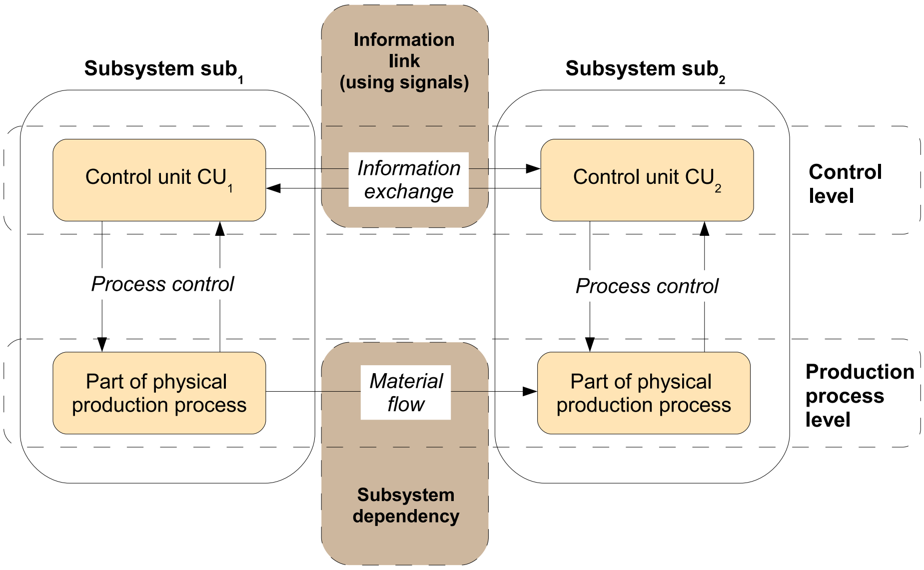


Figure 4.1: Information and production level of two automation subsystems

Modularity causes the need for communicating subsystem dependencies visualized in Figure 4.2 (Def. 24).

Definition 24 (Subsystem dependency)
A subsystem dependency sd_{ik} is a relationship between two Subsystems sub_i and sub_k caused by material flow, safety reasons, etc. represented as information link between this pair of subsystems. A subsystem dependency constrains the behavioral freedom of involved subsystems.

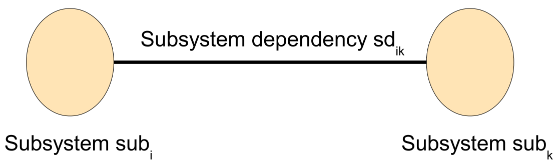


Figure 4.2: Subsystem sub_i and Subsystem sub_k with subsystem dependency sd_{ik}

The cause of subsystem dependencies may be various. The main specificities are mentioned here. *Process-related dependencies* between subsystems arise if subsystems control parts of the overall physical production process. Materials are required to be exchanged between subsystems. This material-flow dependency has implications on the behavioral freedom.

Example 1
Assuming two subsystems in a production line, a transportation *subsystem and a* handling *subsystem. The transportation subsystem supplies the handling subsystem with items which is a process-related dependency. The handling subsystem requires material input from the transportation subsystem. This dependency is represented as an information link using signals. This process-related dependency has to be considered when addressing strategies for unproductive phases.*

Safety-related dependencies exist if the state of a subsystem is influenced by the state of another subsystem affecting safety issues.

Example 2
For instance, functional safety is implemented in two subsystems of the system machine tool *(sub_1: sliding door, sub_2: lathe). Before the closed door can be opened (first state change), the running lathe needs to be turned off (second state change). The second state change in sub_2 needs to be triggered before the first state change in sub_1 can take place.*

Communication-related dependencies can affect state changes of subsystems as well. This kind of dependency between two subsystems frequently needs to be considered in the context of software boot up processes of subsystems.

Example 3
An example is the communication link between two control units (sub_1 with control unit CU_1, sub_2 with control unit CU_2). Assuming both, CU_1 and CU_2, are electronically switched off. CU_1 expects CU_2 as available communication partner as soon as the control program of CU_1 is executing. With this setting, a CU_1 with running control program and CU_2 which is still powered off, causes a fault situation. Therefore this communication-related dependency needs to be considered for strategies in unproductive phases as well.

In order to get an impression of the extent of dependencies between subsystems in a FMS, the visual representation of an automation system with its structure and dependencies is given as UML component diagram (Fig. 4.3). This figure illustrates the existing components as *subsystems* and the bidirectional dependencies between subsystems as *ports* within a system (Test Bed tb_M, Subsection 8.3.2). Information regarding dependencies between components is exchanged via ports which represent interfaces. The lollipop representation is used for inputs (sockets) and outputs (balls) of components. In Figure 4.3, subsystems are grouped according to their function in the system (for instance *Filling*).
This structural view does not provide information about the internal behavior of subsystems. Consequently, this view has to be supplemented by a behavioral view showing the internals of subsystems.

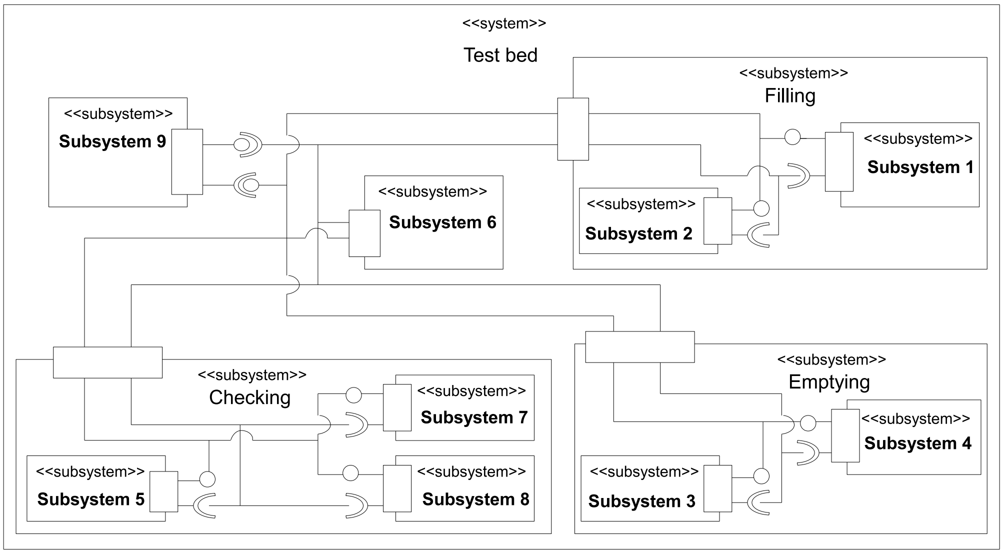


Figure 4.3: Component diagram of hierarchical subsystems

4.1.2 Behavioral view: Energetical behavior of automation subsystems

The structural view of Subsection 4.1.1 needs to be supplemented by a description of subsystem internals. The internal behavior of a subsystem is determined by modes and mode transitions (Def. 25). Regarding mode classification, productiveness and unproductiveness has to be distinguished.

Definition 25 (Mode and transition)
A mode m_k^i is a distinguishable status of a subsystem which can be assessed according to its input power pc. It can be distinguished between modes representing productiveness (productive mode) and modes representing unproductiveness (unproductive mode) in a subsystem. A productive mode is a status in which manufacturing processes take place (handling or transportation of pieces, for instance). Unproductive modes are characteristics in which a subsystem does not produce (standby or off, for example). A subsystem can pass over between mode m_k^i and m_{i+1} using transitions which, in general, takes time.

Modes and mode transitions are illustrated as UML state chart in Figure 4.4. The subsystem behavior is determined by the relationship between modes and mode transitions. The labels on mode transitions between modes respect the event-guard-action pattern (ev_i g_j / a_k). Events ev_i (Def. 8) are used for representing mode changes. Guards g_j are conditions for mode changes (conditions in Def. 13) and actions a_k model operations after mode changes occur (clock resets in Def. 13).
The used UML state charts are based on Harel automata [Harel, 1987] providing a tool for describing modes, transitions, guards, time aspects and activities to be taken in a mode or actions being executed while a transition fires. Activities in a mode are time-dependent and

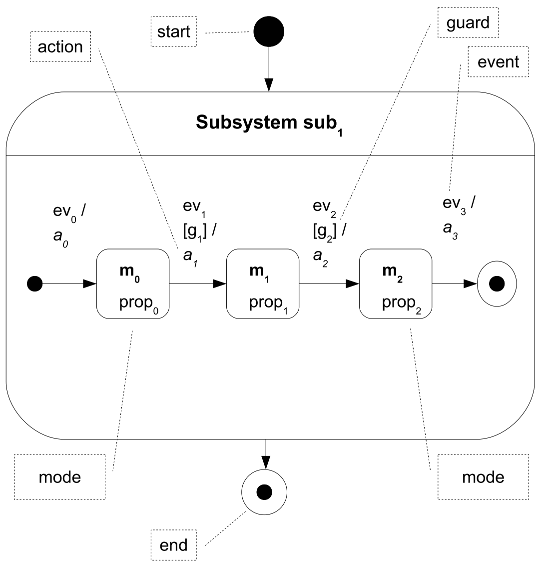


Figure 4.4: State chart elements used for modeling the internals of automation Subsystem sub_1

are characteristic of Moore automata [Moore, 1956], whereas events and actions on transitions are timeless and are a characteristic of Mealy automata [Mealy, 1955]. Concurrent behavior is representable by orthogonal subsystems. Note that guards g_j are enabling conditions which cannot force a transition to be taken. The modes have specific properties (specific input power of a subsystem while being in a mode). Start modes are denoted by filled bullets. End modes are denoted by filled bullets framed by a circle.
The perspective of productive modes and unproductive modes is illustrated in Figure 4.5. The system is modularly composed by three subsystems sub_1, sub_2, and sub_3. Each subsystem encapsulates productive and unproductive modes with specific input powers ([W] labels). The transition from one mode to another can be time-dependent (d labels at transitions). The conceptual elements of this section need to be represented formally to enable analytical evaluation. On this account, structural information and mode-related aspects are mapped to an analytical model in the next section.

4.2 Analytical model

The structural and behavioral aspects of an automation system are presented formally in this section. These aspects are relevant for addressing the energetical point of view within unproductive phases. First, a formalism for representing structural aspects is given in Subsection 4.2.1. Secondly, the internal behavior of an automation subsystem is added to this perspective in Subsection 4.2.2. Automata theory (Subsection 3.3.1) provides a flexible and well-founded

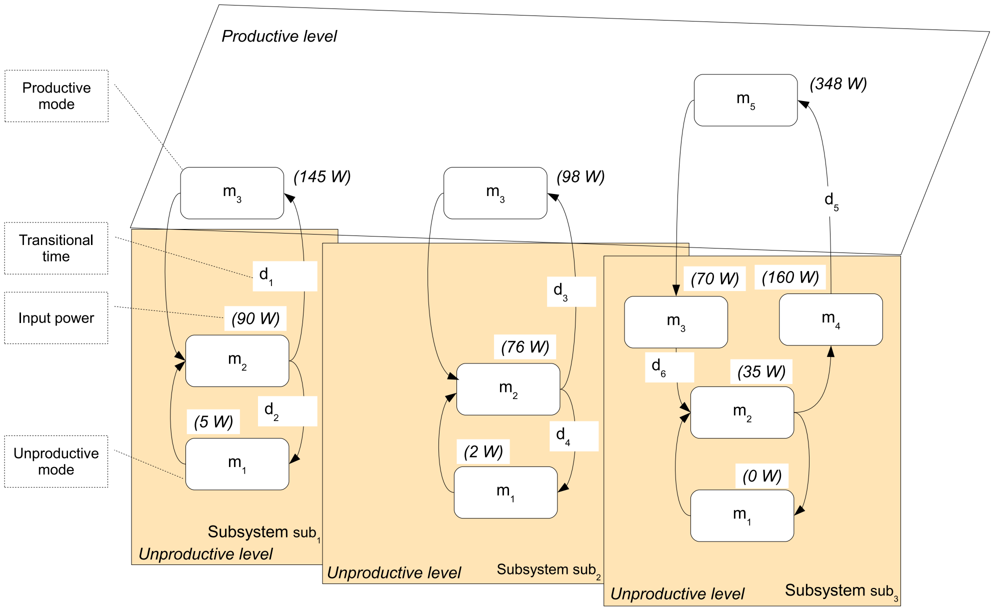


Figure 4.5: Productive and unproductive modes of Subsystems sub_1, sub_2, and sub_3

mathematical basis for this purpose [Mechs, 2012a].

4.2.1 Network of automation subsystems

Modularity is a key concept of industrial automation. This needs to be appropriately supported by a descriptive model. Subsystems require a way to communicate arising dependencies caused by modularity. Automation subsystems are represented as networks of finite automata (Def. 26) using shared variables to communicate dependencies. Values of a shared variables represent the current mode of a subsystem.

Definition 26 (Untimed network of automation subsystems)
A network of (untimed) automation subsystems is a tuple $NAS = (Sub, SV)$ with $sub_i = (M^i, \Sigma^i, m_0^i, Ed^i)$ as a Subsystem $sub_i \in Sub$. The set of shared variables is represented by SV. A shared variable referring to Subsystem sub_i is given by $sv_i \in SV$.

- *M^i is a finite set of modes*
- *An alphabet Σ^i*
- *The initial operating mode $m_0^i \in M^i$*
- *$sv_i \in SV$ as a shared variable with $sv_i^{initial} = m_0^i$*
- *A set of assignments to the shared variable $A^i : sv_i \mapsto m^i$*

- *A set of guards expressing subsystem dependencies with* $G^i(SV) := g^i(sv_l) \wedge ... \wedge g^i(sv_n)$ *in the form of* $g^i := sv_l \sim m_j$ *with* $\sim \in \{==, \neq\}$ *and* $g^i_{neq} := sv_l \neq m_j$, $g^i_{eq} := sv_l == m_j$, $SV \setminus sv_i$
- *A finite set of edges of a subsystem is denoted by* $Ed^i \subseteq M^i \times \Sigma^i \times A^i(sv_i) \times G^i(SV) \times M^i$

Subsystem dependencies (Section 4.1 and Def. 24) are considered as guards (logical constraints) referring to shared variables. Shared variables can be set (assigned) by one subsystem and can be read by other subsystems. In this way, dependencies are transformed into the semantics of guarded transitions using shared variables and being communicated between subsystems. In this thesis, shared variables are applied in guards as enabling conditions.
The formalism of Definition 26 can be graphically represented in form of UML state charts. An example is shown in Figure 4.6. The system contains of Subsystem sub_1 which is orthogonal to Subsystem sub_2 (dashed line between both subsystems). The different modes are denoted as m^i_k. If a transition is taken (upon a timed event σ_j), the shared variable sv_i of subsystem sub_i is updated communicating the current mode of a subsystem to exterior subsystems.
Besides these updates, guards on shared variables are used to express subsystem dependencies between both subsystems. For instance, in sub_2, the transition from m_2 to m_3 is guarded by $\mathrm{sv}_1 \neq \mathrm{m}_0$. This notation expresses the fact that the transition from m_2 to m_3 is enabled if $\mathrm{sv}_1 \neq \mathrm{m}_0$ which corresponds to the formulation that sub_1 must not be in mode m_0 while taking this transition. Required precedence orders of modes may be expressed in this way.

4.2.2 Temporal and energetical model of automation subsystems

Besides describing structural information of the automation system, the behavioral aspects are added in form of time and energy information. Additions to the model in Subsection 4.2.1 are made regarding temporal changeovers between modes and input powers of modes. A delay in mode m^i_k is referenced as d (Def. 27).

Definition 27 (Mode delay)
The mode delay d is defined as $(m^i, t_i) \xrightarrow{d} (m^i, t_i')$ *with* $t_i' = t_i + d$*. This corresponds to a delay transition as defined in Definition 15.*

The energetical aspect is considered in form of the input power of mode is considered using constant rates while delaying in a mode (Def. 28).

Definition 28 (Input power of a mode)
The input power of a mode is represented by a constant rate pc. This constant rate represents the energy demand per time unit of the underlying physical process while staying in that mode.

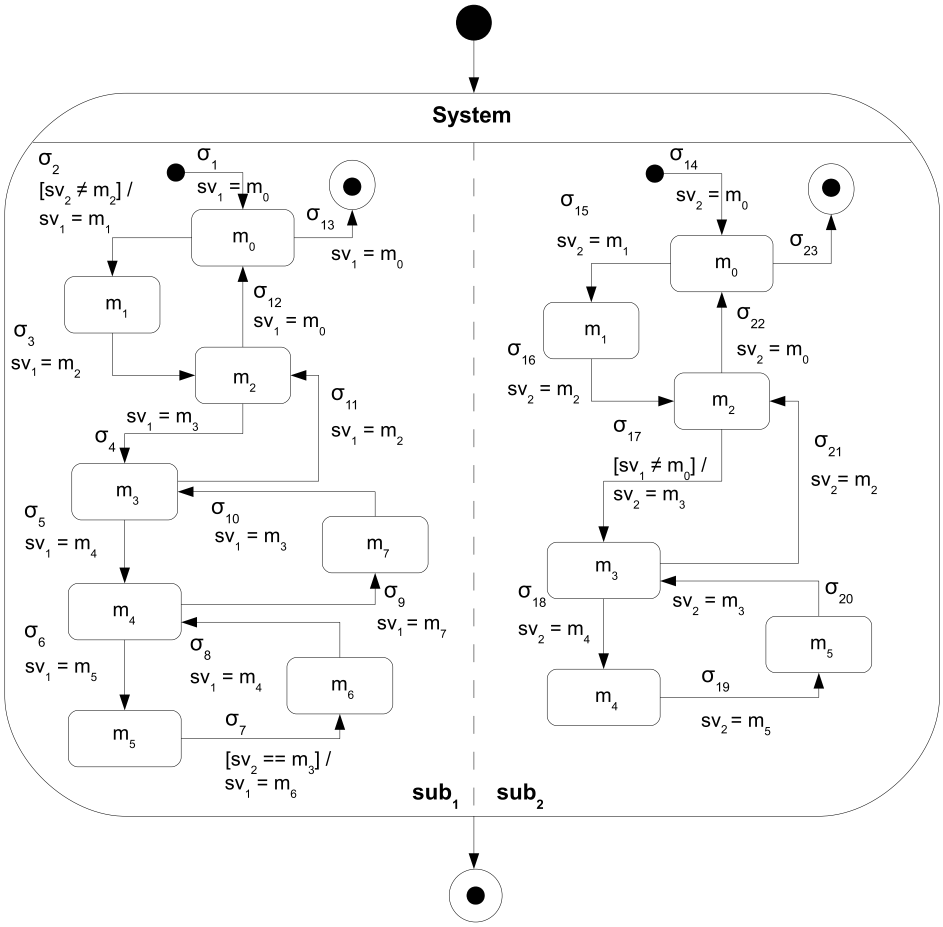


Figure 4.6: Structural aspects of two orthogonal Subsystems sub_1 and sub_2

In addition to Definition 26, the temporal and energetical aspects of an automation subsystem are added to the tuple NAS = (Sub, SV) with $\text{sub}_i = (\text{M}^i, \Sigma^i, \text{m}_0^i, \text{c}, \text{Inv}^i, \text{Ed}^i, \Omega^i)$ as a Subsystem $\text{sub}_i \in$ Sub. The set of shared variables is denoted by SV with $\text{sv}_i \in$ SV as shared variable of Subsystem sub_i (Def. 29).

Definition 29 (Networked timed automation subsystem)

The temporal transition and the energetical operating behavior of a single automation subsystem is represented by a networked priced timed automaton $sub_i = (M^i, \Sigma^i, m_0^i, c, Inv^i, Ed^i, \Omega^i, SV)$:

- *M^i is a finite set of modes consisting of a set of modes*
- *An alphabet Σ^i*
- *The initial operating mode $m_0^i \in M^i$*
- *$sv_i \in SV$ as a shared variable with $sv_i^{initial} = m_0^i$*

- *A set of assignments to the shared variable* $A^i(sv_i) : sv_i \rightarrow m^i$
- *A set of guards on shared variables* $G^i(SV) := g^i(sv_l) \wedge ... \wedge g^i(sv_n)$ *in the form of* $g^i := sv_l \sim m_j$ *with* $\sim \in \{==, \neq\}$ *and* $g^i_{neq} := sv_l \neq m_j$, $g^i_{eq} := sv_l == m_j$, $SV \setminus sv_i$
- *A time variable* $c \in \mathbb{R}_0^+$ *called clock*
- *A set of invariants Inv:* $M \mapsto F(c)$ *in the form of* $f := c \sim t$ *and* $\sim \in \{<, \leq\}$, *and* $t \in \mathbb{R}^+$
- *A set of clock guards* $G^i(c)$: $g^i := c \sim t$ *and* $\sim \in \{<, \leq, =, \geq, >\}$
- *A set of clock resets* $R^i(c)$ *with* $r^i := c = 0$
- *A finite set of transitions is denoted by* $Ed^i \subseteq M^i \times \Sigma^i \times G^i(c) \times R^i(c) \times A^i(sv_i) \times G^i(SV) \times M^i$
- *A cost function* Ω^i: $M^i \mapsto \mathbb{N}_0^+$ *mapping to each mode or transition a constant input power value* pc_i

In addition to Figure 4.6, Figure 4.7 incorporates time and input power values in modes of Subsystem sub_1 and Subsystem sub_2. Besides updates and guards on shared variables, updates (resets) on clock variables and clock guards are used in order to express the timed behavior of the subsystems. Moreover, each mode features the input power *pc*.
The semantics of a timed automation subsystem is analog to that of a timed automaton [Alur, 1994] where two different transition types occur: discrete transitions and delay transitions (Subsection 3.3.1). Discrete transitions are caused by switching commands (Def. 30).

Definition 30 (Switching command)
A changeover from a mode m^i_k *to a mode* $m^i_k{}'$ *(discrete transition) evoked by a timed event* σ^i *which is called a switching command.*

Discrete and delay transitions may be provided with cost information [Alur, 2001], [Behrmann, 2001b], [Behrmann, 2006]. Cost-annotated delay transitions are defined as $(m^i, t) \xrightarrow[pc]{d} (m^i, t')$ with energy demand equal to $d \cdot pc(m^i)$ and $m^i \in M^i$.

4.2.3 Product of automation subsystems

The product of automation subsystems is semantically equal to the orthogonal representation of automation subsystems in Figure 4.7. The product automaton corresponds to the composition of a pair of automation subsystems. The product of timed automation subsystems is given in Definition 31.

Definition 31 (Product of networked automation subsystems)
Given two timed automation Subsystems sub_1, sub_2, *and its network NAS*

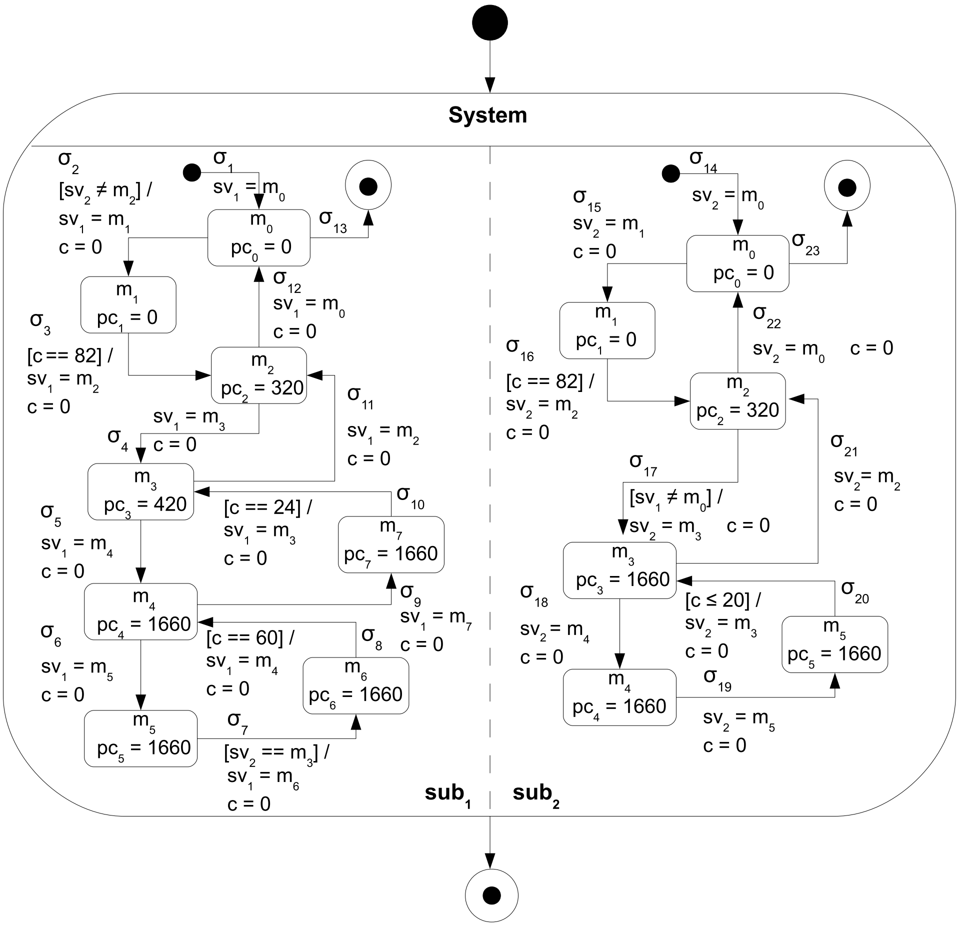


Figure 4.7: Temporal and energetical aspects of two orthogonal Subsystems sub_1 and sub_2

- $sub_1 = (M^1, \Sigma^1, m_0^1, c, Inv^1, Ed^1, \Omega^1)$,
- $sub_2 = (M^2, \Sigma^2, m_0^2, c, Inv^2, Ed^2, \Omega^2)$,
- *NAS = (Subs, SV) with* $sub_1, sub_2 \in Subs$ *and* $sv_1, sv_2 \in SV$

the product of $sub_1 \parallel sub_2$ *is represented as*
timed automation Subsystem $sub_{product} = (M^1 \times M^2, \Sigma^1 \cup \Sigma^2, m_0^1 \times m_0^2, c, Inv, Ed)$ *with:*

- $\Sigma^1 \cap \Sigma^2 = \emptyset$
- $Inv(m_i^1, m_k^2) = Inv^1(m_i^1) \wedge Inv^2(m_k^2)$ *and*
- $\Omega(m_i^1, m_k^2) = \Omega^1(m_i^1) \wedge \Omega^2(m_k^2)$ *and*
- *an edge* $((m_i^1, m_k^2), \sigma, G(C), R(C), A(SV), G(SV), (m_j^1, m_l^2)) \in Ed$ *exists iff:*

- $\sigma \in \Sigma^1 \setminus \Sigma^2$, $m_k^2 = m_l^2$ *with* $(m_i^1, \sigma^1, r^1(c), g^1(c), a^1(sv_1), g^1(sv_2), m_j^1)$ *and*

$$g^1(sv_2) := \begin{cases} sv_2 == m_k^2 \\ sv_2 \neq m_p^2 \text{ with } m_p^2 \neq m_k^2 \end{cases}$$

- $\sigma \in \Sigma^2 \setminus \Sigma^1$, $m_i^1 = m_j^1$ *with* $(m_k^2, \sigma^2, r^2(c), g^2(c), a^2(sv_2), g^2(sv_1), m_l^2)$ *and*

$$g^2(sv_1) := \begin{cases} sv_1 == m_i^1 \\ sv_1 \neq m_d^1 \text{ with } m_d^1 \neq m_i^1 \end{cases}$$

As an example, Figure 4.8 shows the composition of two interrelated timed automation Subsystems sub_1 and sub_2. The subsystem modes are represented by m_i. Each mode has a specific input power pc. One clock c is used to reference the temporal aspects of transitions using guards (*[c == 5]* at outgoing transition from mode m_2 to mode m_1 in sub_2). At each transition the clock is reset (*c = 0*). Two shared variables sv_1 and sv_2 are used for representing the current mode of sub_1 respectively sub_2. These variables are updated at each transition (for example: $\mathrm{sv}_2 = \mathrm{m}_1$ at transition from mode m_2 to mode m_1 in sub_2). Structural aspects like dependencies between sub_1 and sub_2 are expressed by guards. For instance, in Subsystem sub_1, the transition from mode m_1 to mode m_2 is labeled with a subsystem dependency (*[$sv_2 \neq m_1$]*). This guard expresses a sequential order of modes: At the moment, the transition from mode m_1 to mode m_2 in Subsystem sub_1 can be taken, Subsystem sub_2 needs to rest in another mode than mode m_1.

4.3 Summary

In this chapter, an automaton-based system model describing the temporal, energetical, and structural aspects of an automation system is presented. This formalism enables the description of modular automation subsystems in a natural way. At each point in time, each subsystem is in a specific mode while a certain amount of energy has to be dedicated for staying in those modes. Real-world phenomena show that the changeover between modes is time-dependent. This can be represented in form of clock guards. Interrelated subsystems communicate dependencies over shared variables enabling to map the modularity of automation systems to the system model proposed in this chapter. The system model that is formulated in form of networked timed automata serves as analytical fundamentals for the generation of strategies in the next chapter. This makes allowance for evaluating energy savings potentials in early engineering and design phases of an automation system.

While generating the product automaton composed of the automation subsystems, a major drawback has to be faced that is called the *state space explosion*. Since the product automaton is based on the Cartesian product of modes respectively edges, the state space of the product

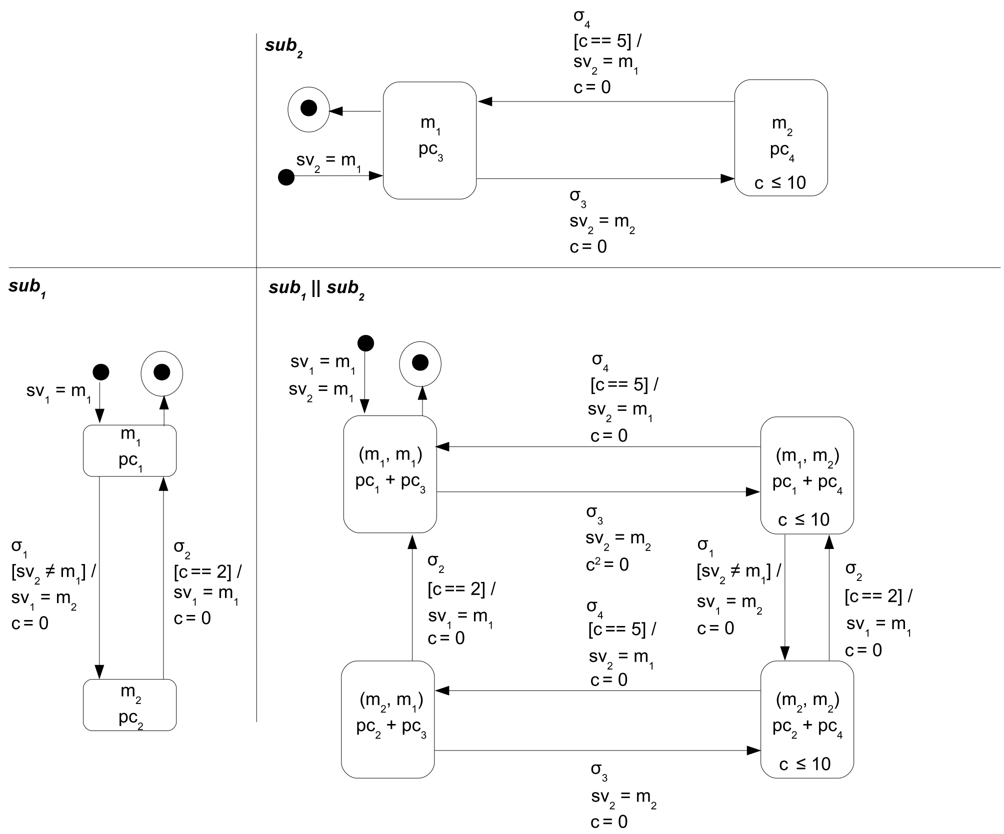


Figure 4.8: Product automaton of sub_1 and sub_2

automaton increases exponentially with the number of subsystems.

Therefore, the structure (modularity) of the automation system needs to be used to compute strategies for unproductive phases. This is focus of the remaining chapters of Part II. The derivation of strategies based on the proposed system model is the central aspect of Chapter 5. Identifying the optimal strategy within the set of alternative strategies is a major challenge which is discussed in Chapter 6.

Chapter 5

Strategies for maximizing energy efficiency

Modeling structural and behavioral information of automation subsystems relevant for unproductive phases has been focus of Chapter 4. The objective of this chapter is the energetical or temporal aspect of strategies derived from this structural and behavioral information. Energetical aspects are considered in form of *strategies* minimizing the energy input of an automation system within a certain time interval which increases the energy efficiency (Section 5.1). In order to enable the optimization of the energy input based on strategies, the switching sequence and strategy optimization problem is stated (Section 5.2) .

5.1 Switching sequences and strategies

In this section, the central term *strategy* is clarified. Thereby, the background for stating the optimization problem is given. First, the elements of strategies are defined which are named *switching sequences* (Subsection 5.1.1). Within a single subsystem, alternative switching sequences can be feasible which is addressed by Subsection 5.1.2. Strategies are composed of switching sequences of different subsystems. Different perspectives on strategies are presented in Subsection 5.1.3.

5.1.1 A switching sequence within a subsystem

Based on the algorithm for symbolic reachability (Algo. 1) presented in Section 3.3.3, switching sequences with predefined initial and target modes as input parameters can be identified for a given time period and a minimum number of switching commands [Mechs, 2012c]. The succession and sequencing of symbolic states is denoted as switching sequence (Def. 32).

Definition 32 (Switching sequence)
A switching sequence seq_k^i of a Subsystem sub_i is the finite succession of modes in the form of

$(m_0, Z_0) \xrightarrow{d_1} (m_0, Z_1) \xrightarrow{\sigma_1} (m_1, Z_1) \xrightarrow{d_2} (m_1, Z_2) \xrightarrow{\sigma_2} \ldots \xrightarrow{\sigma_n} (m_{tar}, Z_{tar})$ with m_{tar} as target mode, m_0 as initial mode, and Z_{tar} as target zone, $m_l \in M$ and $Z_j \neq \emptyset$.

Switching is regarded as costly and may reduce the lifetime of switched subsystems. The negative effects of switching requires to avoid unnecessary switching. A switching sequence has to comply with the minimum-switch property (Def. 33).

Definition 33 (Minimum-switch property)
The minimum-switch property requires the sequence seq_k^i not to contain one mode more than twice. Complying with the minimum-switch property means that the frequency of the event ev_i in sequence seq_k^i is less or equal to two $occ(ev_i) \leq 2$.

For each switching sequence, the minimum energy demand can be determined (Def. 34).

Definition 34 (Minimum energy demand of a switching sequence)
The function $e_{seq} : Seq^i \mapsto \mathbb{R}_0^+$ assigns to each switching sequence seq_k^i of a Subsystem sub_i an energy demand value that represents the achievable minimum energy demand of the switching sequence. The energy demand of a switching sequence is calculated by $e(seq_k^i) := d_1 \cdot pc(m_0) + d_2 \cdot pc(m_1) + \ldots + d_n \cdot pc(m_n)$.

Example 4
In Figure 4.7, assuming Subsystem sub_1 in initial mode m_4, target mode m_4, and a time interval of 30, a switching sequence $seq_1^1 = (m_4, c \geq 0) \xrightarrow[c=0]{\sigma_9, sv_1 = m_7} (m_7, c \geq 0) \xrightarrow{d_1=24} (m_7, c == 24) \xrightarrow[c=0]{\sigma_{10}, sv_1 = m_3} (m_3, c \geq 0) \xrightarrow{d_2=6} (m_3, c == 6) \xrightarrow[c=0]{\sigma_5, sv_1 = m_4} (m_4, c \geq 0)$ has a minimum energy demand of $e(seq_1^1) = 0 \cdot pc(m_4) + 24 \cdot pc(m_7) + 6 \cdot pc(m_3) + 0 \cdot pc(m_4) = 24 \cdot 1660 + 6 \cdot 420 = 39.840 + 2.520 = 42.360$

A switching sequence seq_k^i is an instance within the set of alternative switching sequences of a Subsystem sub_i. Those alternatives are considered next.

5.1.2 Alternative switching sequences within a subsystem

A finite set of switching sequences may exist with predefined initial and target modes. The problem of finding all paths between two modes is of practical importance already introduced by [Thorelli, 1966]. Identifying all alternatives between two modes has the time-complexity of $\mathcal{O}^i(\text{M}^i + \text{Ed}^i)$ with M^i as the number of modes and Ed^i as the number of edges by traversing the directed graph using recursive search [Migliore, 1990].
Computing the number of sequence alternatives $\mid \text{Seq}^i \mid$ of a Subsystem sub_i with sequences $\text{seq}_k^i \in \text{Seq}^i$ can be formulated as recursive search. The recursive depth-first search implemented in Algorithm 2 is used for checking if a target state $\text{St}_{\text{tar}}^{\text{symb}}$ is reachable by the initial state $\text{St}_{\text{reach}}^{\text{symb}}$. The quantity of sequence alternatives $\mid \text{Seq}^i \mid$ in an automation subsystem can be

computed by a recursive depth-first search in the directed graph (Algo. 2). The occurrence of each mode while conducting the depth-first is saved by variables *occ* for each mode separately (initially: *occ(m) = 0*). The quantity of alternative sequences $\mid \mathrm{Seq}^i \mid$ (Line 15) is computed as follows. For each successor state in $\mathrm{St}^{\mathrm{symb}}_{\mathrm{reach}}$, Algorithm 2 is called with parameters $\mathrm{St}^{\mathrm{symb}}_{\mathrm{reach}} \cup \{(\mathrm{m}', \mathrm{t}) \in \mathrm{St}' \mid \exists(\mathrm{m}, \mathrm{t}) : \mathrm{m} \in \mathrm{M} \wedge (\mathrm{m}, \mathrm{t}) \rightarrow (\mathrm{m}', \mathrm{t})\}$ and $\mathrm{St}^{\mathrm{symb}}_{\mathrm{tar}}$ (Line 17).

Data: $\mathrm{St}^{\mathrm{symb}}_{\mathrm{reach}} = (\mathrm{m}, \mathrm{t})$, $\mathrm{St}^{\mathrm{symb}}_{\mathrm{tar}} = (\mathrm{m}_{\mathrm{tar}}, \mathrm{t}_{\mathrm{tar}})$
Result: $\mid \mathrm{Seq}^i \mid = 0$

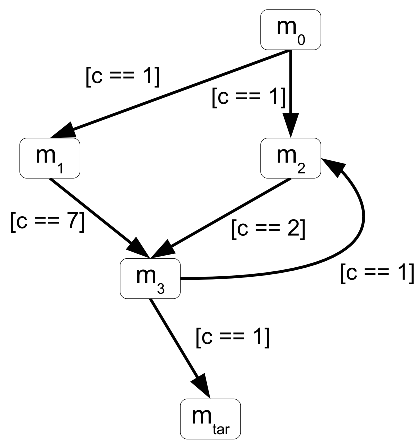
1 **if** $occ(m) > 2$ **then** // minimum-switch property
2   return $\mid \mathrm{Seq}^i \mid$
3 **else**
4   occ(m)++
5   **if** $St^{symb}_{reach} \cap St^{symb}_{tar} \neq \emptyset$ **then**
6   return $\mid \mathrm{Seq}^i \mid = \mid \mathrm{Seq}^i \mid + 1$
7   **else**
8   **if** $m \neq m_{tar}$ **then**
9   **while** $\nexists\, (m', t) : m \in M \wedge (m, t) \rightarrow (m', t)$ **do**
10                 $\mathrm{St}^{\mathrm{symb}}_{\mathrm{reach}} = \mathrm{St}^{\mathrm{symb}}_{\mathrm{reach}} \cup \{(\mathrm{m}, \mathrm{t}') \in \mathrm{St}' \mid \exists(\mathrm{m}, \mathrm{t}) : \mathrm{m} \in \mathrm{M} \wedge (\mathrm{m}, \mathrm{t}) \rightarrow (\mathrm{m}, \mathrm{t}')\}$
11   **end**
12   **else**
13   return $\mid \mathrm{Seq}^i \mid$
14   **end**
15         $\mid \mathrm{Seq}^i \mid = \mid \mathrm{Seq}^i \mid +$
16   **foreach** $\{(m',t)\ in\ St^{symb}_{reach} \mid \exists(m, t) : m \in M \wedge (m, t) \rightarrow (m', t)\}$ **do**
17             $\mathrm{dfs}_{recursive}(\mathrm{St}^{\mathrm{symb}}_{\mathrm{reach}} \cup \{(\mathrm{m}', \mathrm{t}) \in \mathrm{St}' \mid \exists(\mathrm{m}, \mathrm{t}) : \mathrm{m} \in \mathrm{M} \wedge (\mathrm{m}, \mathrm{t}) \rightarrow (\mathrm{m}', \mathrm{t})\}, \mathrm{St}^{\mathrm{symb}}_{\mathrm{tar}})$
18   **end**
19   return $\mid \mathrm{Seq}^i \mid$
20   **end**
21 **end**

Algorithm 2: $\mathrm{dfs}_{recursive}(\mathrm{St}^{\mathrm{symb}}_{\mathrm{reach}}, \mathrm{St}^{\mathrm{symb}}_{\mathrm{tar}})$

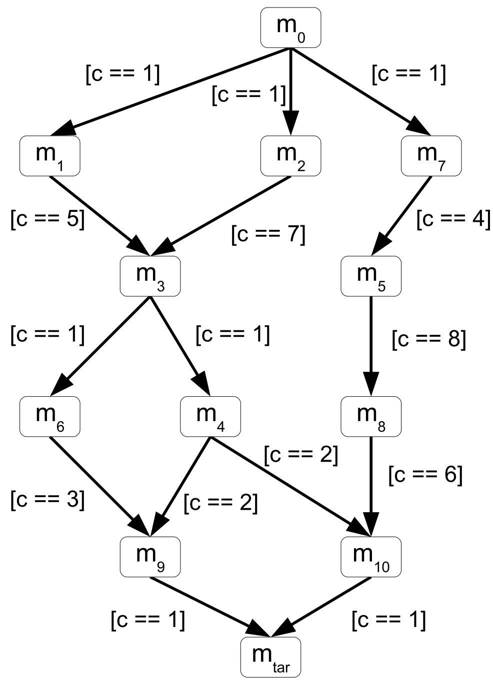
Lemma 5.1.1 (Correctness of Algorithm 2)
Algorithm 2 computes the number of switching sequences $\mid Seq^i \mid$ within a directed graph (timed automaton) with each mode having a successor mode (except the target mode m_{tar}). It terminates after a finite number of steps with a symbolic initial state $St^{symb}_{reach} = (m_i, c \geq 0)$ and a symbolic target state $St^{symb}_{tar} = (m_j, c \geq 0)$, $m_i \neq m_j$.

Proof. *In a first run, the occurrence of mode m_i is initially 0, so that the comparison in Line 1 returns false. $occ(m)$ is increased by 1 in Line 4. $St^{symb}_{reach} \cap St^{symb}_{tar}$ is equal to the empty set, so that the else part* *is activated (Line 7). The while-loop checks for existing successor modes m_i'. In Line 8, it is checked that mode m_i is not the target mode m_{tar}, if so, it is returned $\mid Seq^i \mid$ meaning that m_{tar} could not be reached within t_{tar}. As long as no successor mode m_i' exists, a delay for mode m_i is computed (Line 10). Since each mode m_i has a successor mode m_i' (except the target mode m_{tar}), the while-loop is left after a finite number of steps. Since the set of successor modes m_i' of a mode m_i is finite (except the target mode m_{tar}), the foreach-loop terminates after a finite number of steps (Line 16). The number of switching sequences $\mid Seq^i \mid$ is returned, expressing the number of switching sequences possible to reach the target state, after having identified all alternatives.*

Two examples (Fig. 5.1) illustrate the identification of alternative sequences Seq^i based on Algorithm 2. The graph may contain cycles (see Fig. 5.1a), but no dead ends except the target mode $\mathrm{m_{tar}}$. Time delays are expressed as guards on clock c annotated at transitions ($\mathrm{m_0} \xrightarrow{d=1} \mathrm{m_0} \rightarrow \mathrm{m_1}$ in Figure 5.1a). The clock is reset at each transition.



(a)



(b)

Figure 5.1: Alternative switching sequences with initial mode $\mathrm{m_0}$ and target mode $\mathrm{m_{tar}}$

Example 5

(Fig. 5.1a): The objective is to compute the number of sequence alternatives with initial state St^{symb}_{reach} = (m_0, $c \geq 0$) and target state St^{symb}_{tar} = (m_{tar}, $c \leq 7$). Computing the sequence alternatives $\mid Seq \mid$, results in two different switching sequences: $|Seq|$ = 2 (Tab. 5.1).

Alternative	Sequence	$(\mathrm{m_{tar}}, \mathrm{c})$	$(\mathrm{c} \leq 7)$?
1	$\mathrm{m}_0 \xrightarrow{+1} \mathrm{m}_1 \xrightarrow{+7} \mathrm{m}_3 \xrightarrow{+1} \mathrm{m}_{\mathrm{tar}}$	$(\mathrm{m}_{\mathrm{tar}}, \mathrm{c} \geq 9)$	no
2	$\mathrm{m}_0 \xrightarrow{+1} \mathrm{m}_2 \xrightarrow{+2} \mathrm{m}_3 \xrightarrow{+1} \mathrm{m}_{\mathrm{tar}}$	$(\mathrm{m}_{\mathrm{tar}}, \mathrm{c} \geq 4)$	yes
3	$\mathrm{m}_0 \xrightarrow{+1} \mathrm{m}_2 \xrightarrow{+2} \mathrm{m}_3 \xrightarrow{+1} \mathrm{m}_2 \xrightarrow{+2} \mathrm{m}_3 \xrightarrow{+1} \mathrm{m}_{\mathrm{tar}}$	$(\mathrm{m}_{\mathrm{tar}}, \mathrm{c} \geq 7)$	yes

Table 5.1: Feasible switching sequences in Figure 5.1a

Example 6
(Fig. 5.1b): The objective is to compute the number of sequence alternatives with initial state St^{symb}_{reach} = $(m_0, c \geq 0)$ and target state St^{symb}_{tar} = $(m_{tar}, c \leq 11)$. Computing the sequence alternatives $\mid Seq \mid$, results in three different switching sequences: $|Seq|$ = 3.

In this subsection, the view is limited to alternative switching sequences of a single subsystem. Since only single subsystems are addressed by switching sequences, relationships between switching sequences of different subsystems needs to be established accounting for subsystem dependencies. The link between switching sequences of different subsystems is addressed by the concept of strategies, next.

5.1.3 Strategies within a system

In general, switching sequences of a single subsystem are related to switching sequences of other subsystems. This is due to subsystem dependencies (Def. 24). The combination of switching sequences on a system level is called a *strategy*. The terms *unrelated* and *related* strategy are introduced and distinguished.

Set of unrelated strategies

If the set of alternative switching sequences Seq^i of a Subsystem sub_i and the set of alternative switching sequences Seq^j of Subsystem sub_j do not contain common constraints, the switching sequences seq^i_k and seq^j_o are called independent. The set of unrelated strategies is defined according to Definition 35.

Definition 35 (Set of unrelated strategies)
The set of unrelated strategies $L_{unrelated}$ is the Cartesian product of $Seq^1 \times \ldots \times Seq^n$ with 1, ..., n as the number of subsystems omitting subsystem dependencies. An unrelated strategy is denoted as $l_{p,unrel} \in L_{unrelated}$. The unrelated strategy $l_{p,unrel}$ is a tuple $(seq^1_a, seq^j_b, ..., seq^n_c)$ with $seq^1_a \in Seq^1$, $seq^j_b \in Seq^j$, and $seq^n_c \in Seq^n$.

The energy demand of an unrelated strategy is the linear superposition (sum) of energy demands of its switching sequences (Lemma 5.1.2).

Lemma 5.1.2 (Minimum energy demand of an unrelated strategy)
The minimum energy demand of an unrelated strategy is given by
$e(l_{p,unrel}) = e(seq_k^i, seq_l^j) = e(seq_k^i) + e(seq_l^j)$.

Proof. *Assuming switching sequences seq_k^i and seq_l^j of $l_{p,unrel}$ not containing common constraints. The minimum energy demand $e(seq_k^i)$ can be computed independently from the energy demand $e(seq_l^j)$. For this reason, the minimum energy demand $e(l_{p,unrel})$ of the unrelated strategy is the linear superposition of the minimum energy demand $e(seq_k^i)$ and $e(seq_l^j)$.*

Example 7
In Figure 4.7, two switching sequences in Subsystem sub_1 and Subsystem sub_2 are considered within a time interval of 90 (resets on clocks and updates on shared variables are omitted for a simplified presentation):

- $seq_1^1 = (m_2, c \geq 0) \xrightarrow{\sigma_{12}} (m_0, c \geq 0) \xrightarrow{d=8} (m_0, c = 8) \xrightarrow{\sigma_2} (m_1, c \geq 0) \xrightarrow{d=82} (m_1, c = 82) \xrightarrow{\sigma_3} (m_2, c \geq 0)$
- $seq_1^2 = (m_3, c \geq 0) \xrightarrow{\sigma_{21}} (m_2, c \geq 0) \xrightarrow{d=90} (m_2, c = 90) \xrightarrow{\sigma_{17}} (m_3, c \geq 0)$

Both switching sequences are considered as independent, so that $e(seq_1^1) = 8 \cdot 0 + 82 \cdot 0 = 0$ and $e(seq_1^2) = 90 \cdot 320 = 28.800$. $e(l_{p,unrel}) = e(seq_1^1, seq_1^2) = e(seq_1^1) + e(seq_1^2) = 28.800$.
Subsystem dependencies are omitted in case of the unrelated strategy. The unrelated strategy $l_{p,unrel}$ = (seq_1^1, seq_1^2) corresponds to switching sequences seq_1^1 and seq_1^2.

The set of unrelated strategies can be ordered with regard to the energy demand of its elements (Lemma 5.1.3).

Lemma 5.1.3 (Finite ordered set of unrelated strategies)
The finite ordered set of unrelated strategies is given by ($L_{unrelated}$, $\leq$) with
$l_{p,unrel} \leq l_{p+1,unrel} \Leftrightarrow e(l_{p,unrel}) \leq e(l_{p+1,unrel})$, *p = 1, ..., n - 1.*

Proof. *The set of switching sequences (Seq^i, $\leq$) of a Subsystem sub_i is ordered by $seq_k^i \leq seq_{k+1}^i$ $\forall$ k $\Leftrightarrow$ e(seq_k^i) $\leq$ e(seq_{k+1}^i). Since the energy demand of an unrelated strategy is calculated by $e(l_{p,unrel}) = \sum_{i=1}^{n} e(seq_k^i)$ with k = 1, 2, ... n, the set of unrelated strategies can be ordered in a monotonically increasing way in the form of $e(l_{p,unrel}) \leq e(l_{p+1,unrel})$ $\forall$ p = 1, ..., n - 1.*

Set of related strategies

In general, subsystems are not independent from each other (Subsection 4.1.1). Caused by subsystem dependencies, automation subsystems are interdependent which must be reflected by a related strategy (Def. 36).

Definition 36 (Set of related strategies)
The set of related strategies $L = Seq^i \times Seq^j \times ... \times Seq^n$ *consists of related strategies* $l_p = (seq^i_a, seq^j_b, ..., seq^n_c)$ *with* $seq^i_a \in Seq^i$, $seq^j_b \in Seq^j$, *and* $seq^n_c \in Seq^n$. *The switching sequences* seq^i_a, seq^j_b, *and* seq^n_c *exhibit subsystem dependencies.*

The minimum energy demand of a related strategy is greater than or equal to the linear superposition of minimum energy demands of its switching sequences (Lemma 5.1.4).

Lemma 5.1.4 (Minimum energy demand of a related strategy)
The minimum energy demand of a related strategy is greater-equal the linear superposition of the minimum energy demands of its switching sequences: $e(l_p) = e(seq^i_k, seq^j_l) \geq e(seq^j_l) + e(seq^j_l)$.

Proof. *Assuming switching sequences* seq^i_k *and* seq^j_l *of* l_p *containing common constraints. If the common constraints have no impact on* $e(seq^i_k)$ *respectively* $e(seq^j_l)$, *then* $e(l_{p,unrel}) = e(seq^j_l) + e(seq^j_l)$. *If the common constraints affect the minimum energy demand of* $e(seq^i_k)$ *respectively* $e(seq^j_l)$, *then* $e(l_{p,unrel}) > e(seq^j_l) + e(seq^j_l)$ *holds. In summary, the minimum energy demand for* $e(l_{p,unrel})$ *is equal to or greater than the sum of minimum energy demands* $e(seq^j_l)$ *and* $e(seq^j_l)$: $e(l_p) \geq e(seq^j_l) + e(seq^j_l)$.

The following Theorem 5.1.5, can be stated by the use of Lemma 5.1.2 and Lemma 5.1.4.

Theorem 5.1.5 (Lower bound)
Caused by common constraints of a related strategy l_p, *the minimum energy demand* $e(l_{p,unrel})$ *of an unrelated strategy is the lower bound for the minimum energy demand* $e(l_p)$ *of a corresponding related strategy :* $e(l_p) \geq e(l_{p,unrel})$.

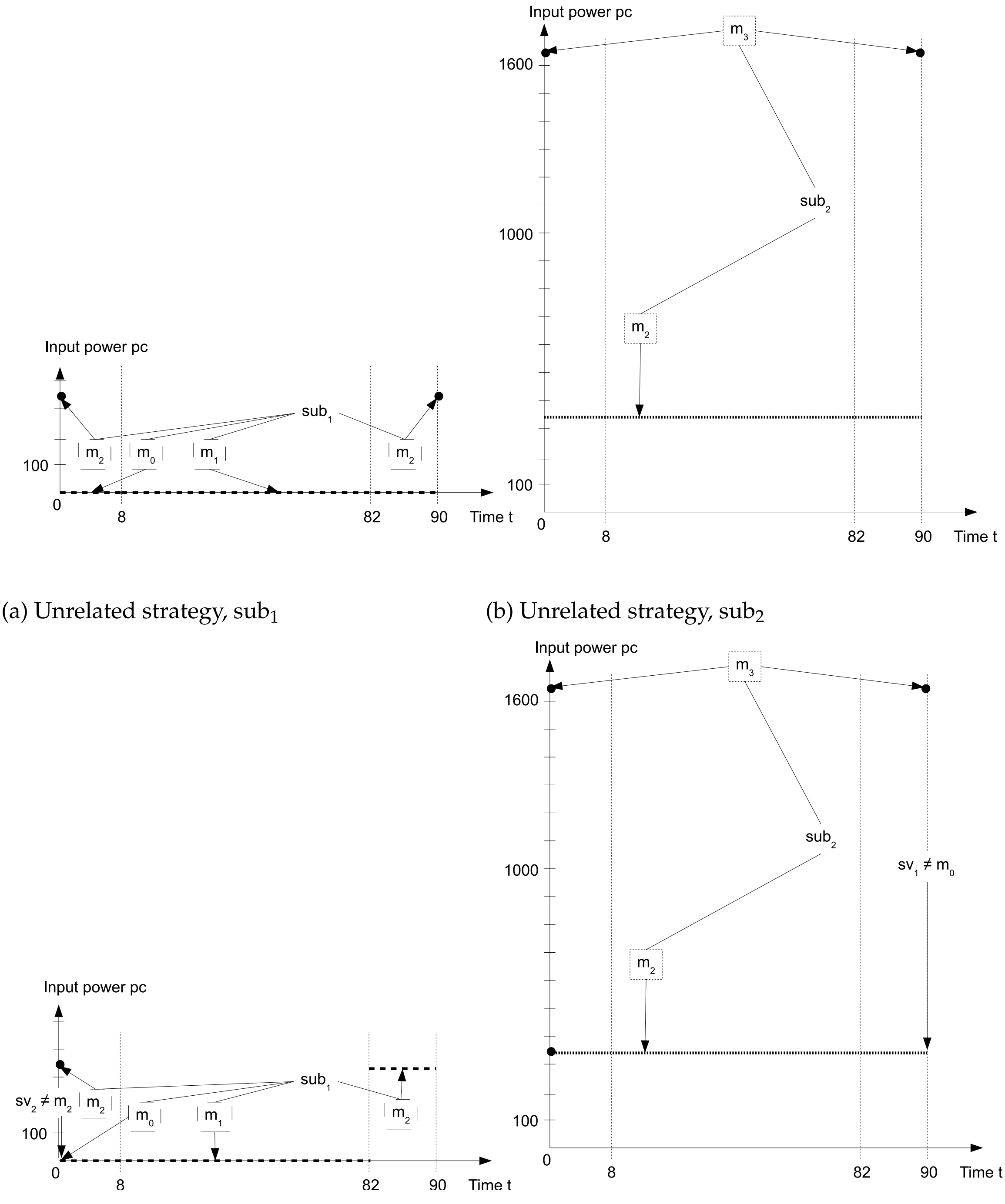
Proof. *Using Lemma 5.1.2 and Lemma 5.1.4, it holds:* $e(l_p) \geq e(seq^j_l) + e(seq^j_l) = e(l_{p,unrel})$.

Example 8
In Figure 4.7, two switching sequences in Subsystem sub_1 *and Subsystem* sub_2 *are considered within a time interval of 90 (resets on clocks and updates on shared variables are omitted for a simplified presentation):*

- $seq^1_1 = (m_2, c \geq 0) \xrightarrow{\sigma_{12}} (m_0, c \geq 0) \xrightarrow{\sigma_2, sv_2 \neq m_2} (m_1, c \geq 0) \xrightarrow{d=82} (m_1, c = 82) \xrightarrow{\sigma_3} (m_2, c \geq 0) \xrightarrow{d=8} (m_2, c = 8)$
- $seq^2_1 = (m_3, c \geq 0) \xrightarrow{\sigma_{21}} (m_2, c \geq 0) \xrightarrow{d=90} (m_2, c = 90) \xrightarrow{\sigma_{17},, sv_1 \neq m_0} (m_3, c \geq 0)$

Both switching sequences are dependent, so that $e(seq^1_1) = 82 \cdot 0 + 8 \cdot 320 = 2.560$ *and* $e(seq^2_1) = 90 \cdot 320 = 28.800$. $e(l_{p,unrel}) = e(seq^1_1, seq^2_1) = e(seq^1_1) + e(seq^2_1) = 31.360$. *Subsystem dependencies are effective in case of the related strategy. The related strategy* l_p = (seq^1_1, seq^2_1) *corresponds to switching sequences* seq^1_1 *and* seq^2_1.

Figure 5.2 enables the comparison of the energy demand in Example 7 (unrelated strategy) with Example 8 (related strategy). Switching sequence (Fig. 5.2a, Fig. 5.2b, Fig. 5.2c, Fig. 5.2d) for Subsystem sub_1 and Subsystem sub_2 are presented with minimum energy demand.
It is shown that the minimum energy demand (integral of the input power over time) for the unrelated strategy $l_{p,unrel}$ (Integral in Fig. 5.2a and Fig. 5.2b) is smaller than the minimum energy demand for the related strategy l_p (Integral in Fig. 5.2c and Fig. 5.2d). This is due to the fact that the subsystem dependency $sv_2 \neq m_2$ while changing from mode m_0 to m_1 requires Subsystem sub_2 not being in mode m_2. However, it is the best alternative that Subsystem sub_2 stays 90 time units in mode m_2. Since Subsystem sub_1 is required to distribute 90 time units among the modes of switching sequence seq_1^1, Subsystem sub_1 must spend 8 time units in mode m_2 (Fig. 5.2c) instead of 8 time units in mode m_0. The delay in mode m_1 within Subsystem sub_1 is fixed to 82 time units. The subsystem dependency $sv_1 \neq m_0$ has no effects on the switching sequences (Fig. 5.2d).
So far, identifying the minimum energy demand for switching sequences of subsystems and the minimum energy demand for strategies was made intuitively by comparing the input powers of different modes manually. This is feasible for a small set of switching sequences and subsystems. Having larger sets of switching sequences in practice which results in a larger set of strategies to be compared, computational support is required. A formalism solving the set of optimization problems is necessary for automated computation. Therefore, the transformation of switching sequences and strategies into optimization problems is presented next.



(a) Unrelated strategy, sub$_1$

(b) Unrelated strategy, sub$_2$

(c) Related strategy, sub$_1$

(d) Related strategy, sub$_2$

Figure 5.2: Input power over time of the unrelated and related strategy of Examples 7 and 8 based on switching sequences seq_1^1 and seq_1^2

5.2 Strategy optimization problem

In this section, the strategy optimization problem is introduced using the theoretical background presented in Section 3.4. The optimization problem of this thesis is in the field of combinatorial optimization with discrete decision variables. The objective is to identify the optimal strategy in the set of strategies. Each strategy is expressed as its own optimization problem [Mechs, 2012b].

In the remainder of this section, the decision variables of the optimization problem are introduced (Subsection 5.2.1). In Subsection 5.2.2, two objective functions are defined. The constraints of the optimization problem are given in Subsection 5.2.3. It is distinguished between constraints for unrelated strategies (Def. 35) and related strategies (Def. 36).

5.2.1 Decision variables

The decision variables of the optimization problem are based on *interval variables* (Def. 37).

Definition 37 (Interval variable)
An interval variable $v_i(seq_k^j)$ *is a discrete variable which belongs to a switching sequence* seq_k *in subsystem* sub_j *with the domain* $dom[v_i(seq_k^j)] = \{[start[v_i(seq_k^j)], end[v_i(seq_k^j)]] \mid start[v_i(seq_k^j)], end[v_i(seq_k^j)] \in \mathbb{N}_0^+$ *and* $start[v_i(seq_k^j)] \leq end[v_i(seq_k^j)]\}$.

The lengths of interval variables are the decision variables of the optimization problem (Def. 38).

Definition 38 (Length of an interval variable)
The length of an interval variable is defined by $length[v_i(seq_k^j)] = end[v_i(seq_k^j)]$ - $start[v_i(seq_k^j)]$.

Required for stating the optimization problem, the modes of a switching sequence seq_k^j (Def. 32) are mapped to the set of interval variables V (Equ. 5.1).

$$\mathrm{P} : \mathrm{M}(\mathrm{seq}_k^j) \mapsto \mathrm{V}(\mathrm{seq}_k^j) \tag{5.1}$$

The input power $\mathrm{pc}(\mathrm{m}_k^i)$ of a mode is mapped to the input power $\mathrm{pc}(\mathrm{v}_i(\mathrm{seq}_k^j))$ of the interval variable (Equ. 5.2).

$$\mathrm{U} : \mathrm{PC}\left[\mathrm{M}(\mathrm{seq}_k^j)\right] \mapsto \mathrm{PC}\left[\mathrm{V}(\mathrm{seq}_k^j)\right] \tag{5.2}$$

Using Definition 39, the energy demand for a decision variable can be quantified. The input power $\mathrm{pc}(\mathrm{v}_i(\mathrm{seq}_k^j))$ of an interval variable $\mathrm{v}_i(\mathrm{seq}_k^j)$ is a constant weight (Subsection 3.4.1).

Definition 39 (Energy demand of an interval variable)
The energy demand of an interval variable is defined by $e[v_i(seq_k^j)] := pc[v_i(seq_k^j)] \cdot length[v_i(seq_k^j)]$.

5.2.2 Objective functions

Considerations towards optimality require an objective function *obj*. In the context of reduced energy demand within unproductive phases, the interval variables as decision variables are mapped to their energy demands by obj (Equ. 5.3).

$$\text{obj} : \text{V}(\text{seq}_k^j) \mapsto \mathbb{N}_0^+ \tag{5.3}$$

The objective $\text{obj}_{\text{energy}}$ accounts for the minimum energy demand of a set of interval variables (Equ. 5.4).

$$\text{Minimize obj}_{\text{energy}} = \text{Minimize} \sum_{\text{i}}^{\text{n}} \text{pc}_{\text{i}}[\text{v}_{\text{i}}(\text{seq}_k^j)] \cdot \text{length}[\text{v}_{\text{i}}(\text{seq}_k^j)] \tag{5.4}$$

Based on Equation 5.4, the minimum energy demand of an unrelated strategy (Lemma 5.1.2) or a related strategy (Lemma 5.1.4) is quantified.
The objective obj_{time} expresses that a given target mode m_{tar} (represented by $\text{v}_{\text{n}}(\text{seq}_k^j)$) is reached in minimum time (Equ. 5.5).

$$\text{Minimize obj}_{\text{time}} = \text{Minimize} \sum_{\text{i}}^{\text{n}} \text{length}[\text{v}_{\text{i}}(\text{seq}_k^j)] \tag{5.5}$$

Comparing objective functions $\text{obj}_{\text{energy}}$ and obj_{time}, both objective functions differ in the factor $\text{pc}_{\text{i}}[\text{v}_{\text{i}}(\text{seq}_k^j)]$. Objective function obj_{time} can be interpreted as objective function $\text{obj}_{\text{energy}}$ with $\text{pc}_{\text{i}}[\text{v}_{\text{i}}(\text{seq}_k^j)] = 1$.

5.2.3 Strategy constraints

Representing strategy constraints in the optimization problem, it is distinguished between the constraints for unrelated strategies and related strategies.

Constraints for unrelated strategies

Clock guards $G^j(c)$ on an outgoing edge *ed* of mode *m* with
$\text{g}^j := \text{c} \sim \text{t}$ and $\sim \in \{<, \leq, =, \geq, >\}$ and $\text{t} \in \mathbb{N}^+$ are transformed to temporal constraints (Equ. 5.6).

$$\text{Q} : \text{g}^j \mapsto \textit{length}(\text{v}_i(\text{seq}_k^j)) \sim \text{t} \tag{5.6}$$

Furthermore, the succession of modes within a switching sequence seq_k^j is expressed by the set of precedence constraints *Pre*. A precedence constraint represents the relative temporal position of an interval variable with reference to another interval variable. The precedence of $\text{m}_{\text{p}}(\text{seq}_k^i)$ and $\text{m}_{\text{p+1}}(\text{seq}_k^i)$ with $\text{p} \in \mathbb{N}^+$ is transformed into a precedence constraint of interval variables by Equation 5.7.

$$\mathrm{pre} := \mathrm{end}\left[\mathrm{v_p}(\mathrm{seq}_k^i)\right] = \mathrm{start}\left[(\mathrm{v_{p+1}}(\mathrm{seq}_k^i)\right] \tag{5.7}$$

The usage of precedence constraints is similar to the precedence constrained scheduling problem presented in Subsection 3.4.1.

Constraints for related strategies

In addition to the constraints for unrelated strategies, subsystem dependencies need to be expressed as constraints between switching sequences of different subsystems. Guards g_{eq}^j, g_{neq}^j on shared variables sv_k defined on transitions of the switching sequence of subsystems sub_j and sub_k with $j \neq k$ are transformed into temporal constraints.

For guards $\mathrm{g}_{\mathrm{eq}}^j := \mathrm{sv}_k == \mathrm{m}_o^k$ holds:

$$\mathrm{start}[\mathrm{v}(\mathrm{seq}_i^j)] \ \leq \ \mathrm{end}[\mathrm{v}_o(\mathrm{seq}_l^k)] \ \wedge \mathrm{start}[\mathrm{v}(\mathrm{seq}_i^j)] \ \geq \ \mathrm{start}[\mathrm{v}_o(\mathrm{seq}_l^k)] \tag{5.8}$$

For $\mathrm{g}_{\mathrm{neq}}^j := \mathrm{sv}_k \neq \mathrm{m}_o^k$ holds:

$$\mathrm{end}[\mathrm{v}(\mathrm{seq}_i^j)] \ < \ \mathrm{start}[\mathrm{v}_o(\mathrm{seq}_l^k)] \ \vee \mathrm{start}[\mathrm{v}(\mathrm{seq}_i^j)] \ > \ \mathrm{end}[\mathrm{v}_o(\mathrm{seq}_l^k)] \tag{5.9}$$

Example 9

Figure 5.3 exemplifies the mapping of subsystem dependencies to temporal constraints of the optimization problem. In 5.3 (1a), the transition from m_1^1 to m_2^1 is guarded by $g_{eq}^1 := sv_2 == m_2^2$. The guard expresses that Subsystem sub_2 needs to be in mode m_2^2 to enable the transition from m_1^1 to m_2^1 in Subsystem sub_1. This results in $start[v_1(seq_1^1)] \ \leq \ end[v_2(seq_1^2)] \ \wedge start[v_1(seq_1^1)] \ \geq \ start[v_2(seq_1^2)]$ (Equ. 5.8). This is illustrated in Figure 5.3 (1b).

In Figure 5.3 (2a), the transition from m_1^1 to m_2^1 is guarded by $g_{neq}^1 := sv_2 \neq m_2^2$. This guard expresses that Subsystem sub_2 must not be in mode m_2^2 if the transition from m_1^1 to m_2^1 in Subsystem sub_1 fires. This results in $end[v_1(seq_1^1)] \ < \ start[v_2(seq_1^2)] \ \vee start[v_1(seq_1^1)] \ > \ end[v_2(seq_1^2)]$ (Equ. 5.9). A graphical representation of the temporal constraints is given (Figure 5.3 (2b) or (2c)).

In this way, logical guards on shared variables expressing the precedence of modes within different subsystems are mapped to common constraints between switching sequences of the optimization problem.

Pause interval constraints

The energy-minimizing objective obj_{energy} (Equ. 5.4, Subsection 5.2.2) can only be formulated using a pause interval constraint (Equ. 5.10). For this purpose, the domain of interval variables $\mathrm{dom}[\mathrm{v}_i(\mathrm{seq}_k^j)]$ needs to be constrained according to the pause interval $d^{global} \in \mathbb{N}^+$. This is formulated as constraints on the sum of interval variables for each switching sequence seq_k^j of a Subsystem sub_j in Equation 5.10:

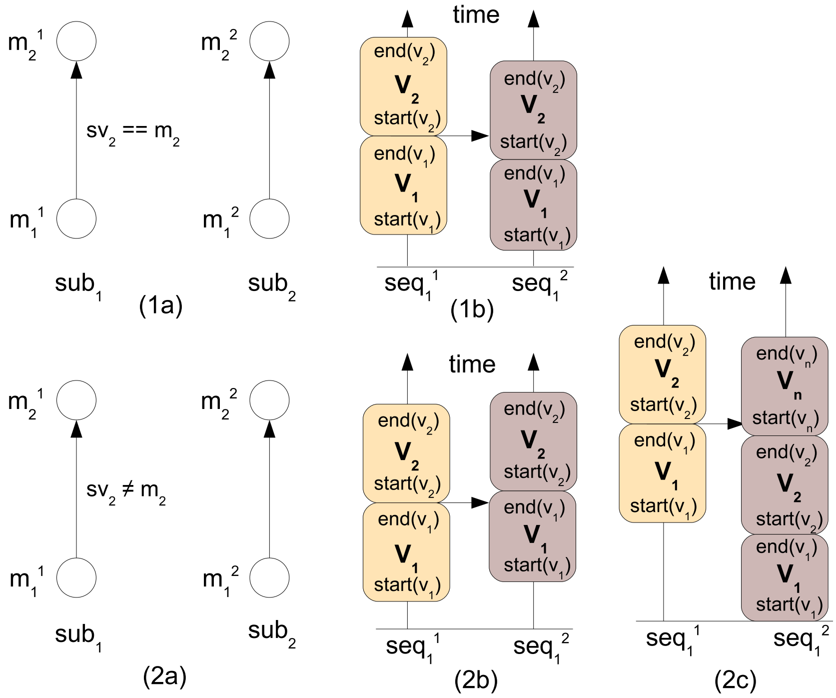


Figure 5.3: Mapping of guarded transitions caused by subsystem dependencies (1a), (2a) to temporal constraints on interval variables (1b), (2b), and (2c)

$$\sum_{\mathrm{i}} \mathrm{length}(\mathrm{v_i}(\mathrm{seq}_k^j)) = \mathrm{d}^{\mathrm{global}} \tag{5.10}$$

It has to be noted, that the sum of interval variable lengths needs to be equal to the pause interval $\mathrm{d}^{\mathrm{global}}$. This forces a subsystem sub_i to rest in a specific mode for each time point $\mathrm{t}_i \in \mathrm{d}^{\mathrm{global}}$.

5.3 Summary

In this chapter, the optimization problem for single strategies expressing a system view has been introduced (Section 5.1). A strategy is composed of switching sequences. It has been distinguished between an unrelated strategy (omitting subsystem dependencies) and a related strategy (Subsection 5.1.3). Within a related strategy, subsystem dependencies are expressed as common constraints between interval variables of different switching sequences. In contrast, an unrelated strategy omits common constraints between interval variables of different switching sequences. The corresponding constraint optimization problem has been introduced based on the distinction between unrelated strategies and related strategies (Section 5.2). Stating the optimization problem for a single strategy, the minimum energy demand of single unrelated and related strategies can be computed (Lemma 5.1.2 and Lemma 5.1.4).

For a given unproductive phase within an automation system, there exists a finite set of related strategies. Consequently, the strategy optimization problem has to be computed n times in order to identify the energy-optimal related strategy and to address the research objective of this thesis (Section 1.3).
A finite, but huge number of related strategies may exist for automation systems like the test bed M presented in Subsection 8.3.2. As it will be shown in Section 9.1 of Part III, the identification of the energy-optimal related strategy is a challenging task.
For this purpose, the distinction between unrelated and related strategies is made. An unrelated strategy is a relaxed correspondent of a related strategy, since both are based on the same switching sequences. Divide-and-conquer procedures are applicable if the optimization problem can be split up into several subproblems. Within unrelated strategies, the independent switching sequences are subproblems of the strategy optimization problem. This enables the efficient computation of the minimum energy demand of unrelated strategies. Theorem 5.1.5, stating that the minimum energy demand of a related strategy cannot be lower than the minimum energy demand of the unrelated correspondent, is used to propose a procedure in Chapter 6 which supports the identification of the energy-optimal strategy out of a set of alternative related strategies.

Chapter 6

Bounded investigation of the set of strategies

In Chapter 5, stating the optimization problem for a single strategy (unrelated strategy or related strategy) has been focus. The objective of this chapter is to propose a procedure for identifying the energy-optimal related strategy l_{opt} out of the set of related strategies L (Def. 40).

Definition 40 (Energy-optimal related strategy)
The energy-optimal related strategy $l_{opt} \in L$ is a related strategy l_p with energy demand $e(l_p) \leq e(l_k)$ $\forall$ k and $p \neq k$.

First, the reduced set of strategies being considered is stated as constraint satisfaction problem in Section 6.1. The analytical basis for selecting the energy-optimal related strategy from the set L is given in Section 6.2. By means of a selective procedure (Subsection 3.4.3) the problem structure (modularity) is exploited to avoid exhaustive search. This procedure enables the efficient investigation of the set of related strategies by means of constraint relaxation (Section 6.3).

6.1 Reduced set of strategies within a system

Based on the Cartesian product of switching sequences, a finite set of strategies exists (Def. 35 and Def. 36). The set of strategies to be considered can be reduced by computing this set by a Constraint Satisfaction Problem (CSP) (Def. 41) [Montanari, 1971].

Definition 41 (Constraint satisfaction problem)
The constraint satisfaction problem is a tuple CSP = (s, dom(s), X) with

- *s = { s^i, ..., s^n } as variables of Subsystems sub_i to sub_n*
- *dom(s) = { dom(s^i), ..., dom(s_m) } as the set of domains of variables of Subsystems sub_i to sub_m in the form of dom(s^i) = [seq_1^i, ..., seq_n^i] with seq_k^i as switching sequence k for subsystem sub_i*

- *X(s) is a finite set of if-then constraints stated between switching sequences of different subsystems* sub_i *and* sub_j

An if-then constraint is defined as follows (Def. 42).

Definition 42 (If-then constraint)
If a switching sequence seq_k^j *contains an equality guard* g_{eq}^j *(Def. 29) in the form of* $g_{eq}^j := sv_l == m_i$ *referring to a shared variable* sv_l*, then this is expressed as if-then constraint as follows:*
if (seq_k^j) *then* (seq_o^l) *with* $m_i \in seq_o^l$
The if-then constraint implies for a strategy containing seq_k^j *and* seq_o^l*:* $(seq_k^j, ..., seq_o^l)$*.*

Example 10
In Figure 4.7, a switching sequence $seq_2^1 = (m_5, c \geq 0) \xrightarrow{\sigma_7, sv_2 == m_3} (m_6, c \geq 0) \xrightarrow{d^{trans}=60} (m_6, c = 60) \xrightarrow{\sigma_8} (m_4, c \geq 0)$ *requires Subsystem* sub_2 *to include mode* m_3 *into a switching sequence* seq_o^2*. Therefore, switching sequence* seq_2^1 *and* $seq_2^2 = (m_4, c \geq 0) \xrightarrow{\sigma_{19}} (m_5, c \geq 0) \xrightarrow{d^{trans}=10} (m_5, c = 10) \xrightarrow{\sigma_{20}} (m_3, c \geq 0)$ *can be used in a strategy* (seq_2^1, seq_2^2)*, since* $m_3 \in seq_2^2$*. The switching sequence* $seq_3^2 = (m_2, c \geq 0) \xrightarrow{\sigma_{22}} (m_0, c \geq 0) \xrightarrow{\sigma_{15}} (m_1, c \geq 0) \xrightarrow{d^{trans}=82} (m_1, c = 82) \xrightarrow{\sigma_{16}} (m_2, c \geq 0)$ *cannot occur in a strategy together with* seq_2^1*.*

Applying Definition 42, the power of the set of strategies (Def. 35 and Def. 36) can be reduced. In general, the maximum number of related strategies is based on the Cartesian product of the sets of switching sequences $\mathrm{Seq}^1 \times \ldots \times \mathrm{Seq}^n$ and increases exponentially with the number of Subsystems sub_i [Kolbe, 2000] (Equ. 6.1).

$$| L | = \prod_{i=1}^{n} \mathrm{Seq}^i \tag{6.1}$$

with Seq^i as the quantity of alternative switching sequences in Subsystem sub_i.
Using the procedure introduced in the remainder of this chapter, the set of related strategies can be selectively investigated. The analytical basis for the selective investigation is presented next.

6.2 Identification of the energy-optimal related strategy

The strategy optimization problem consists of an objective function (Subsection 5.2.2) subject to several constraints (Subsection 5.2.3). Relaxation (Subsection 3.4.3) means to simplify an optimization problem by reducing the number of constraints [Günter, 1991] and can be applied for decomposing an optimization problem into optimization subproblems omitting common constraints of subproblems.

An unrelated strategy $l_{p,unrel}$ (Def. 35) omits subsystem dependencies between switching sequences and therefore relaxes the accordant optimization problem (Subsection 5.2.3). Since an unrelated strategy $l_{p,unrel}$ and a related strategy l_p are based on the same switching sequences, it can be hypothesized (Theorem 6.2.1).

Theorem 6.2.1 (Identification of the energy-optimal related strategy)
The related strategy $l_p \in L$ is the energy-optimal related strategy l_{opt} if it holds:
$e(l_{opt}) = e(l_p) \leq e(l_{k,\, unrel})\ \forall\ k > p$ *and* $\nexists\ e(l_j) < e(l_p)$ *with* $1 < j < k$

Proof. *Proving Theorem 6.2.1 true requires the use of Lemma 5.1.3. There, $L_{unrelated}$ is given as finite ordered set. For the minimum energy demand of two unrelated strategies holds: $e(l_{p,unrel}) \leq e(l_{k,unrel})$ with $k > p$. Additionally, Theorem 5.1.5 is required. This theorem implies that $e(l_p) \geq e(l_{p,unrel})$ and is used to distinguish between two cases for $e(l_{opt}) = e(l_p) \leq e(l_{k,\, unrel})$:*
Case 1: $e(l_p) = e(l_{p,unrel})$: This directly results in $e(l_{opt}) = e(l_p) \leq e(l_{k,\, unrel})\ \forall\ k > p$.
Case 2: $e(l_{p,unrel}) < e(l_p) \leq e(l_{k,\, unrel})$: This results in $e(l_{p,unrel}) < e(l_p) \leq e(l_{k,\, unrel})\ \forall\ k > p$:
Since $e(l_j) \geq e(l_p) = e(l_{opt})$ for $1 < j$ and $p < k$, it can be stated that $e(l_{opt}) = e(l_p) \leq e(l_{k,\, unrel})\ \forall\ k > p$ and $\nexists\ e(l_j) < e(l_p)$ with $1 < j < k$.

Theorem 6.2.1 is used to propose a procedure for selective investigation of L which is subject of Section 6.3.

6.3 Procedure for bounded investigation

The objective of this section is to implement a procedure that helps to identify the energy-optimal related strategy in the set of related strategies L [Mechs, 2013a].

Data: Finite ordered set $(L_{unrelated}, \leq) \neq \emptyset$
$e(l_{opt}) = \infty$, l_{opt} = null
Result: Energy-optimal, related strategy l_{opt}, $e(l_{opt})$

for $p = 1; l_{p,unrel}; p$++ **do**
  if $\exists\ l_p$ **then** // if related strategy exists
    if $e(l_p) \leq e(l_{opt})$ **then** $e(l_{opt}) = e(l_p); l_{opt} = l_p$; // save lowest $e(l_p)$
    if $e(l_{opt}) \leq e(l_{p,\, unrel})$ **then** // check investigation stop
      return l_{opt};
    end
  end
end

Algorithm 3: Algorithm for selective investigation of L

A procedure oriented by methods presented in Subsection 3.4.3 is applied here using unrelated strategies, for selective investigation of L. Algorithm 3 uses Theorem 6.2.1 for identifying the energy-optimal related strategy. This algorithm investigates iteratively the set of related strategies L by the use of the minimum energy demand of unrelated strategies. The algorithm implements Theorem 6.2.1 as stop criterion to abort the investigation of L as soon as the energy-optimal related strategy l_{opt} is found. The investigation of L can be aborted as long as if it can be guaranteed that the energy-optimal related strategy is found, which is expressed by Theorem 6.2.1. Algorithm 3 is explained using Example 11 illustrated in Figure 6.1 and Figure 6.2.

Example 11

DATA: *Initially, all unrelated strategies $l_{p,unrel} \in L_{unrelated} \neq \emptyset$ and $p = 1, 2, ..., n$ are generated and the minimum energy input $e(l_{p,unrel})$ is determined. The set of unrelated strategies $L_{unrelated}$ is ordered according to the energy demand of unrelated strategies $e(l_{p,unrel}) \leq e(l_{p+1,unrel})$.*

1^{st} step: The first unrelated strategy $l_{p,unrel}$ is picked and it is tried to generate the corresponding related strategy l_p. The related strategy exists with minimum energy input $e(l_1)$. This represents the (global) lower bound $e(l_{opt}) = e(l_1)$, and $l_{opt} = l_1$ since $e(l_1) < e(l_{opt}) = \infty$. At the end of this step, $e(l_{opt}) = e(l_1)$ is compared to $e(l_{1,unrel})$. Since $e(l_{opt}) > e(l_{1,unrel})$, it is proceeded with step 2.

2^{nd} step: The unrelated strategy $l_{2,unrel}$ is picked and it is tried to generate the corresponding related strategy l_2. Since l_2 is not feasible (for instance because of temporal constraints violated), it is proceeded with step 3.

3^{rd} step: The unrelated strategy $l_{3,unrel}$ is selected and it is tried to generate the corresponding related strategy l_3. Since $e(l_3) < e(l_{opt}) = e(l_1)$, it is assigned to the global lower bound $e(l_{opt}) = e(l_3)$, and $l_{opt} = l_3$. The comparison $e(l_{opt}) \leq e(l_{3,unrel})$ is false, so that the next step is initiated.

4^{th} step: The unrelated strategy $l_{4,unrel}$ is picked and it is tried to generate the corresponding related strategy l_4. Since l_4 is not feasible (for instance because of temporal constraints violated), it is proceeded with step 5.

5^{th} step: The unrelated strategy $l_{5,unrel}$ is selected and it is tried to generate the corresponding related strategy l_5. Since $e(l_5) > e(l_{opt}) = e(l_3)$, it holds $e(l_{opt}) = e(l_3)$, and $l_{opt} = l_3$. The comparison $e(l_{opt}) \leq e(l_{5,unrel})$ is true, so that $l_{opt} = l_3$ is returned.

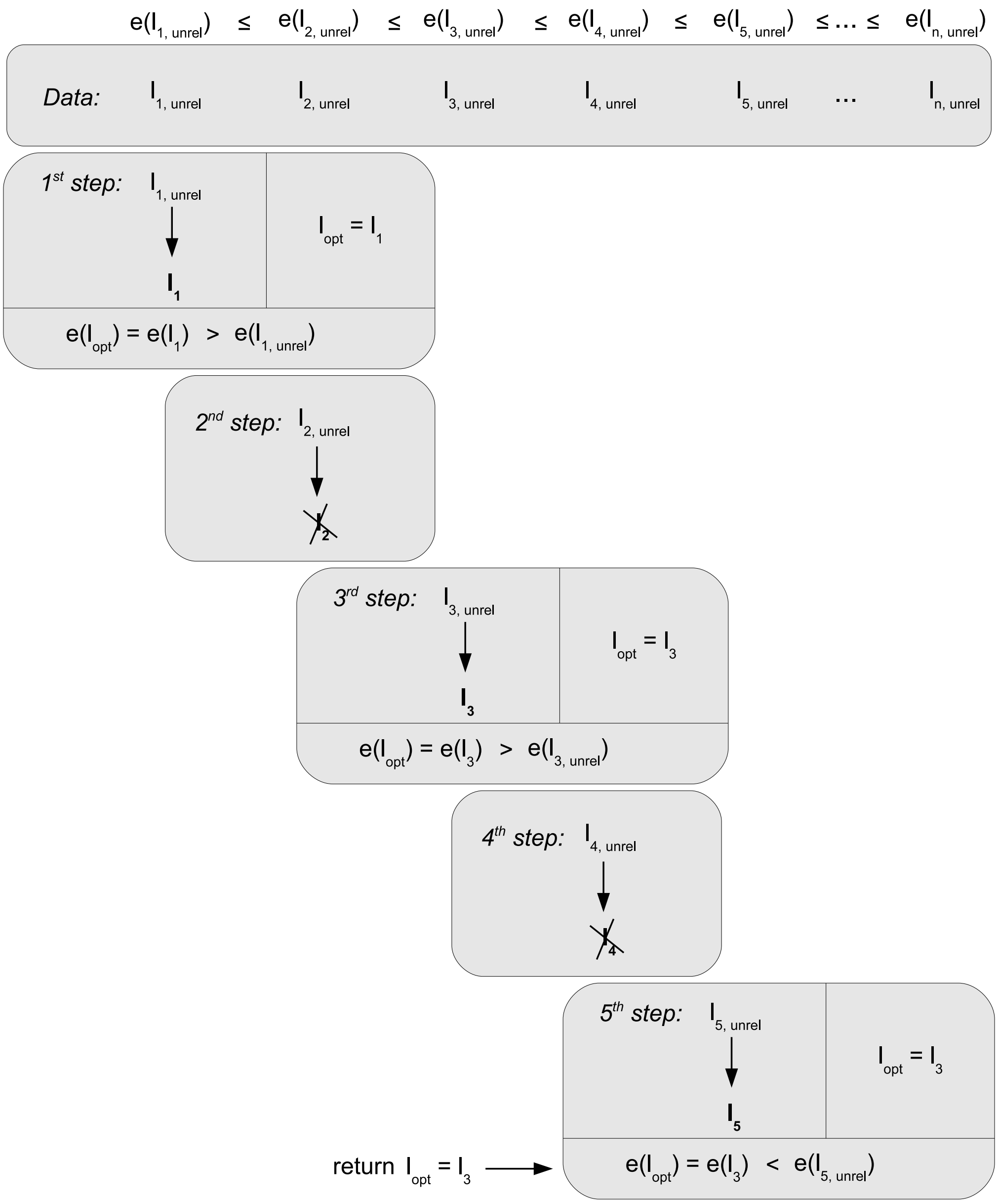


Figure 6.1: Identification of the energy-optimal related strategy l_{opt} after five steps

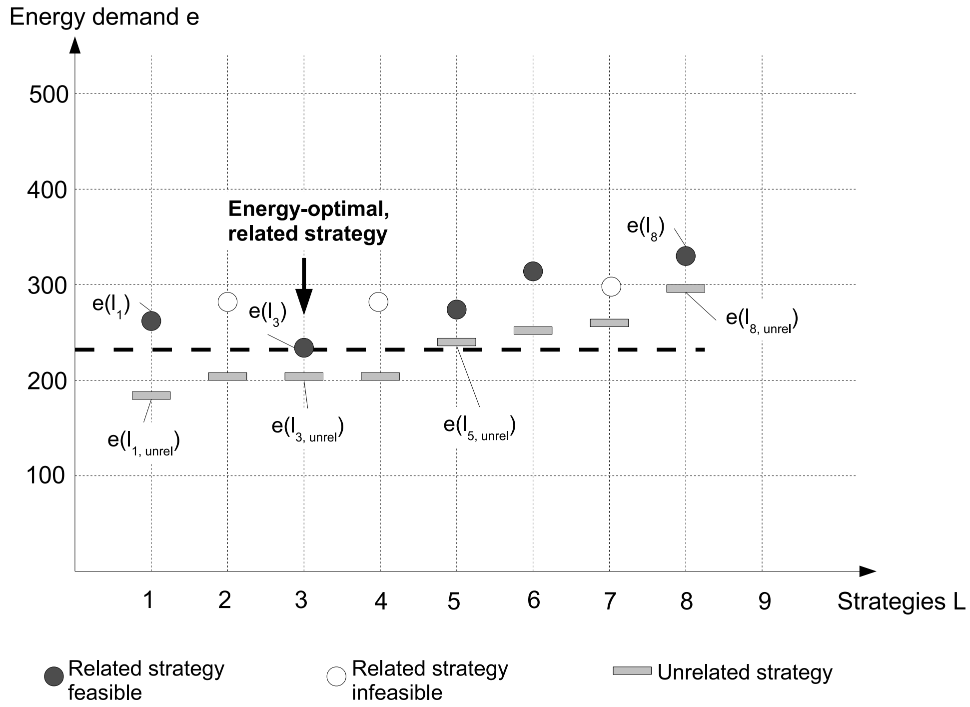


Figure 6.2: Unrelated and related strategies valued by the minimum energy input, | L | = 8

Lemma 6.3.1 (Correctness of Algorithm 3)
Algorithm 3 identifies the energy-optimal related strategy l_{opt}, if at least one feasible related strategy exists, otherwise l_{opt} = null.

Proof. *Given a finite ordered set $(L_{unrelated}, \leq) \neq \emptyset$. $L_{unrelated}$ comprises at least one unrelated strategy $l_{1,\, unrel}$. The unrelated strategy $l_{p,\, unrel}$ is chosen (Line 1). If a corresponding l_p exists (Line 2), its energy value $e(l_p)$ is compared to $e(l_{opt})$. If $e(l_p) \leq e(l_{opt})$ is true (Line 3), then $e(l_{opt}) = e(l_p)$ and $l_{opt} = l_p$ is assigned. Two different cases can be distinguished for the comparison in Line 4:*

- *Case 1: $e(l_{opt}) \leq e(l_{p,\, unrel})$: comparison is true, the investigation is aborted by returning l_{opt}*
- *Case 2: $e(l_{opt}) > e(l_{p,\, unrel})$: comparison is false, the next unrelated strategy $l_{p+1,\, unrel}$ is investigated, if it exists (Line 1).*

If for all $l_{p,\, unrel}$ with p = 1, ..., n, no feasible related strategy can be identified, Algorithm 3 returns l_{opt} = null.

Lemma 6.3.2 (Termination of Algorithm 3)
Algorithm 3 terminates after a finite number of steps.

Proof. *The loop stated in Line 1 terminates at the latest after* n *steps with p = 1, ..., n as the number of unrelated strategies in $L_{unrelated}$.*

Lemma 6.3.3 (Runtime complexity of Algorithm 3)
Algorithm 3 has worst-case runtime complexity $\mathcal{O}(L) = \prod_{i=1}^{n}[Seq^i]$ (Equ. 6.1).

Proof. *In the worst case, investigating L, Line 1 in Algorithm 3 is called n-times. The power of the set of unrelated strategies $\mid L \mid$ is the Cartesian product of subsystem alternatives (Def. 35).*

6.4 Summary

In this chapter, the mathematical basis has been established for bounded investigation of the set of related strategies L (Section 6.2). Theorem 6.2.1 guarantees to find the energy-optimal related strategy after k-steps with $1 < k \leq n$. The investigation of L can be aborted if $\mathrm{e}(\mathrm{l}_{opt}) = \mathrm{e}(\mathrm{l}_p) \leq \mathrm{e}(\mathrm{l}_{\mathrm{k,\, unrel}})$.
This is implemented in the procedure proposed in Section 6.3. The procedure aligns to dynamic programming proposed in [Bellman, 1957] where problems are solved by decomposing techniques. Unrelated strategies, which are related strategies with relaxed constraints, serve as decomposing instrument for investigating most promising related strategies first. Based on this approach, energy-optimal related strategies can be computed by using structural knowledge about the problem (Section 4.1). In the worst case (k = n), to each unrelated strategy, the

corresponding related strategy has to be generated if it exists.
In Section 9.1 of Part III, the computational efficiency of the procedure introduced in this chapter is evaluated. The efficient identification is an essential part of the applicability of the proposed analytical procedure for practical purposes.
The proposed procedure of this chapter is a core element of a framework for strategy execution and supervision presented in Chapter 7. Such a framework is necessary for actually realizing and executing the energy-optimal related strategy in the system.

Chapter 7

Framework for robust execution of strategies

After having identified the energy-optimal related strategy, it is of interest to execute this strategy in the automation system. For this purpose, a framework for robust execution of strategies is proposed in this chapter (Fig. 7.1). This framework [Mechs, 2013b] is implemented offering a software tool for setting up system models (Chapter 4), the identification of the energy-optimal related strategy (Chapter 6), and the execution and supervision of computed strategies in the automation system.
The framework links the engineering phase (strategy generation based on a system model) to runtime aspects (strategy execution in the automation system). The elements of the framework are referenced in the following list and in Figure 7.1 enabling strategies to be computed and to be executed in the automation system:

Section 7.1 The automaton-based model of the automation system (Chapter 4) is generated using a graphical user interface.

Section 7.2 A mode-based description is transformed into a platform-dependent control program specification. This control program specification can directly be compiled into machine code to be executable in the CU of the subsystem.

Section 7.3 Using the automaton-based system model, computerized strategies are derived based on the procedure presented in Chapter 6.

Section 7.4 In order to make strategies robustly executable a modification step towards a robust execution of a strategy is proposed.

Section 7.5 The strategy execution and supervision manages the execution of the executable strategy in the CU.

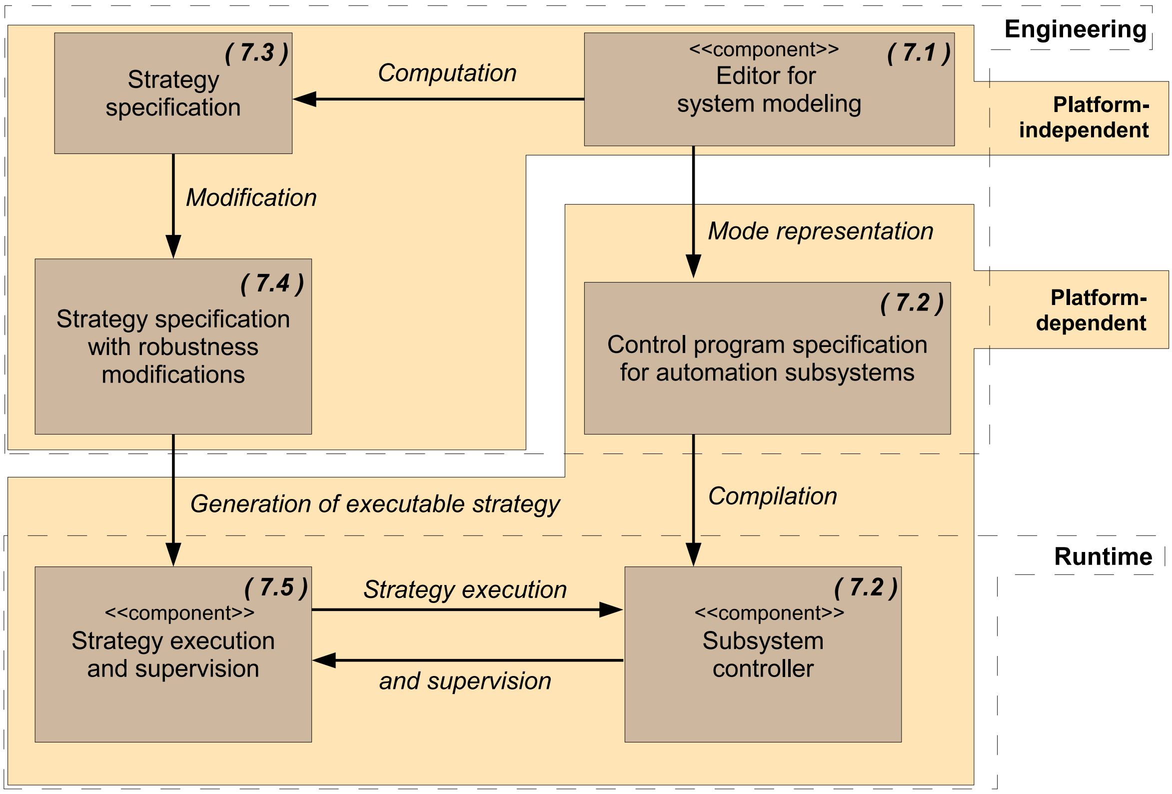


Figure 7.1: Framework for strategy execution

7.1 Engineering of the system model

A graphical user interface supports the generation of the system model of networked automation subsystems. The system model represents the temporal, energetical, and structural properties of the automation system. The graphical user interface for editing the model of Subsystem sub_1 is shown in Figure 7.2 (a). Properties of transitions (for instance time guards) and properties of modes (for example input power annotations) can be edited in Figure 7.2 (b) respectively Figure 7.2 (c). It is distinguished between operating and transitional modes. For operating modes, the delay within the mode can be determined by the strategy execution and supervision (Section 7.5) whereas for transitional modes, the delay within the mode is given by the subsystem and can only be monitored (Section 7.2). The structural information like subsystem dependencies are modeled between a pair of subsystems as illustrated in Figure 7.3.

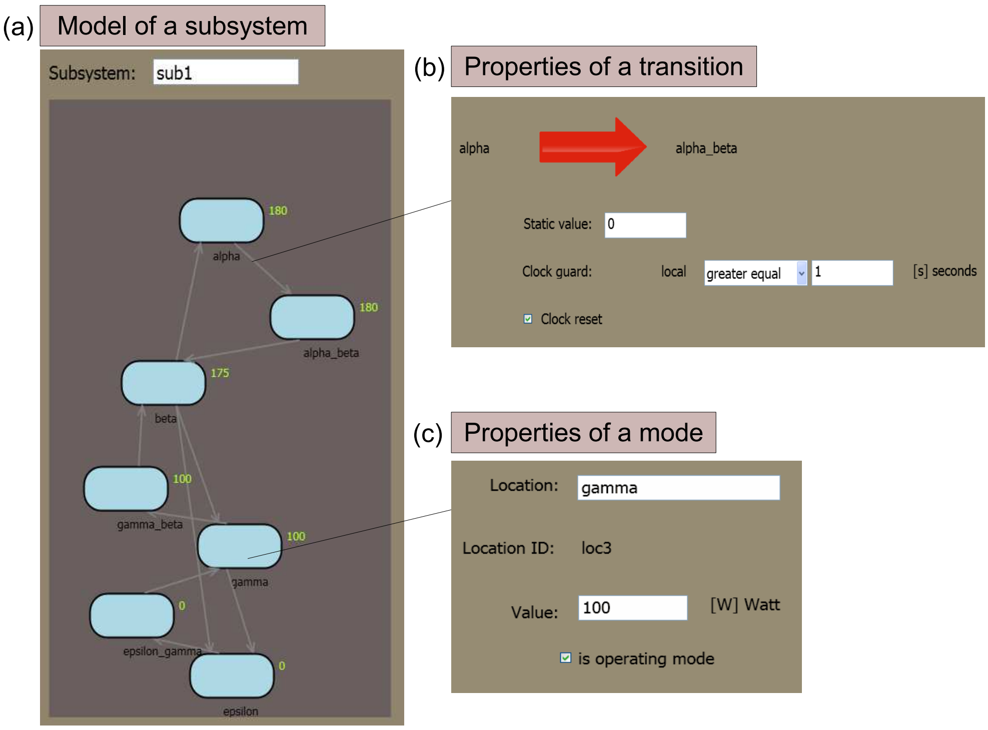


Figure 7.2: Editing the model of Subsystem sub_1

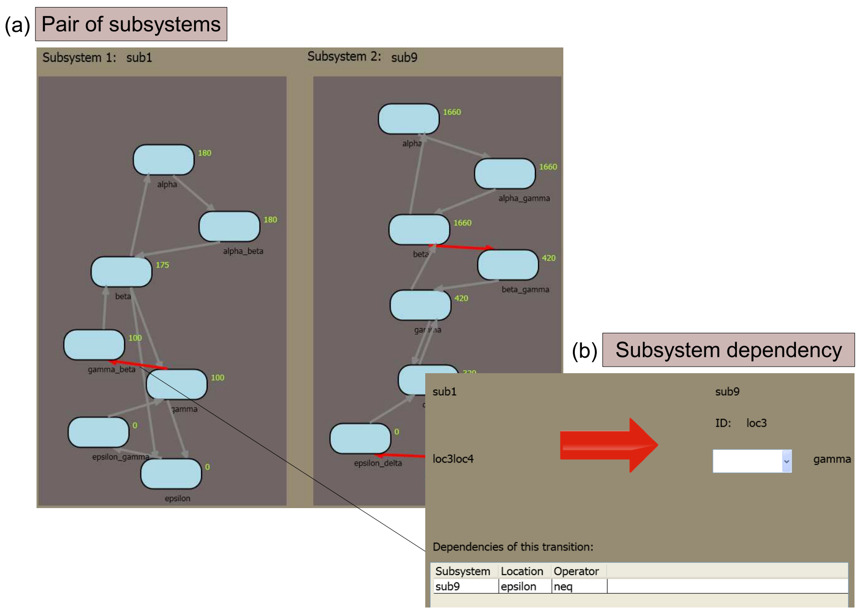


Figure 7.3: Editing subsystem dependencies between Subsystem sub_1 and Subsystem sub_2

7.2 Control program specification

The structural and behavioral aspects of the automation system are determined in a platform-independent way during engineering. Each subsystem of the automaton-based system model has a correspondent in the automation system. Each subsystem in the automaton-based system model has to represent the behavior of each subsystem in the automation system (*Mode representation* in Fig. 7.1). This implies to specify the control program of each automation subsystem according to the platform-independent subsystem behavior stated in the automaton-based system model (Section 7.1). In this context, only the elements of the system model, which are necessary for the behavioral description to run the control program, are mapped.

On subsystem side (CU), inputs and outputs are read and updated in each cycle. Between reading inputs and updating outputs, the CU spends time for computation. For the three steps, reading inputs, computing, and writing outputs, the maximum time, called maximum *scan or cycle time*, is predefined. For a detailed introduction to the operation of PLCs, see [Bolton, 2009], [Wellenreuther, 2008].

The control program specification used here complies with the international standard IEC 61131-3 [IEC, 2003]. IEC 61131-3 specifies the syntax and semantics of programming languages for programmable controllers (CU) in industrial automation. The standard defines textual (*structured text* (ST), and *instruction list* (IL)) and graphical languages (*ladder diagram* (LD), and *function block diagram* (FBD)), as well as *sequential function charts* (SFC). The IEC 61131-3 conform control program must reflect the modes of the system model in Section 7.1. Modes can be interpreted as encapsulation of specific control actions. In order to encapsulate the mode-based control program in form of a state machine, a Function Block (FB) is used. A FB is a function with a dedicated memory space called instance Data Block (DB) in the CU of the automation subsystem.

The state machine accepts three input variables: *swCommand*, *delay*, *inputx*. The variable *swCommand* accepts switching commands for indicating the desired mode the automation subsystem should be in. The desired delay within a mode is represented by the variable *delay*. Variable *inputx* represents a specific input value received by the controlled hardware.

As output, the state machine provides the variables *enteringMode*, *finishingMode*, *transTime*, and *outputx*:

- *enteringMode*: current mode ID while entering a mode
- *finishingMode*: current mode ID representing the mode ready to be left
- *transTime*: transitional time as difference between end time and start time of a transitional mode
- *outputx*: output variable whose value is sent to the controlled hardware

The internal timer and variables used are:

- *TimerSM1*: timer based on the internal clock of the CU
- *tr_startTime*, *tr_endTime*: time variables to compute transitional times (durations)
- *tr_startDInt*, *tr_endDInt*: integer variables to compute transitional times (durations)
- *measure*: boolean variable as indication if transitional time should be measured

The initialization of variables is conducted via the instance DB: *swCommand := 0, enteringMode := 0, finishingMode := 0, measure := true, inputx := false, outputx := false*

The example FB of the mode representation in the IEC 61131-3 conform control program in form of a state machine is given by the following program based on Structured Control Language (SCL) including two different modes (one operating mode 1, and one transitional mode 2). Case 1 (operating mode 1) and Case 2 (transitional mode 2) have correspondents in the model illustrated in Figure 7.2 (a). The detail shown in the FB represents the platform-dependent interface to the modes available or addressable in an automation subsystem.

State machine detail representing modes:

```
FUNCTION_BLOCK FB200

VAR_INPUT
    swCommand : DWORD;               // required mode ID
    delay : TIME;                    // variable delay
    inputx : BOOL;                   // input from the controlled hardware
END_VAR

VAR
    TimerSM1 : SFB4;                 // timer
    tr_startTime : TIME;             // duration start time
    tr_endTime : TIME;               // duration end time
    tr_startDInt : DINT;             // duration start time as integer
    tr_endDInt : DINT;               // duration end time as integer
    measure : BOOL;
END_VAR

VAR_OUTPUT
    enteringMode : DWORD;            // subsystem has entered mode
    finishingMode : DWORD;           // subsystem has finished mode
    transTime : DINT;                // transitional time
    outputx : BOOL;                  // output to the controlled hardware
END_VAR
```

```
CASE DWORD_TO_INT(swCommand) OF
    0: // initial mode 0
        enteringMode := 0;
        finishingMode := 0;
    1: // operating mode 1
        enteringMode := 1;                      // entering mode 1
        TimerSM1(IN := true, PT := delay);      // variable delay: x seconds
        IF TimerSM1.Q = 1 THEN
            TimerSM1(IN := false);              // reset timer
            finishingMode := 1;                 // finishing mode 1
        END_IF ;
    2: // transitional mode 2
        enteringMode := 2;              // entering mode 2
        IF measure = true THEN          // start time of transitional mode
            tr_startTime := TIME_TCK();
            measure := false;
        END_IF;
        tr_endTime := TIME_TCK();       // end time of transitional mode

        outputx := true;

        IF inputx = true THEN           // input from the controlled hardware
            // compute transitional time
            tr_endDInt := TIME_TO_DINT(tr_endTime);
            tr_startDInt := TIME_TO_DINT(tr_startTime);
            transTime := tr_endDInt - tr_startDInt;
            measure := true;
            finishingMode := 2;         // finishing mode 2
        END_IF ;
END_CASE ;

END_FUNCTION_BLOCK
```

If the switching command *swCommand = 1*, the variable *enteringMode = 1* is updated. The timer *TimerSM1* is started with the given delay. After finishing the timer, the timer is reset and the variable *finishingMode* is updated.

If Case 2 is triggered in the state machine (*swCommand* = 2), the variable *enteringMode = 2* is updated. Initially, the variable *measure* is true, so that the current subsystem time is written to variable *tr_startTime* while measure is set to false. The variable *tr_endTime* is updated with every CU cycle. After that, the output *outputx* is set which corresponds to sending a state change command to the controlled hardware. As soon as the controlled hardware changes its state, the variable *inputx* is set to true, so that the transitional time *transTime* is calculated from values *tr_endTime* and *tr_startTime*. In this way, the actual transitional time can be computed and monitored. The variable *finishingMode* is updated at the end.

Having introduced the editor for generating the automaton-based system model and the corresponding mode-based representation in the CU, computerized strategies can be specified. These strategies are computed based on the information in the automaton-based system model and are executed in the CU.

7.3 Strategy specification

In order to specify the energy-optimal related strategy, some parameters need to be provided (Subsection 7.3.1). A resulting strategy is presented in Subsection 7.3.2.

7.3.1 Parametrization for strategy computation

Identifying a strategy requires parameters like a pause interval as well as the initial and target modes for each subsystem (Fig. 7.4).

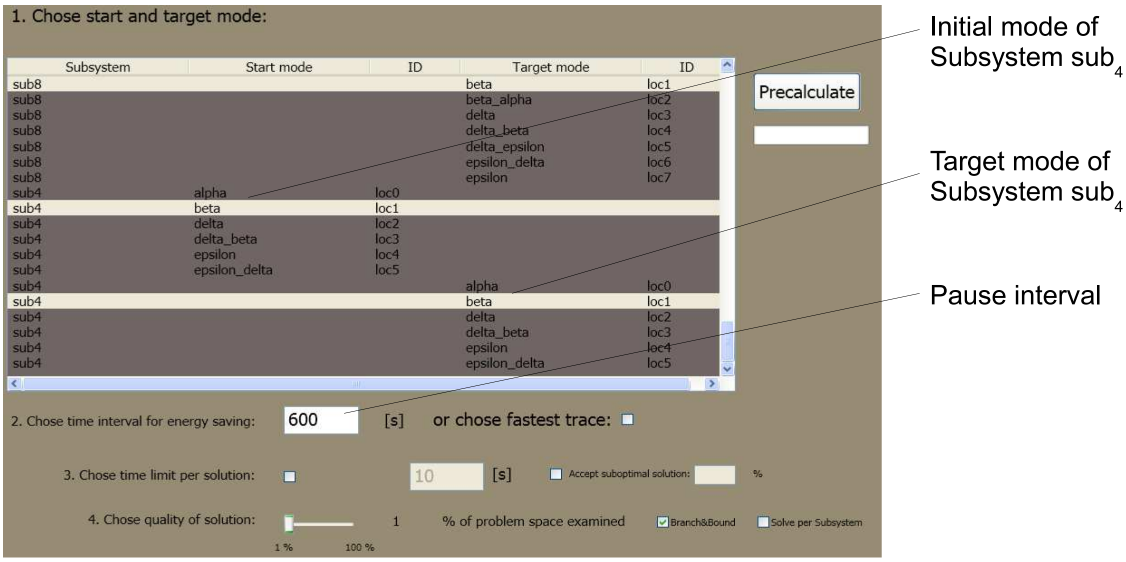


Figure 7.4: Parametrization before identifying an energy-optimal related strategy

The approach for identifying an energy-optimal related strategy has been proposed in Chapter 5 and Chapter 6. Based on this approach and the provided parametrization (Fig. 7.4), the set of alternative switching sequences Seq^i for each subsystem sub_i is determined (Subsection 5.1.2). After identifying the sets of switching sequences Seq, the sets of unrelated and related strategies (Subsection 5.1.3) are computed for the given parametrization.

The optimization based on the formulation in Section 5.2 for computing optimal switching sequences and strategies is implemented based on the C# .NET library of the IBM ILOG CP Optimizer, version 12.4. The CP (Constraint Problem) Optimizer can solve combinatorial optimization problems with discrete decision variables [IBM, 2010]. The solver integrates techniques for solving scheduling problems with constraint programming techniques. It enables the formulation of high-level structures resulting in the generation of compact models. Constraint programming is especially suited for answering feasibility questions of strongly constrained discrete manufacturing [Harjunkoski, 2002]. "The effectiveness of CP depends on the propagation mechanism behind constraints" [Méndez, 2006, page 938], for instance constraint propagation [Fromherz, 2001]. Constraint propagation can effectively reduce the domain of the variables for finding efficiently (optimal) solutions. For guiding the search for optimal related strategies, the optimization engine receives initially a proposal for assignments to decision variables in order to obtain a first feasible solution. Once a feasible solution is found by the solver, this solution is improved step by step resulting in a computerized strategy for the automation system.

7.3.2 Resulting strategy

A strategy specification (Def. 35 and Def. 36) contains the information about the sequential order of modes of all subsystems respecting temporal and structural (mode preconditions) constraints in form of tuples of switching sequences (Fig. 7.5).

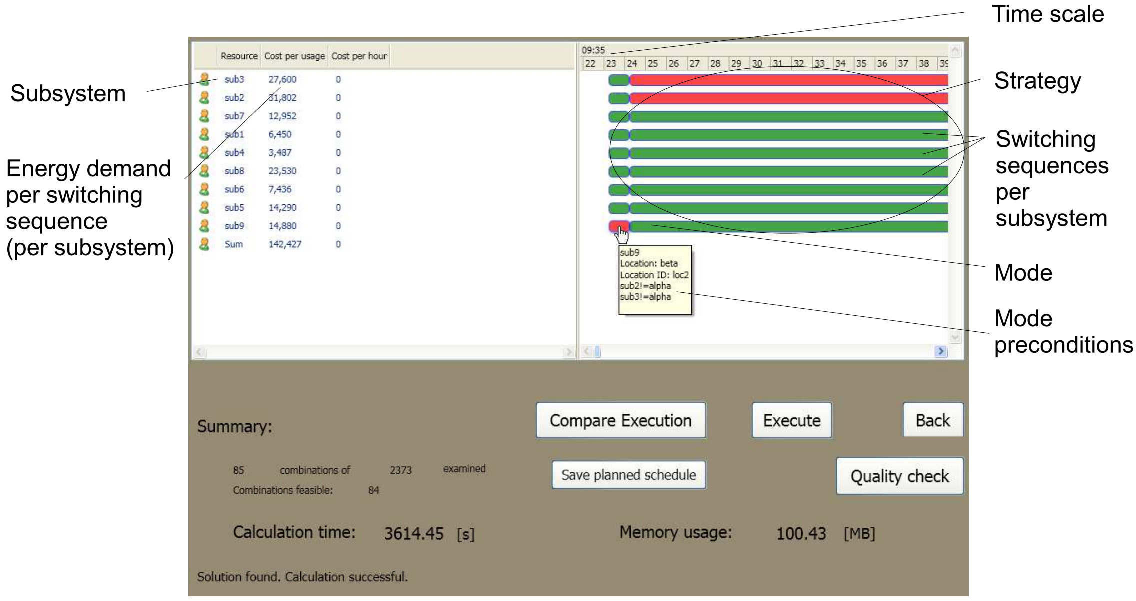


Figure 7.5: Energy-optimal related strategy in Gantt chart representation for given parameters

Explaining further concepts, an example strategy is used (Example 12).

Example 12

The specification of an example strategy (based on two switching sequences) is given in Figure 7.6. As a result of strategy computation, for each mode in the strategy, the start and end times are determined. These start and end times are visualized using the timed automata representation.

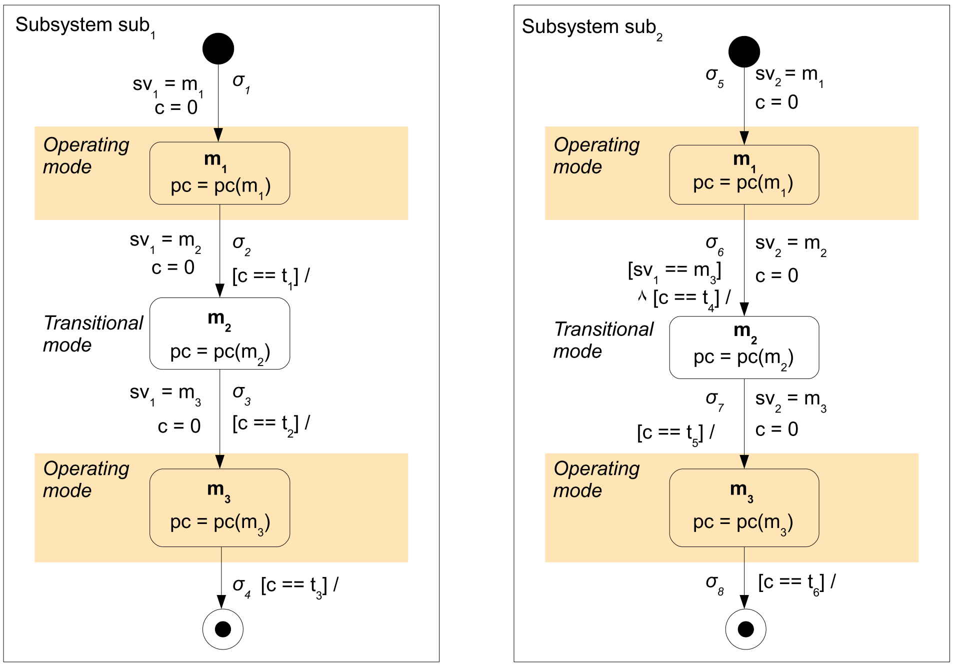


Figure 7.6: Graphical representation of a related strategy (seq_{11}^{1}, seq_{11}^{2})

Based on the calculated strategy, the resulting timed automata in Figure 7.6 represent a specification of a strategy to be executed. Each mode of a timed automaton owns exactly one incoming and exactly one outgoing edge. An additional clock guard is added at each outgoing edge of the operating modes as a result of the strategy computation conducted in Subsection 7.3.1.

7.4 Strategy specification with robustness modifications

A detail of a strategy (Fig. 7.6) is computed based on a given system model. The model information can differ from the system. Model-to-system differences regarding time information can result in the strategy not being executable in the system. This may be caused by subsystem dependencies not being met. However, as a priority, subsystem dependencies need to be met. The adherence to specified temporal constraints is subordinated to ensure the adherence to subsystem dependencies. Because of model-to-system deviations regarding temporal aspects,

robustness modifications of the strategy specification (Section 7.3) need to be made proposed by a *blocking concept* presented in this section.
In order to make execution robust towards temporal model-to-system deviations, this concept is exemplified using Figure 7.7. Model-to-system deviations are especially critical if one subsystem requires another subsystem being in a specific mode for a mode changeover (Fig. 7.7, Subsystem sub_2, guarded transition from m_1 to m_2: [$sv_1 == m_3$]).

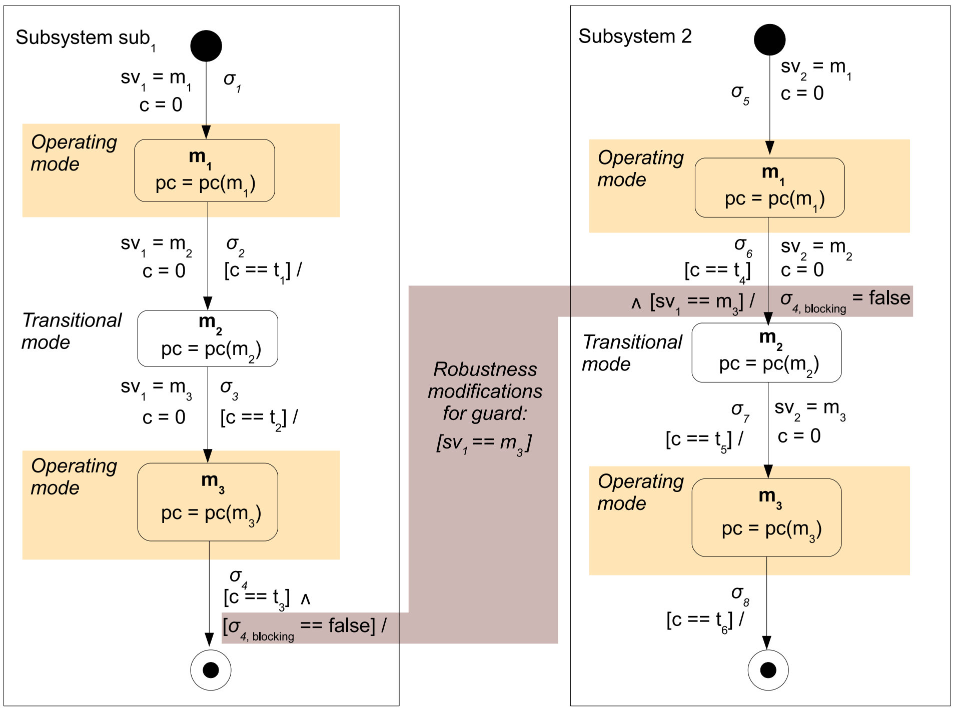


Figure 7.7: Specification of a robustly executable strategy

The feasibility of strategies needs to be ensured even if temporal assumptions (materialized in clock guards) exhibit deviations between system and model regarding transitional times between modes. For execution purposes, the adherence to temporal constraints (clock guards) is relaxed. Especially the delay in an operating mode is not determined by the subsystem, so that the delay in an operating mode can be adjusted during execution of a strategy. Operating modes can be left before or after the specification given in a clock guard.

Example 13

In Figure 7.7, operating mode m_3 in Subsystem sub_1 can be left before or after t_3 while executing the strategy.

Temporal constraints of operating modes are subordinated to the adherence to subsystem dependencies providing flexible execution. In order to meet subsystem dependencies, a boolean shared variable $\sigma_{p,\text{ blocking}}$ for communicating the critical dependency between two subsystems is introduced. This boolean variable is used in the following way:

1. The boolean variable is initially true:
   $\sigma_{\mathrm{p,\,blocking}} = \mathit{true}$
2. The outgoing transition (Subsystem sub_2: from mode m_1 to m_2) which references the shared variable in the guard [$\mathrm{sv}_1 == \mathrm{m}_3$] resets the boolean variable:
   $\sigma_{\mathrm{p,\,blocking}} = \mathit{false}$
3. The outgoing transition of the *referenced mode* m_3 in Subsystem sub_1 is labeled with a guard referring to the boolean variable $\sigma_{\mathrm{p,\,blocking}}$ in the form of:
   $\left[\sigma_{\mathrm{p,\,blocking}} == \mathit{false}\right]$

Example 14
Without a shared blocking variable, while executing the related strategy presented in Figure 7.6, it may happen that mode m_3 in Subsystem sub_1 is left before the transition from mode m_1 to m_2 in Subsystem sub_2 is enabled. The subsystem dependency expressed as guard [$sv_1 == m_3$] would be violated. This is given, if $t_1 + t_2 + t_3 < t_4$.
To avoid such a violation, the guard $\left[\sigma_{p,\,blocking} == \mathrm{false}\right]$ on shared variable $\sigma_{p,\,blocking}$ prevents Subsystem sub_1 leaving mode m_3 before Subsystem sub_2 has entered mode m_2. This ensures the compliance with the subsystem dependency [$sv_1 == m_3$] at transition from mode m_1 to m_2 in Subsystem sub_2. Adherence to time constraints formulated in guards [$c == t_1$], [$c == t_2$], and [$c == t_3$] is subordinated to the observance of subsystem dependencies.

7.5 Strategy execution and supervision

Before carrying out strategies, the sequences of modes within a strategy are transformed into job sequences of Quarz.NET task scheduler, version 2.0.1 [QuartzNET, 2012]. Quarz.NET task scheduler is capable to execute job instances at a predefined point in time and with a given duration. The term *job* and *mode* are used synonymously here. Mode specifications (delays for instance) are mapped to jobs of the task scheduler. For each mode/job (Fig. 7.8 (a)) the preconditions (for instance the precedence constraints of modes) and the length of a mode respectively a job are defined and checked before execution (b) and (c). A job is initiated using triggers (d). The implementation takes care of triggering the jobs in the determined sequence respecting temporal constraints and subsystem dependencies as predefined by the strategy. After triggering a mode, the mode is active in the subsystem (e). After a specific time period, the mode has finished (f) and is executed (g). The finished execution of a mode has impacts on the preconditions of other modes, suggested by (h). The remaining modes to be executed pass the stages of Figure 7.8 the same way.
Figure 7.9 represents the SCL-implemented modes in the example FB graphically (Section 7.2). It illustrates the interaction between the CU of the subsystem and the strategy execution and supervision (Fig. 7.1). A time line is given for mapping events to points in time.

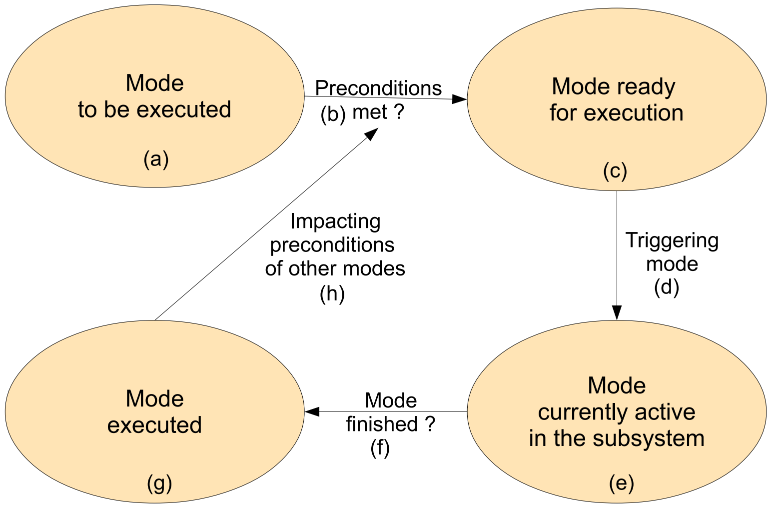


Figure 7.8: Stages of mode/job execution

Example 15
Initially, the integer variables swCommand, enteringMode, *and* finishingMode *are set to zero. The strategy execution and supervision starts to trigger* mode m_1 *by sending* swCommand = 1 *with* delay = t_1 *to the system.* Mode m_1 *is known to be an* operating mode, *so that a* delay *is provided by the strategy execution and supervision. The subsystem enters* mode m_1 *after receiving the input* swCommand = 1 *and updates the variable* enteringMode = 1. *It stays for a time period of* delay = t_1 *in* mode m_1 *by using a timer. After having finished the timer, the subsystem updates* finishingMode = 1. *This variable is read by the strategy execution and supervision and is informed about the subsystem having rested in* mode m_1 *during the time period* delay = t_1. *Afterwards, the next mode in the strategy specification can be triggered. The scheduler sets* swCommand = 2 *to cause the system change to* mode m_2. *A delay is not sent form the strategy execution and supervision to the subsystem because* mode m_2 *is known to be a* transitional mode. *The transition time* transTime *is determined by the subsystem* (transTime = t_2) *and cannot be specified by the strategy execution and supervision. After receiving the* swCommand = 2, *the subsystem enters* mode m_2 *by setting* enteringMode = 2. *The subsystem stays in* mode m_2 *for a time period* transTime = t_2 *determined by the subsystem. After completion, the subsystem sets* finishingMode = 2 *which informs the strategy execution and supervision about the successful termination of* mode m_2. *The next mode is triggered by* swCommand = 3.

7.6 Summary

A framework (implemented in C# .NET 4.0) for robust execution of strategies has been presented in this chapter. This framework uses the approach presented in Chapter 4, Chapter 5, and Chapter 6. In Section 7.1, a graphical user interface is given for systems engineering based

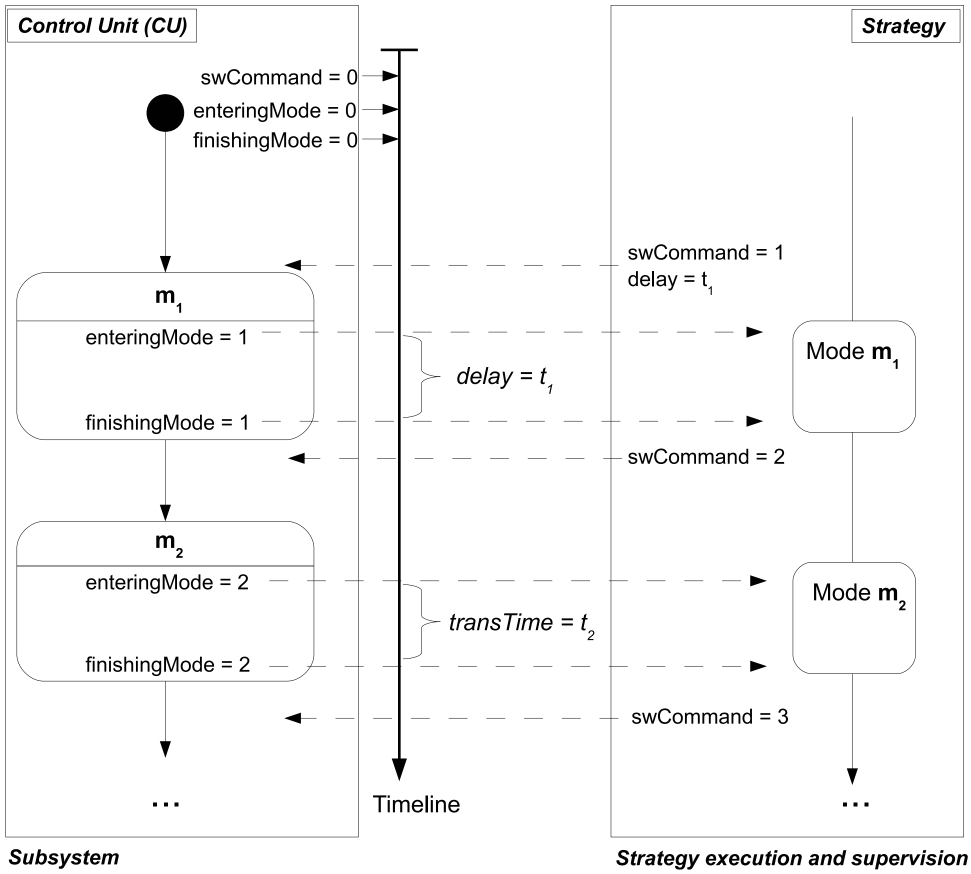


Figure 7.9: Execution and specification of strategies based on state machine representation (detail of implementation)

on the content of Chapter 4. The mode representation of the system model in the control programs is given in Section 7.2. Consequently, the model of the subsystems and the subsystems comprise the same mode set enabling the use of computerized strategies.

Based on the system model, strategies are derived using parametrization in Section 7.3. A state-of-the-art tool is applied identifying the energy-optimal related strategy using the procedure of Chapter 6. Accounting for model-to-system deviations, the identified strategy is modified for robust execution in Section 7.4. Temporal constraints are subordinated to the observance of subsystem dependencies during execution.

The strategy execution and supervision (Section 7.5) uses the strategy specification for robust execution and generates jobs for each mode in the strategy. Considering preconditions, jobs representing the modes are executed, using the task scheduler, according to the specified strategy in the CU. After execution of the strategy in the CU, the planned strategy and an executed strategy can be compared which is necessary for evaluation purposes presented in Section 9.2 and Section 9.3 of Part III.

The evaluation (Part III) of this thesis relies on the framework proposed in this chapter. The framework enables and supports the engineering of the system model (Contribution 1), the computation of strategies (Contribution 2 and Contribution 3), the specification of feasible strategies (Contribution 4), the model validation using strategies (Contribution 5), and consequently the reduction of energy demands within unproductive phases (Contribution 6).

Part III

Evaluation and presentation of results

Chapter 8

Methodology and test environment for evaluation

Part III presents the results regarding the assessment of the proposed approach of Part II. Chapter 8 introduces the applied methodology (Section 8.2) and the test environment for evaluation (Section 8.3). Using specific methods, the approach of Part II is evaluated against the capability to contribute to the research objectives given in Section 1.3 (Section 8.1). Chapter 9 presents the results of the evaluation.

8.1 Evaluation perspectives and objectives

This section introduces the evaluation perspectives and objectives relevant for checking the capability of the proposed approach in Part II regarding the contribution to the research objectives defined in Section 1.3. This comprises to evaluate the approach regarding the capability to identify optimal strategies. Furthermore, the approach has to contribute to the specification of feasible strategies, the model validation using strategies, and the reduction of energy demands by strategies. The framework presented in Chapter 7 supports the evaluation.

Identification of optimal strategies (Section 9.1)
The computation of strategies has to be applicable in industrial settings. The computability of strategies with regard to industrial systems is a key aspect to be ensured with special focus to runtime constraints. Influencing factors on the computation of optimal strategies have to be identified and used for a performance evaluation. Special specificities of input parameters enable statements about computational performance for strategies in similar systems. The tractable procedure of Chapter 6 is checked for the capability to identify optimal strategies.

Specification of feasible strategies (Section 9.2)
A feasibility check must be conducted providing credibility regarding the execution of strate-

gies in the automation system. As intensifying factor, strategies are computed based on models incorporating incomplete knowledge about the automation system. Consequently, strategy execution requires to cope with model-to-system deviations.

Model validation using strategies (Section 9.3)
Models abstract from system details and simplify aspects of the system which enables to reduce complexity. Abstracting from details makes system predictions and statements manageable. Additionally, modeled information is created at a specific time point. Therefore, the model information is static. The static and abstracting view has implications on the validity of the strategy prediction towards energy demand. On this account, the validity the system model is examined and quantified.

Reduction of energy demands by strategies (Section 9.4)
Strategies need to reduce the energy demand of an automation system considerably within unproductive phases. Applying strategies is justifiable if the energy efficiency can be essentially increased for unproductive phases. Therefore, the evaluation has to prove the economic benefit of strategies with regard to energy demand. The effectiveness of the proposed approach is quantified as the degree of achieved energy savings.

8.2 Methods for evaluation

In this section, relevant methods for evaluating the proposed approach with regard to the research objectives of this thesis are presented. It is distinguished between formal and analytical methods (Subsection 8.2.1) respectively experimental methods (Subsection 8.2.2).

8.2.1 Formal and analytical methods

Formal and analytical methods are based on a formal or mathematical problem description. Two relevant methods of formal and analytical evaluation are presented next and are discussed with regard to applicability to the problem context of this thesis: *(semi-)automated model checking* and *mathematical verification*.

Applicability of automated model checking

Model checking is a formal verification technique that checks analytically if a model M (often automaton model) complies with a specification [Larsen, 1995]. If the model satisfies the specification, model checking provides a certificate as proof of correctness. If not, the algorithm stops and provides a counter example proofing that the model is not consistent with regard to the specification.

This (semi-) automated approach is especially interesting for specific properties to be ensured for a network of automation subsystems. Proving the temporal reachability of modes and showing the absence of deadlocks for all possible strategies as a safety property, both, increase the confidence in the model. However, in practice, today's model checking is not efficiently computable [Bengtsson, 1996], [Bozga, 1998], [Beyer, 2003] for medium-size systems (Test Bed tb_M, Subsection 8.3.2) stated by [Dierks, 2006, pages 5, 60]. Model checking proves the absence of faults in a model (for instance deadlocks) in the model of the automation system. Strategy feasibility aims on another verification objective. However, verifying the strategy execution despite of model-to-system deviations requires to incorporate a system view.

Analytical verification

Mathematical verification checks the correctness of assumptions using analytical procedures. Mathematical proving is applied in this thesis to give evidence of the correctness of theorems and a lemma (for instance in Subsection 5.1.3) and the correctness of algorithms (Subsection 5.1.2, Section 6.3).

8.2.2 Experimental methods

In contrast to formal and analytical methods, experimental methods are designed to reveal the causality between input parameters and output parameters. Output parameters are often equal to the objective parameters of the experiment. Two main experimental methods are discussed which are relevant in this thesis: *direct experiments* and *simulation-based experiments*.

Direct experiments

Direct experiments require tests in an existing physical automation system. They can give evidence of technical feasibility of concepts. Therefore, a Test Bed tb_S (Subsection 8.3.1) is used to show the technical realization of the approach of this thesis (Part II). Direct experiments conducted in Test Bed tb_S provide a proof of applicability in industrial automation. Moreover, direct experiments are required for analyzing a system model towards validity and feasibility of strategies (Subsection 9.3.2). Thereby, a system model can be fine-tuned according to a given system. Furthermore, the feasibility of strategies (Subsection 9.2.1) and the energy demand reductions possible by using the proposed approach (Subsection 9.4.1) are evaluated based on direct experiments in Test Bed tb_S.
Since a physical system (Test Bed tb_S for instance) provides a specific system behavior and a certain structure, there exist some limitations on the practicability of direct experiments. In order to test different settings, the change of system behavior and the reorganization of the structure is linked to high expenditures in the physical system. Furthermore, conducting direct experiments in a working manufacturing system are often too costly to be executed because

this reduces availability for production purposes. Experiments that disturb or even interrupt the running production are not acceptable because this would reduce the productive output. Besides this, laboratory conditions in order to keep low environmental changes (e.g. temperature or air moisture) can hardly be guaranteed for industrial manufacturing systems. This affects the repeatability of direct experiments. Additionally, during engineering, the physical automation system is (partially) not available for tests so that simulation-based approaches must be applied.

Simulation-based experiments

Simulation-based testing conducts experiments in a setting where the system is mimicked by a simulation system. In this thesis, simulation is applied using the model of a medium-size Test Bed tb_M (Subsection 8.3.2). The control units (CU) of the Test Bed tb_M are mimicked by an emulator.
There exist various reasons for experiments not being available to be conducted directly in the physical system. First, the approach of this thesis should enable tests while the system is not physically present. In this way, occurring problems can be analyzed during system engineering before operation of the automation system. Consequently, simulation-based experiments help towards testing the early designs of the automation system. Additionally, simulation is a (cost-)efficient way to test the impact of different input parameters on the reaction (output parameters) of the simulation system. On the one hand, using simulation-based testing ensures reproducibility of testing results which is not necessarily true for direct experiments. On the other hand, the system behavior and the system structure can be adjusted easily during engineering so that different system settings can be virtually tested. The method of simulation is applied for evaluation presented in Section 9.1, Section 9.2, Section 9.3, and Section 9.4.

8.3 Test environment

The direct and simulation-based experiments conducted in this thesis rely on a test environment. Test Bed tb_S is used for direct experiments (Subsection 8.3.1). In case of simulation-based experiments, the more extended setting of Test Bed tb_M is applied (Subsection 8.3.2). The test environment is based on the framework which has been introduced in Chapter 7. The test environment is the combination of this framework with Test Bed tb_S respectively Test Bed tb_M illustrated in Figure 8.1.

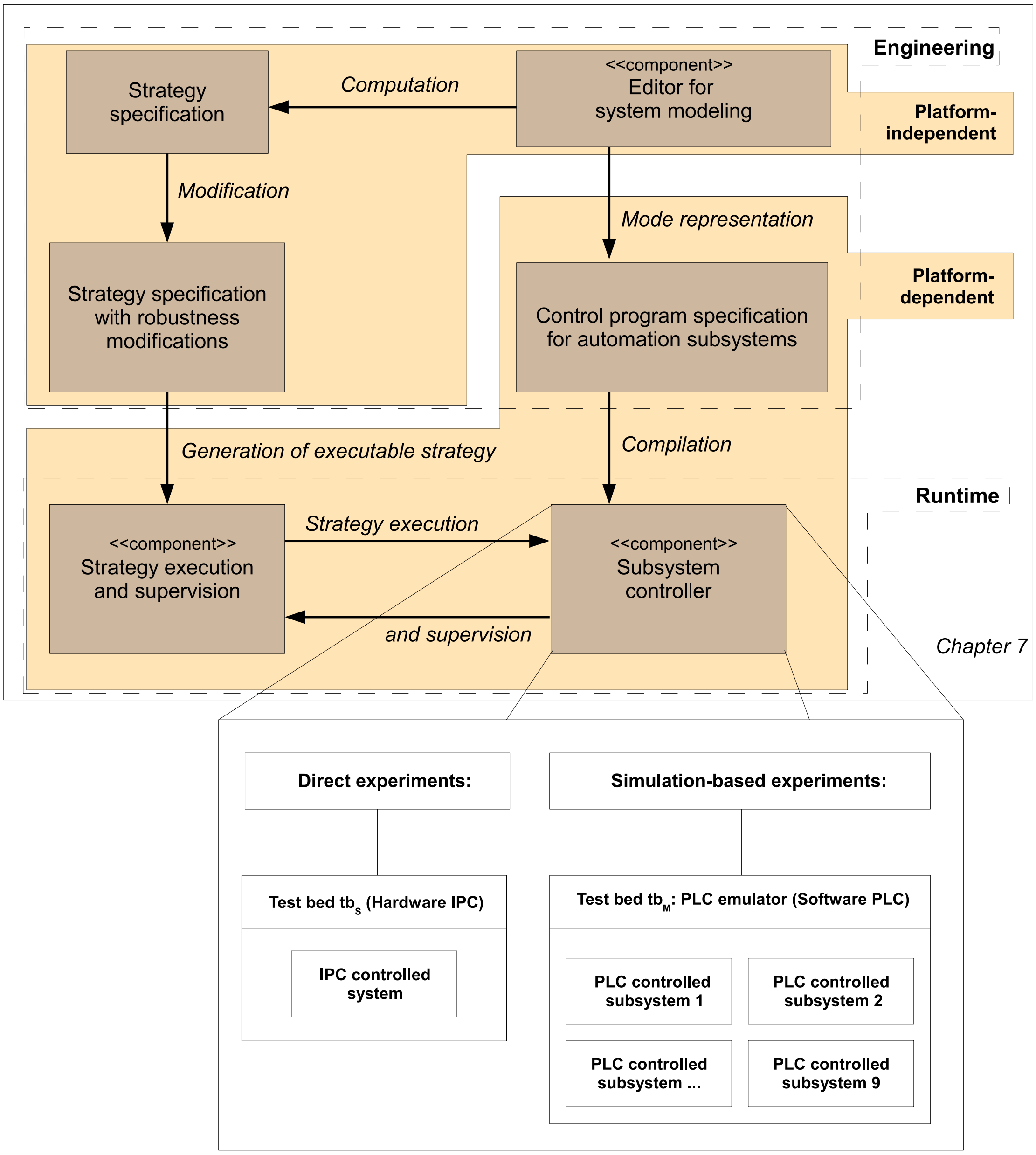


Figure 8.1: Elements of the test environment

8.3.1 Test Bed tb_S for direct experiments

Test Bed tb_S[1] serves as proof of technical feasibility of the approach presented in Part II (Fig. 8.2). The test bed has a conveyor belt run by an electric motor for transferring products supplied by several sensors. The process is controlled by a control unit CU. Measurements of electric units are conducted using the measurement unit. The test bed consists of four different elements: the measurement unit, the control unit CU using a peripheral device I/O, the process-related components, and the strategy-related functions. The components (*P1.1*, *P1.2*, *P2.1*, *P2.2*, *P3.1*, *P3.2*) of the *physical production process* are controlled by the control unit CU while sensors *P2.1*, *P2.2*, *P3.1*, *P3.2* are interfaced by the *peripheral device I/O* (Fig. 8.3). Motor controller *P1.1* and the *CU* have a direct communication link established. The electric motor *P1.2* is controlled by *P1.1*.

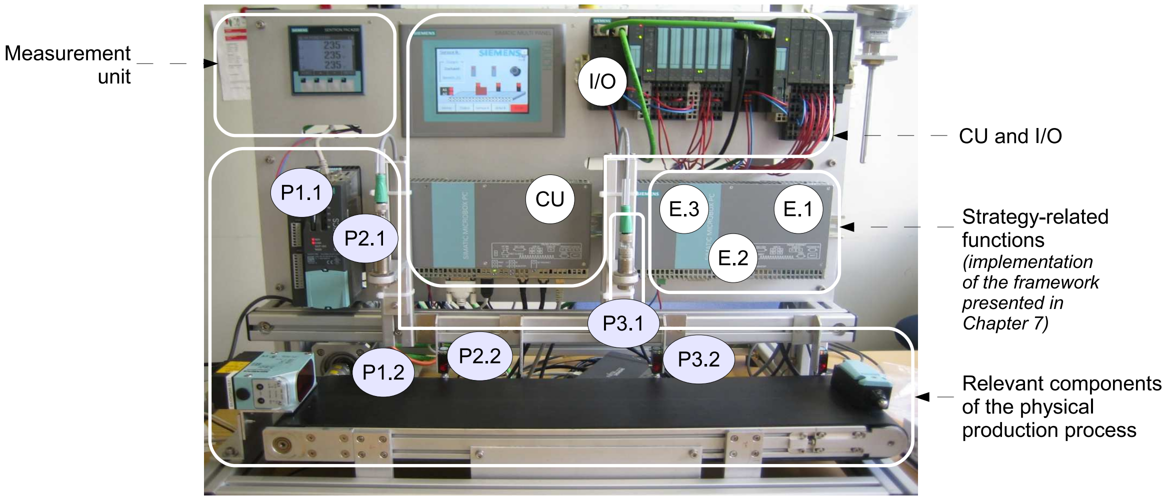


Figure 8.2: Test Bed tb_S and its components

The *strategy-related functions* for realizing strategies within unproductive phases proposed in Part II are implemented in components *E.3*, *E.2*, *E.1*. The *strategy execution and supervision E.3* executes and monitors the computed strategies for components *I/O*, *P2.1*, *P2.2*, *P3.1*, *P3.2* as well as *P1.1*, *P1.2*. Optimal strategies are provided by the *strategy specification E.2* that computes strategies based on user parametrization and the *automaton-based system model E.1*.

The system of Test Bed tb_S and its components is described using an automaton-based system model which can be found in Appendix A.1. The meaning of modes is explained in Appendix A.2.

[1]Test bed of the Siemens AG, Corporate Technology, Munich, Germany

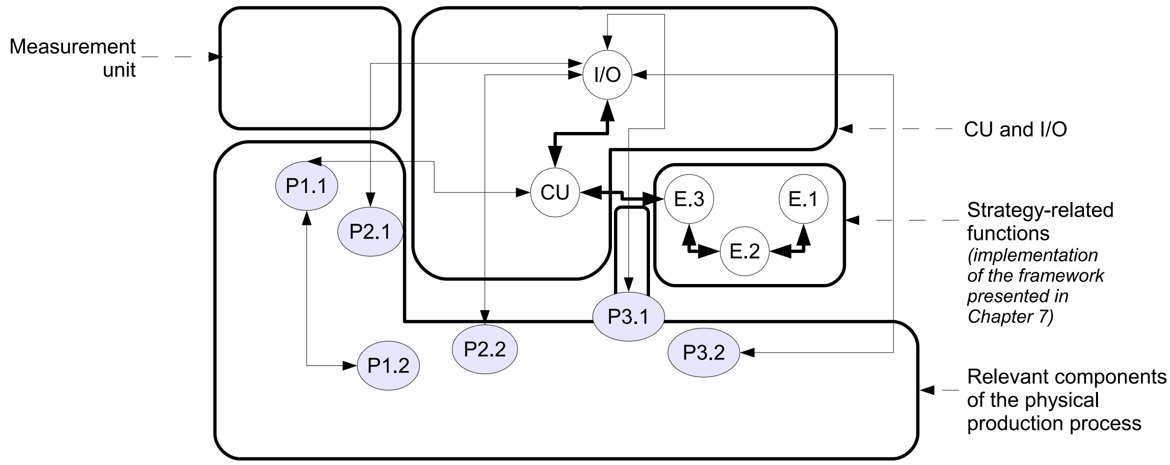


Figure 8.3: Component interaction

8.3.2 Test Bed $\mathbf{tb}_M$ for simulation-based experiments

In order to assess the approach of this thesis in a more extended setting than Test Bed tb_S, a model of the test bed tb_M[2] is used as evaluation basis (Fig. 8.4). The FMS (Section 3.1) consists of nine subsystems which are strongly coupled in terms of process-related dependencies. The models in form of networked automation subsystems can be found in Appendix B.1.2. Two different discrete processes are realized in the test bed: a *recycling* process (process *r*) and a *filling* process (process *f*) of bottles. The commissioning (Subsystem sub_6) serves as storage for empty bottles (*f*) and filled bottles (*r*). The uncapping and the discharging stations (Subsystems sub_3 and sub_4) are used to realize the recycling process *r*. In the scope of the filling process *f*, bottles are successively filled (Subsystem sub_1), picked (Subsystem sub_5), checked (Subsystems sub_7 and sub_8), and capped (Subsystem sub_2). The transportation subsystem (Subsystem sub_9) serves as material flow unit to transport bottles of both processes on a conveyor belt.

The FMS is equipped with a MES (Subsection 3.1.3). Each subsystem has its own CU which is communicating with other CUs in oder to realize the overall automation task. In the test bed, the automation process and its control infrastructure is distributed. The nine subsystems are coupled by subsystem dependencies (*sd*). A reason for modularization of the test bed can be seen in the ease of expandability of the system and fault localization.

To assess the performance of the approach, the control implemented by the CUs and the behavior of each subsystem is mimicked by a simulation. The hardware of the CUs is replaced by the software component *PLCSim*, version 5.4, which is a commercially available PLC emulator for Siemens S7-300 and S7-400 controllers [PLCSim, 2012]. *PLCSIM* allows the simulation of (multiple) controllers in order to conduct functional tests on the control program. This emu-

[2] Test bed of the Siemens AG, Industry Automation, Nuremberg, Germany

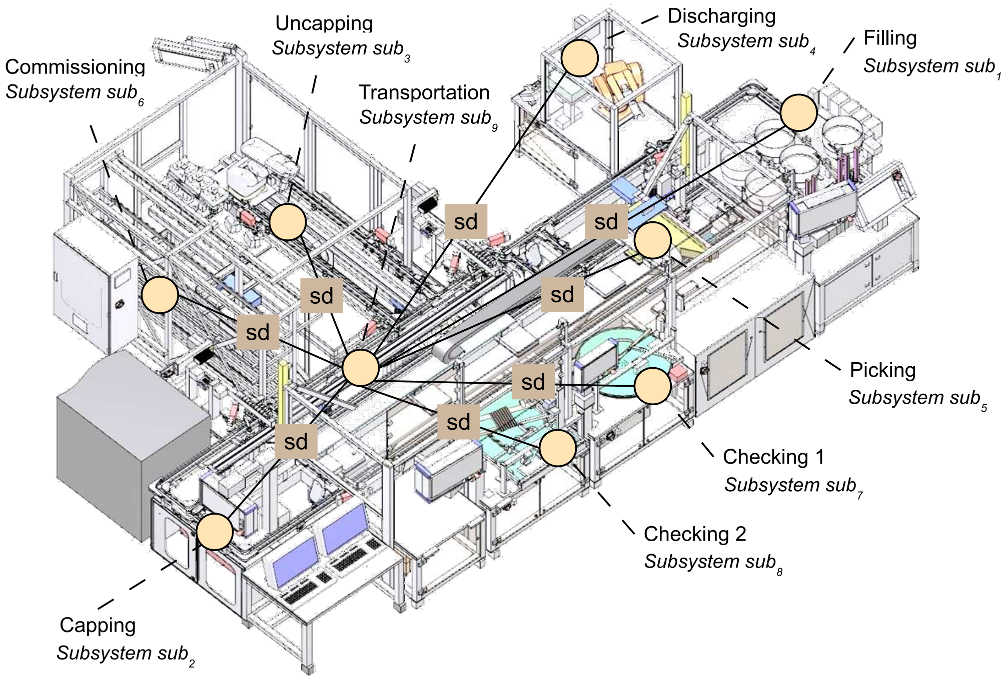


Figure 8.4: Test Bed tb_M with nine subsystems, subsystem dependencies are denoted by *sd*

lator comes with a C# .NET interface. The *PLCSIM* library provides a COM[3] object to directly communicate with emulated PLCs. This interface is used to integrate the PLC emulator in the framework presented in Chapter 7 resulting in the test environment illustrated in Figure 8.1. After having introduced the evaluation objectives, evaluation methods, and the test beds for evaluation, the approach of Part II can be evaluated in Chapter 9.

[3]Component Object Model

Chapter 9

Evaluation of the approach

In this chapter, the proposed approach of Part II is evaluated regarding its capability to contribute to the research objectives introduced in Section 1.3. On the one hand, Section 9.1 deals with evaluating the computational performance of identifying optimal strategies (related to Contribution 3). On the other hand, the test environment of Section 8.3 in combination with Test Bed tb_S and Test Bed tb_M is used for evaluating the proposed approach with regard to the research objectives given in Section 9.2 (related to Contribution 4), Section 9.3 (related to Contribution 5), and Section 9.4 (related to Contribution 6).

9.1 Identification of optimal strategies

This section analyzes influencing parameters on the identification of optimal strategies. Possible influencing parameters are the system scale and the system structure represented by the model (Subsection 9.1.1). These parameters are varied for measuring the impact on computational runtime and memory consumption (Subsection 9.1.2). A summary is given in Subsection 9.1.3.

9.1.1 Determination of the system scale and system structure

The *system scale* and the *system structure* are the parameters which can have an effect on the computational performance of the procedure proposed in this thesis (Chapter 6).

System scale

System scale is determined by three characteristics:

- the number of modes per subsystem: *mod*
- the number of transitions per subsystem: *tra*
- the number of subsystems: $\mid Sub \mid$

System structure

The system structure arises from subsystem dependencies. Subsystem dependencies sd_{ik} are links and relationships between subsystems (Def. 24, Subsection 4.1.1) materialized as guards on shared variables sv_i. In this section, it is assumed that at most one subsystem dependency sd_{ik} can exist between one pair of subsystems sub_i and sub_k, k = 1, ..., n. The system structure is determined by the number and distribution of subsystem dependencies and is classified based on three parameters:

- the dependency density of the system: *dep*
- the average number of subsystem dependencies per subsystem: *deg*
- the global clustering coefficient of the system with regard to subsystem dependencies: *clu*

Dependency density

The dependency density (Def. 43) is a global measure for the distribution of subsystem dependencies sd_{ik}.

Definition 43 (Dependency density)

The dependency density dep is the ratio of the number of existing subsystem dependencies and the number of possible subsystem dependencies: $dep = \frac{sd_{ref}}{\frac{1}{2}\cdot|Sub|\cdot(|Sub|-1)}$ *with* $\mathrm{sd}_{ref} = \sum_{i=1}^{|\mathrm{Sub}|}\sum_{k=1}^{|\mathrm{Sub}|} \mathrm{sd}_{ik}$ *$(i \neq k)$ as the number of existing subsystem dependencies between pairs of subsystems.*

Example 16

Figure 9.1 shows seven subsystems ($\mid Sub \mid = 7$): sub_1 to sub_7. The number of existing subsystem dependencies (represented as edges between subsystems) is $sd_{ref} = 6$. This results in a dependency density of $dep = \frac{6}{\frac{1}{2}\cdot 7\cdot(7-1)} \sim 29\%$.

Average number of subsystem dependencies

The average number of subsystem dependencies (Def. 44) is a measure for the relatedness of a subsystem to other neighboring subsystems.

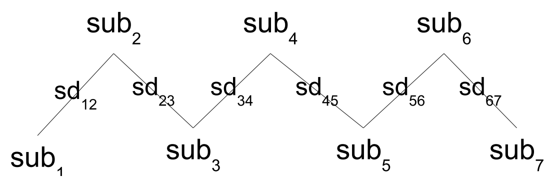


Figure 9.1: Line-structured Subsystems sub_1 to sub_7

Definition 44 (Average number of subsystem dependencies)
The parameter deg gives the average number of dependencies of a Subsystem sub_i to other Subsystems sub_k: $deg = \frac{1}{|Sub|} \cdot \sum_{i=1}^{|Sub|} \sum_{k=1}^{|Sub|} sd_{ik} = \frac{sd_{ref}}{|Sub|}, i \neq k$
with sd_{ref} as the number of existing subsystem dependencies of a Subsystem sub_i to a Subsystem sub_k.

Example 17
In Figure 9.1, there exist sd_{ref} = 6 subsystem dependencies and $\mid Sub \mid$ = 7 subsystems. This results in $deg = \frac{6}{7} \sim 0{,}86$.

Clustering coefficient
The clustering coefficient for undirected graphs (Def. 45) gives an indication of how strong the subsystems of the system tend to build groups [Watts, 1998].

Definition 45 (Clustering coefficient)
The local clustering coefficient of an automation subsystem is given by the ratio of actual subsystem dependencies sd_{jk} between the neighboring subsystems Sub_i^{neigh} of a Subsystem sub_i and the number of subsystem dependencies that could possibly exist between the neighboring subsystems Sub_i^{neigh} of Subsystem sub_i (Equ. 9.1). The global clustering coefficient clu (Equ. 9.2) is the arithmetic average of the sum of local clustering coefficients (Equ. 9.1).

$$c_i = \frac{\mathrm{sd}_{jk}}{\frac{1}{2} \cdot \mid \mathrm{Sub}_i^{\mathrm{neigh}} \mid \cdot (\mid \mathrm{Sub}_i^{\mathrm{neigh}} \mid -1)} \tag{9.1}$$

with sd_{jk} as the actual subsystem dependencies between neighboring subsystems of Subsystem sub_i, and $\mid \mathrm{Sub}_i^{\mathrm{neigh}} \mid$ as the number of neighboring subsystems of Subsystem sub_i. The local clustering coefficient is only defined for $\mid \mathrm{Sub}_i^{\mathrm{neigh}} \mid > 1$. If $\mid \mathrm{Sub}_i^{\mathrm{neigh}} \mid \leq 1$, then $c_i = 0$.

$$\mathrm{clu} = \frac{1}{\mid \mathrm{Sub} \mid} \cdot \sum_{i=1}^{|\mathrm{Sub}|} c_i \tag{9.2}$$

The local clustering coefficient gives the probability for the neighbors $\mathrm{Sub}_i^{\mathrm{neigh}}$ of a subsystem sub_i being linked by subsystem dependencies building a clique. Equation 9.2 shows the overall clustering of the system whereas Equation 9.1 provides a hint of the probability of a single subsystem being embedded in a cluster.

Example 18

Figure 9.2 (b) exemplifies the term neighboring subsystems Sub_i^{neigh} *of a Subsystem* sub_i*. Subsystem* sub_1 *and Subsystem* sub_3 *are the neighboring subsystems* Sub_2^{neigh} *of Subsystem* sub_2*. The neighboring subsystems* Sub_2^{neigh} *have one single subsystem dependency* sd_{13}*.*
In Figure 9.2 (a), five Subsystems sub_1 *to* sub_5 *are linked by subsystem dependencies* sd_{ik}*. First, the local clustering coefficients* c_i *are calculated:*

- $c_1 = \frac{2}{\frac{1}{2}\cdot 3\cdot(3-1)} \sim 0{,}67$
- $c_2 = \frac{1}{\frac{1}{2}\cdot 2\cdot(2-1)} = 1$
- $c_3 = \frac{2}{\frac{1}{2}\cdot 4\cdot(4-1)} \sim 0{,}33$
- $c_4 = \frac{1}{\frac{1}{2}\cdot 2\cdot(2-1)} = 1$
- $c_5 = 0$*, since* $\mid Sub_i^{neigh} \mid = 1$

The arithmetic average of c_i *is the global clustering coefficient* $clu = \frac{1}{5} \cdot \sum_{i=1}^{5} c_i = \frac{1}{5} \cdot (0,67 + 1 + 0,33 + 1 + 0) = 60\%$*.*

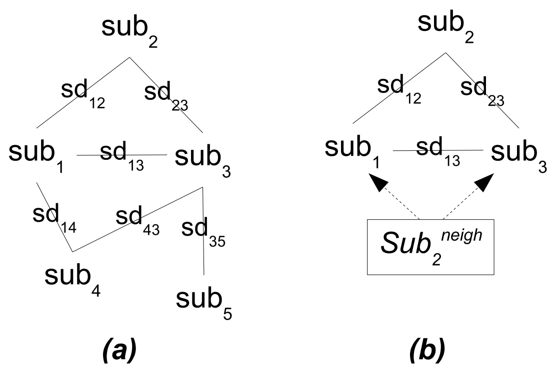


Figure 9.2: (a) Highly-meshed Subsystems sub_1 to sub_5, (b) Neighboring Subsystems sub_1 and sub_3 of Subsystem sub_2

The parameters determining the system scale and the system structure are varied for measuring their impact on computational runtime identifying optimal strategies, next.

9.1.2 Parameter variation identifying optimal strategies

Parameters defined in Subsection 9.1.1 are varied to evaluate their impact on performance regarding the identification of optimal strategies using different optimization models and methods. The variation of influencing parameters is conducted based on scenarios.
The evaluation towards computational performance regarding the identification of optimal

strategies is classified according to the addressed objectives. There are two objectives (denoted by obj_{time} and obj_{energy}, Subsection 5.2.2) that can be addressed by the approach of this thesis (Fig. 9.3, Objectives).

- obj_{time} addresses the task of reaching a target mode starting in an initial mode within each subsystem in minimum time. This corresponds to the problem of weighted minimum completion time with weight $w_i = 1$ (Subsection 3.4.1).
- obj_{energy} addresses the task of reaching a target mode starting in an initial mode within each subsystem requiring a minimum energy demand for a given pause interval (Subsection 5.2.3).

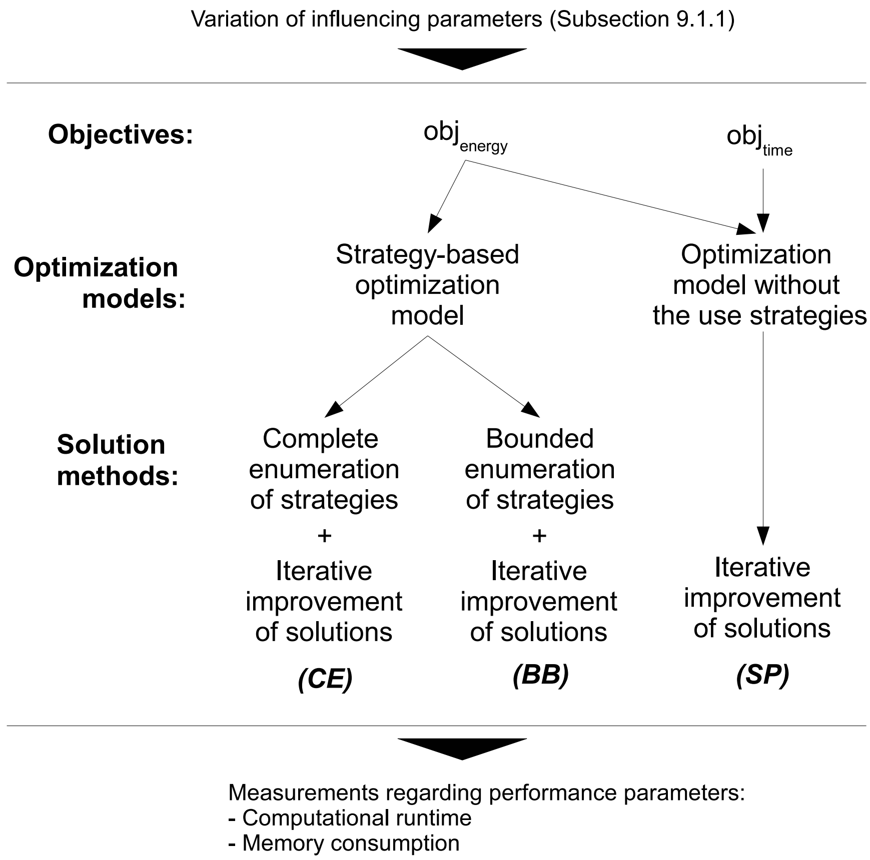


Figure 9.3: Performance of identifying optimal strategies: influencing parameters, objectives, optimization models, and solution methods

Each objective is materialized by an optimization model (Fig. 9.3, *Optimization models*). It is distinguished between two types:

- Strategy-based optimization model: A formulation according to Chapter 5 is applied for identifying the optimal strategy

- Optimization model without the use of strategies: This is the classical approach without using the structural knowledge (modularity) of the automation system materialized by strategies (Subsection 5.1.3). The information distributed among strategies is represented in a compact way in a single optimization model.

The objective obj_{time} cannot be efficiently supported by a strategy-based optimization model since no valuation using energy demand (compare obj_{energy}) is applicable. The objective obj_{time} has to use an optimization model without the use of strategies.
Different optimization models enable the application of specific methods to find an optimal solution or strategy (Fig. 9.3, Solution methods):

- Complete enumeration of strategies (Complete enumeration of the set of strategies (CE)): the complete enumeration of strategies according to Section 6.3. This is realized by renouncement of the investigation stop in Algorithm 3. Each strategy is iteratively improved to identify the minimum energy demand.

- Bounded enumeration of strategies (Bounded investigation of the set of strategies (BB)): the bounded enumeration of strategies according to the procedure presented in Section 6.3. This is realized by the use of the investigation stop in Algorithm 3. Each strategy is iteratively improved to identify the minimum energy demand.

- Complete enumeration of solutions (Complete enumeration of solutions without the use of strategies (SP)): Identifying the optimal solution to the single optimization model, an iterative improvement of found solutions is applied.

Regarding objective obj_{energy}, the chosen evaluation approach enables the comparison of the approach proposed in Chapter 6 in the following way:

- The complete enumeration (CE) using a strategy-based optimization model guarantees to identify the energy-optimal strategy. Is the bounded enumeration approach (BB) capable of identifying the same energy-optimal strategy? If so, what is the performance gain using a bounded investigation of the set of strategies?

- Comparing with a customary approach (SP), what is the performance gain using the bounded investigation (BB) of the set of related strategies?

Identification of energy-minimal strategies with 4 subsystems

Addressing objective obj_{energy}, the initial and target mode configuration of the subsystems is $<m_0, ..., m_0>$. The internals of subsystems is illustrated in Figure C.1 and Figure C.2. The pause interval is provided with 300 seconds. The elementary time interval t_{i+1} - t_i is one second, so that the 300 seconds pause interval is divided into 300 uniform elementary time intervals. Four

Scen.	Method	System scale			System structure			Objective	
		mod	tra	|Sub|	dep	clu	deg	obj_{time}	obj_{energy}
Eff.1.1	SP , CE , BB	10	12	4	67%	75%	1,0		x
Eff.1.2	SP , CE , BB	10	12	4	100%	100%	1,5		x
Eff.1.3	SP , CE , BB	19	24	4	50%	0%	0,75		x
Eff.1.4	SP , CE , BB	19	24	4	83%	83%	1,25		x

Table 9.1: *Eff.1.x*: Parameter variations, 4 subsystems

different scenarios with varying system scale and system structure are chosen for a system with four subsystems (Tab. 9.1).

The corresponding performance parameters are presented in Table 9.2. The search for energy-optimal solutions using SP is aborted when reaching the computational runtime of CE. Using methods CE and BB, the computational runtime *run* includes the computation for infeasible strategies (Section 6.3).

In non of the scenarios, SP is capable to find adequate good solutions comparable to CE and BB indicated by the value in column *Best strategy* within the given time. The strategy-based methods CE and BB find the optimal strategy (same value in column *Best strategy*), however the method BB requires less computational time.

The difference in computational runtime between CE and BB can be shown graphically in Figure 9.4 and Figure 9.5) referring to Scenario *Eff.1.4*. The BB procedure terminates after investigating 16 strategies (2,2% of L) identifying the strategy with energy demand of 342 [kJ] ($e(l_1) \leq e(l_{16,unrel})$). The complete enumeration of strategies (CE) investigates all 728 strategies comming to the result (energy demand of the best strategy l_1 is $e(l_1)$ = 342 [kJ]).

Scen.	**Method**	**Best strategy**	**Performance**		**Strategies**		
		[kJ]	mem [MB]	run [s]	L investigated	L	$\sim$ % of L investigated
Eff.1.1	SP	1.023	75	2.655#	–	–	–
	CE	183	74	2.655*	68	68	100
	BB	183	67	7*	2	68	3
Eff.1.2	SP	997	76	2.563#	–	–	–
	CE	156	67	2.563*	67	67	100
	BB	156	67	8*	2	67	3
Eff.1.3	SP	1.055	98	> 55.000#	–	–	–
	CE	293	143	> 55.000*	1.087	1.087	100
	BB	293	74	55*	2	1.087	0,2
Eff.1.4	SP	935	100	> 41.000#	–	–	–
	CE	342	134	> 41.000*	728	728	100
	BB	342	74	1.134*	16	728	2,2

Table 9.2: *Eff.1.x*: Performance of SP, CE and BB investigation during identification of energy-optimal strategies, *) Runtime includes computational time for investigating infeasible related strategies, #) SP is aborted

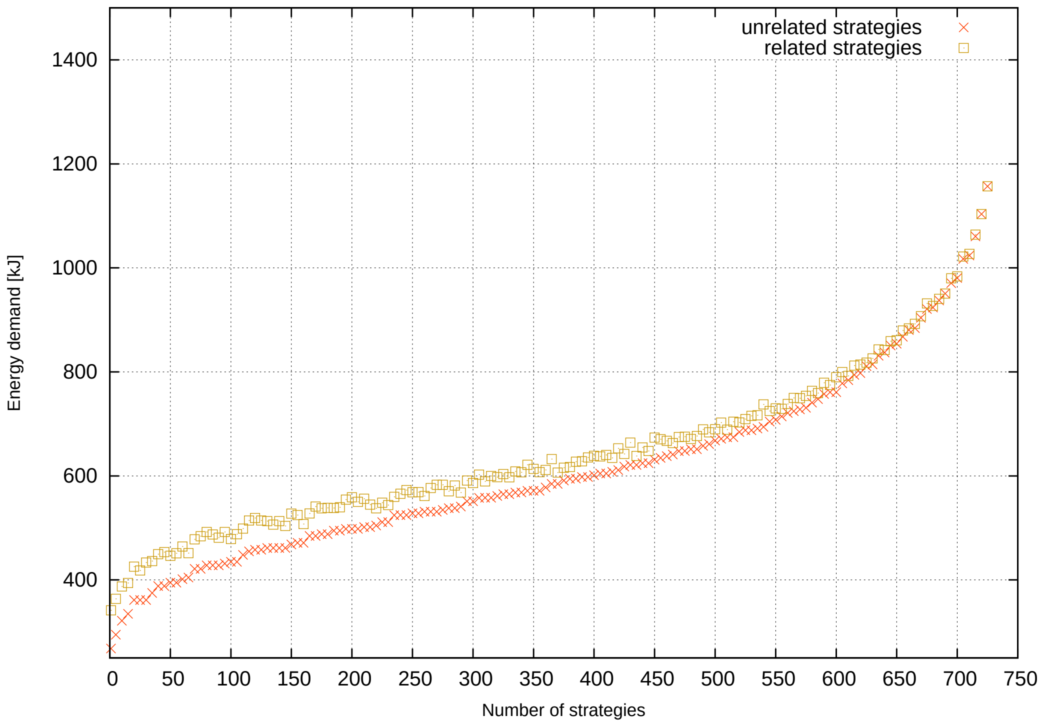


Figure 9.4: *Eff.1.4*: Complete enumeration of related strategies

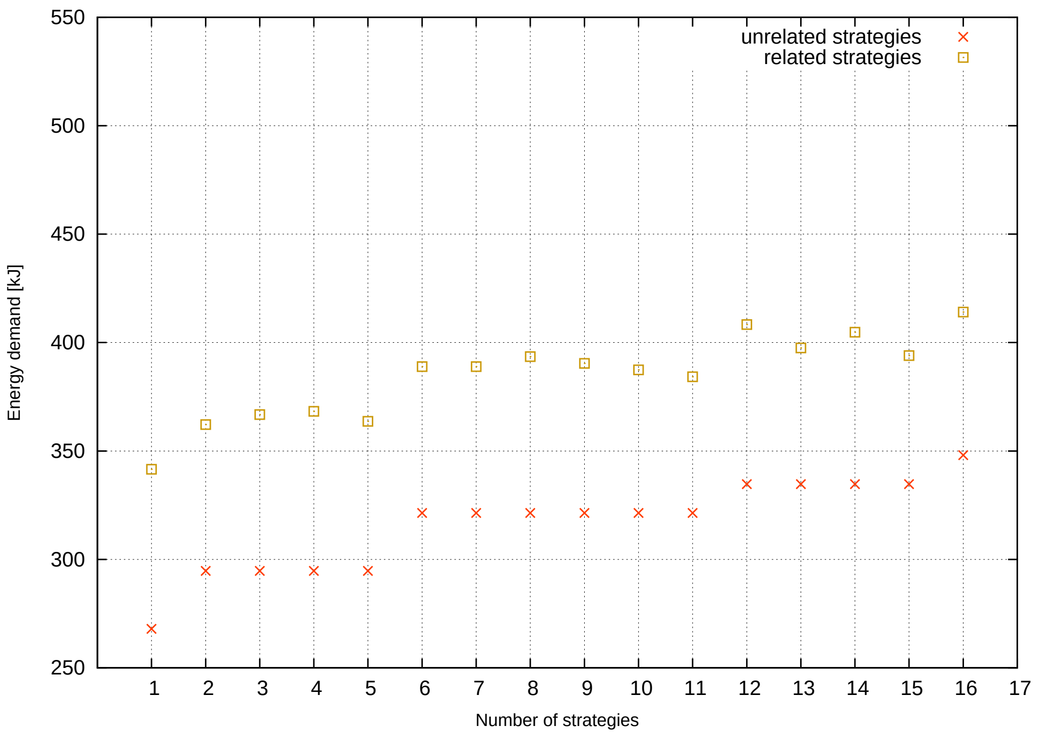


Figure 9.5: *Eff.1.4*: BB investigation of related strategies

Identification of energy-minimal strategies with 5 subsystems

In Scenarios *Eff.2.x*, the number of subsystems is augmented to five subsystems (Tab. 9.3). Moreover, in Scenario *Eff.2.2*, the pause interval has been set to 100 seconds in order to reduce the number of possible strategies. With a 300 seconds pause interval, 2.591 strategies would have to be investigated in Scenario *Eff.2.2* (Tab. 9.4).

Scen.	**Method**	**System scale**			**System structure**			**Objective**	
		mod	tra	|Sub|	dep	clu	deg	obj_{time}	obj_{energy}
Eff.2.1	SP , CE , BB	10	12	5	40%	0%	0,8		x
Eff.2.2	SP , CE , BB	19	24	5	70%	76%	1,4		x

Table 9.3: *Eff.2.x*: Parameter variations, 5 subsystems

Table 9.4 presents the influence of the parameters variation on computational performance.

Scen.	**Method**	**Best strategy**	**Performance**		**Strategies**		
		[kJ]	mem [MB]	run [s]	L investigated	L	$\sim$ % of L investigated
Eff.2.1	SP	1.260	77	15.252#	–	–	–
	CE	205	118	15.252*	560	560	100
	BB	205	69	92*	2	560	0,3
Eff.2.2	SP	396	90	29.320#	–	–	–
	CE	373	497	29.320*	1.180	1.180	100
	BB	373	363	7.671*	559	1.180	47

Table 9.4: *Eff.2.x*: Performance of SP, CE and BB investigation during identification of energy-optimal strategies, *) Runtime includes computational time for investigating infeasible related strategies, #) SP is aborted

Comparing Scenarios *Eff.2.1* and *Eff.2.2* reveals that the more complex the system structure gets, the harder (column *run* in Tab. 9.4) is the identification of the optimal strategy using method SP, CE, and BB. However, BB outperforms SP and CE in both scenarios.

Identification of energy-minimal strategies with 7 subsystems

In Scenarios *Eff.3.x*, the performance of SP and BB approaches are evaluated for seven interrelated subsystems (Tab. 9.5). The identification of the energy-optimal strategy by the complete enumeration approach CE is not feasible in reasonable time, so that only 10% (Scenario *Eff.3.1*) respectively 5% (Scenario *Eff.3.2*) of L is enumerated and denoted as PE (Tab. 9.6).

Scen.	**Method**	**System scale**			**System structure**			**Objective**	
		mod	tra	|Sub|	dep	clu	deg	obj_{time}	obj_{energy}
Eff.3.1	SP , PE , BB	10	12	7	29%	0%	0,9		x
Eff.3.2	SP , PE , BB	10	12	7	48%	48%	1,4		x

Table 9.5: *Eff.3.x*: Parameter variations, 7 subsystems

Neither a CE approach (investigating 100% of L) nor a SP approach can provide the energy-optimal strategy within reasonable time shown in Table 9.6.

Scen.	**Method**	**Best strategy**	**Performance**		**Strategies**		
		[kJ]	mem [MB]	run [s]	L investigated	L	$\sim$ % of L investigated
Eff.3.1	SP	1.946	86	6.982#	–	–	–
	PE	303	120	6.982*	157	1.569	10
	BB	303	73	53*	2	1.569	0,1
Eff.3.2	SP	2.074	86	2.957#	–	–	–
	PE	250	128	2.957*	86	1.719	5
	BB	250	73	51*	2	1.719	0,1

Table 9.6: *Eff.3.x*: Performance of SP, CE and BB investigation during identification of energy-optimal strategies, *) Runtime includes computational time for investigating infeasible related strategies, #) SP is aborted

The SP method and CE method meet their limits identifying the energy-optimal strategy. Because of the bounded investigation of the set of strategies, BB is capable to identify the energy-optimal strategy for Scenario *Eff.3.1* and Scenario *Eff.3.2*.

Identification of time-minimal strategies

Addressing the objective obj_{time}, subsystems are initially in mode <m_0, ..., m_0> (Fig. C.1). The target modes are given with <m_9, ..., m_9>. The objective is to compute the strategy in which each subsystem requires least time from the initial mode to the target mode. Scenarios with the same system scale and differing system structure are chosen (Tab. 9.7).

Scen.	Method	System scale			System structure			Objective	
		mod	tra	|Sub|	dep	clu	deg	obj_{time}	obj_{energy}
Eff.4.1	SP	10	12	7	29%	0%	0,9	x	
Eff.4.2	SP	10	12	7	48%	48%	1,4	x	
Eff.4.3	SP	10	12	7	52%	82%	1,6	x	

Table 9.7: *Eff.4.x*: Parameter variations

The performance results are presented in Table 9.8.

Scen.	Method	Best strategy	Performance	
			mem [MB]	run [s]
Eff.4.1	SP	found	64	1,8
Eff.4.2	SP	found	63	5,3
Eff.4.3	SP	found	63	53

Table 9.8: *Eff.4.x*: Performance of SP investigation identifying minimal-time strategies

The more complex the system structure (number and distribution of subsystem dependencies) is, the harder it is to identify the time-optimal strategy.

9.1.3 Summary

In this section, the computational performance to identify an optimal strategy of three different methods SP (iterative improvement of found solutions in a single optimization problem), CE (complete enumeration of strategies), and BB (bounded enumeration of strategies) has been evaluated towards influencing parameters (system scale and system structure). Big differences in computational runtime are identified between SP on the one hand and CE and BB on the other hand. Especially the incorporation of structural knowledge (modularity) into the computation of energy-optimal strategies (obj_{energy}) enables the significant reduction of computational runtime (BB approach outperforms a CE approach). Referring to Contribution 3, the following result can be stated (Conclusion 1).

Conclusion 1 (Efficient identification of energy-optimal strategies)
The selective procedure (using structural knowledge) for investigating the set of related strategies L (Chapter 6) provides a more efficient technique to identify the energy-optimal related strategy compared with an approach without the use of strategies. The gain in efficiency is explainable by the tractable technique applied by the selective procedure. The comparison of the selective procedure with a complete enumeration of strategies shows that the selective procedure is able to identify the energy-optimal strategy.

9.2 Specification of feasible strategies

In this section, strategy specifications are checked towards correctness and the absence of errors preventing the execution of strategies using the framework of Chapter 7. Correctness includes the execution of a strategy complying with its specification. This refers to the question if the computed sequence of modes of a strategy can be executed in the right temporal and successive order in the system.

Strategy feasibility is ensured by proposing two procedures. Using the first procedure, computed strategies are executed directly in the system (using Test Bed tb_S, Subsection 9.2.1) which gives direct feedback towards feasibility. The second procedure is applicable, if the physical manufacturing equipment is not present, for instance during engineering of the automation system (using Test Bed tb_M, Subsection 9.2.2). In this case, a simulation-based approach can be used introducing artificial model-to-system deviations in order to put the strategy execution under stress.

The relevant model-to-system deviation influencing the feasibility of strategies is related to the mode delay of modes (Def. 27, Subsection 4.2.2). The modeled mode delays may differ from the mode delay actually present in the system. The average deviation between modeled mode delays and actual mode delays of the system of a strategy for n subsystems is given by Equation 9.3.

$$\text{dev}_{\text{delay}}[\%] = \frac{1}{n} \sum_{i=1}^{n} \left[\frac{\text{pa}(\text{sub}_i)^{\text{planned}}}{\text{pa}(\text{sub}_i)^{\text{actual}}} - 1 \right] \cdot 100 \tag{9.3}$$

with $pa(sub_i)^{planned}$ as planned pause interval of a subsystem sub_i in the strategy, $pa(sub_i)^{actual}$ as pause interval of a subsystem sub_i in the actually executed strategy, and $\text{pa}(\text{sub}_i)^{\text{planned}} > 0$. In addition to the statement if a planned strategy is feasible, the parameter $\text{dev}_{\text{delay}}$ is given indicating the robust feasibility of a strategy.

9.2.1 Scenario-based feasibility analysis in Test Bed tb_S

Strategies derived from the system model (Fig. A.1) are executed in the Test Bed tb_S. Five strategies are examined with different pause parametrization in five scenarios (*Ver.tbS.1*: 10 [s], *Ver.tbS.2*: 30 [s], *Ver.tbS.3*: 60 [s], *Ver.tbS.4*: 300 [s], *Ver.tbS.5*: 600 [s]). Each strategy requires an initial mode β and a target mode β.

Table 9.9 presents the results of executed strategies in Test Bed tb_S. Column *Model/System* shows the planned pause interval (model) and the pause interval of the executed strategy (system). The actual pause interval is always bigger than the planned one since communication between strategy execution and supervision (Subsection 7.5) and the system extends planned pause intervals. This fact is represented in the sum of the model-to-system deviations regarding the mode delays $\text{dev}_{\text{delay}}$. The negative value of $\text{dev}_{\text{delay}}$ expresses the underestimation of the actual mode delays. Nevertheless, each strategy that is planned (*Ver.tbS.1* through *Ver.tbS.5*) is correctly executed in Test Bed tb_S (● := *planned strategy is feasible*, ○ := *planned strategy is not feasible*).

Scen.	**Model/ System**	**Input**				**Output**	
		Pause planned/ actual [s]	Initial	Final	Average mode delay deviation dev_{delay} [$\sim$ %]	Energy input [J] Model/ System	Planned strategy feasible? (●/○)
Ver.tbS.1	Model	10	β	β	-10,7	840	●
	System	11,2	β	β		939	
Ver.tbS.2	Model	30	β	β	-10,2	2.455	●
	System	33,4	β	β		2.723	
Ver.tbS.3	Model	60	β	β	-6,4 ± 0,3	4.880	●
	System#	64,1 ± 0,2	β	β		4.868 ± 51	
Ver.tbS.4	Model	300	β	β	-1,3 ± 0,1	20.860	●
	System#	304,0 ± 0,1	β	β		20.530 ± 277	
Ver.tbS.5	Model	600	β	β	-0,8 ± 0,1	41.260	●
	System#	604,5 ± 0,7	β	β		40.133 ± 106	

Table 9.9: Test Bed tb_S: Scenario overview for feasibility analysis, #average, n = 4

Compared to Test Bed tb_S which consists of a single subsystem, Test Bed tb_M comprises nine interacting subsystems.

9.2.2 Scenario-based feasibility analysis in Test Bed tb_M

Using simulation-based testing of Test Bed tb_M (Subsection 8.3.2), the robustness of the strategy execution is assessed by inducing dysfunctions in the form of specific degrees of dev_{delay} which represent possible model-to-system deviations in terms of planned and actually executed mode delays. The output (sequential order of modes) produced by the executed strategy is used to determine if the computation is correct and feasible. Executed strategies are traced and compared to the planned strategy.

Table 9.10 presents eight scenarios with different mode delay deviations between model and system (*in* %). The requirement of each scenario is to return to the initial mode tuple $\zeta^{(n)} := \alpha^{(9)} = <\alpha, \alpha, \alpha, \alpha, \alpha, \alpha, \alpha, \alpha, \alpha>$ after a 600 seconds interval. During this time period, the automation system is operated in modes representing unproductiveness using strategies. Planned and actual energy input [kJ] is shown as well as the feasibility of the planned strategy (● := *planned* *strategy is feasible, ○ := planned strategy is not feasible*).

Scen.	**Model/ System**	**Input**			**Output**	
		Initial	Final	Average mode delay deviation $\text{dev}_{\text{delay}}$ [∼ %]	Energy input [kJ] Model/ System	Planned strategy feasible? (●/○)
Ver.tbM.0	Model System	$\alpha^{(9)}$ $\alpha^{(9)}$	$\alpha^{(9)}$ $\alpha^{(9)}$	-0,2	428,7 429,8	●
Ver.tbM.1	Model System	$\alpha^{(9)}$ $\alpha^{(9)}$	$\alpha^{(9)}$ $\alpha^{(9)}$	+0,4	432,2 428,9	●
Ver.tbM.2	Model System	$\alpha^{(9)}$ $\alpha^{(9)}$	$\alpha^{(9)}$ $\alpha^{(9)}$	-2,1	440,0 479,4	●
Ver.tbM.3	Model System	$\alpha^{(9)}$ $\alpha^{(9)}$	$\alpha^{(9)}$ $\alpha^{(9)}$	-0,4	443,2 468,9	●
Ver.tbM.4	Model System	$\alpha^{(9)}$ $\alpha^{(9)}$	$\alpha^{(9)}$ $\alpha^{(9)}$	2,1	451,3 430,0	●
Ver.tbM.5	Model System	$\alpha^{(9)}$ $\alpha^{(9)}$	$\alpha^{(9)}$ $\alpha^{(9)}$	-3,4	425,4 441,3	●
Ver.tbM.6	Model System	$\alpha^{(9)}$ $\alpha^{(9)}$	$\alpha^{(9)}$ $\alpha^{(9)}$	-1,8	417,5 434,3	●
Ver.tbM.7	Model System	$\alpha^{(9)}$ $\alpha^{(9)}$	$\alpha^{(9)}$ $\alpha^{(9)}$	-3,6	407,2 440,0	●

Table 9.10: Test Bed tb_S: Scenario overview for feasibility analysis

Table 9.10 shows that each strategy is actually feasible in Test Bed tb_M. Giving two examples[1] of feasible strategies, the strategy of Scenario *Ver.tbM.0* and Scenario *Ver.tbM.4* are graphically illustrated in Figure 9.6 respectively Figure 9.7. Scenario *Ver.tbM.0* has a $\text{dev}_{\text{delay}}$ = -0,2 ± 0%. This reflects an ideal representation of the test bed by the model. The small $\text{dev}_{\text{delay}}$ = -0,2% is a model-to-system devition induced by the communication between strategy execution and supervision on the one hand and Test Bed tb_M on the other hand. This communication comprises the initiation of a switching action by the strategy execution and supervision and a response back by the test bed. Scenario *Ver.tbM.4* has a $\text{dev}_{\text{delay}}$ = -2,1% ± 1,5%. In both scenarios,

[1]The graphical illustration of *Ver.tbM.1*, *Ver.tbM.5*, *Ver.tbM.7* can be found in Appendix C.2.

the succession of modes of a strategy for the nine subsystems of Test Bed tb_M is presented over time. For each subsystem, the planned (*Subsx-plan*) and the actually executed (*Subsx-actl*) switching sequence is displayed using color codes. The colored bars indicate for each subsystem the time spent in a mode. Comparing the planned and executed strategy shows that the mode succession is correctly executed respecting temporal and structural constraints of Test Bed tb_M.

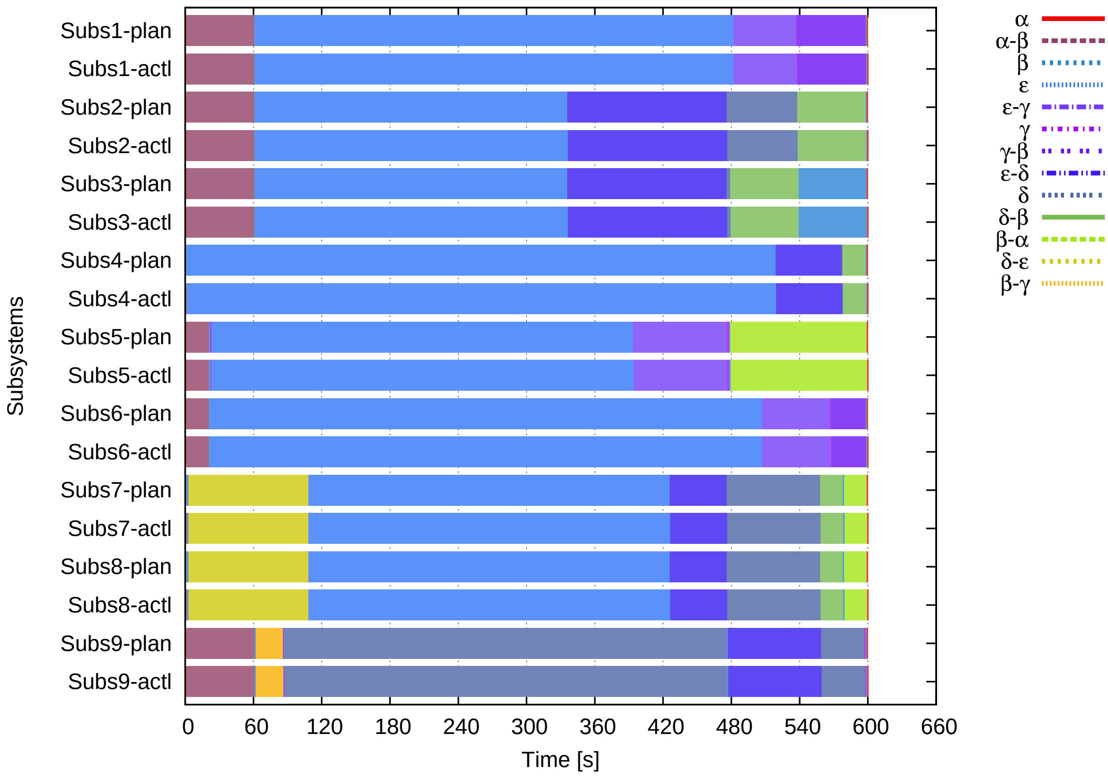


Figure 9.6: *Ver.tbM.0*: Planned and executed strategy (dev_{delay} = -0,2%)

9.2.3 Summary

In this section, strategies are checked towards feasibility in two automation systems. The Test Beds tb_S and tb_M are used to analyze strategy feasibility in different settings. In Test Bed tb_S, strategies are specified for different pause intervals while the physical test bed determines the model-to-system deviations dev_{delay} (Subsection 9.2.1). In Test Bed tb_M, model-to-system deviations are artificially introduced in order to vary dev_{delay} and produce different degrees of dev_{delay} (Subsection 9.2.2).

In all test scenarios (Ver.tbS.1 to Ver.tbS.5, Ver.tbM.0 to Ver.tbM.7), the planned strategies are feasible and robustly executed using the framework proposed in Chapter 7. The modeled structural constraints can be met during execution although model-to-system deviations in form of different degrees of dev_{delay} occur.

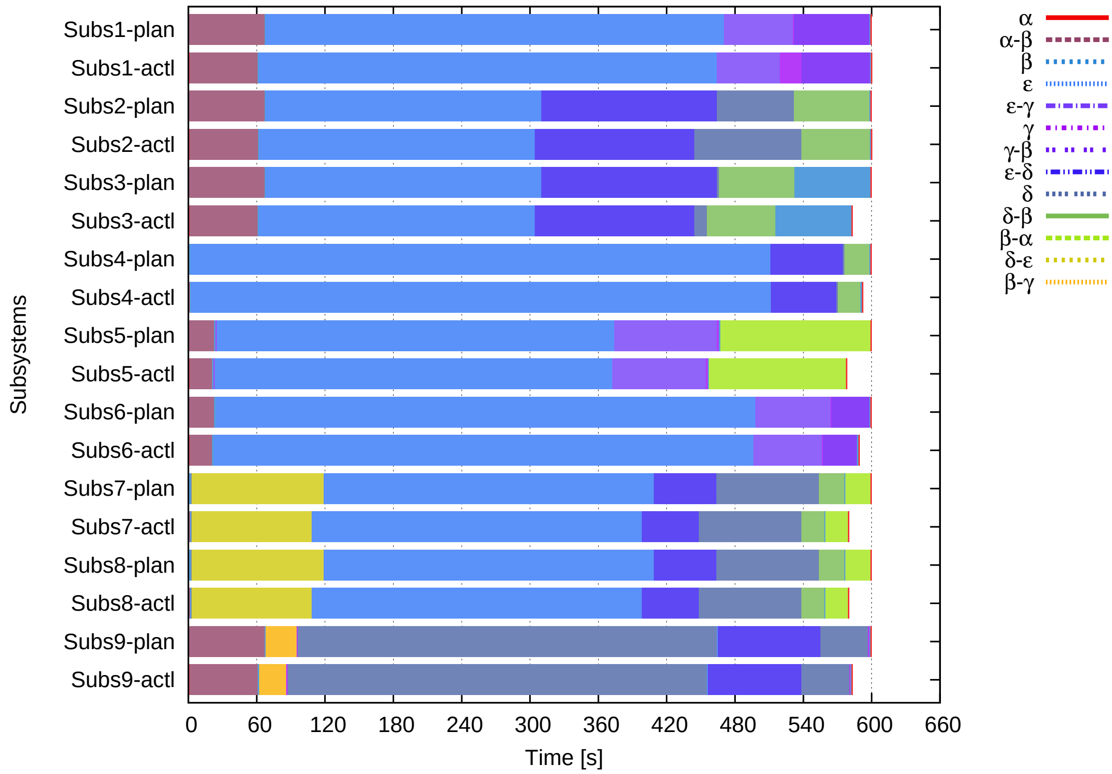


Figure 9.7: *Ver.tbM.4*: Planned and executed strategy (dev_{delay} = +2,1%)

Conclusion 2 (Feasibility of strategies)
Despite of model-to-system deviations dev_{delay}, each tested strategy is feasible in the system respecting subsystem dependencies using the proposed framework of Chapter 7.

Although the effects of model-to-system deviations do not affect the strategy feasibility, model-to-system deviations influence the accuracy of the prediction of the energy demand. On this account, model-to-system deviations need to be analyzed towards their effects on model validity which can be evaluated using strategies in Section 9.3.

9.3 Model validation using strategies

The feasibility of planned strategies has been shown in the preceding section. Besides feasibility, the validity of strategies needs to be ensured using validation tests. Model validation is the continuing process of determining the accuracy of a model regarding the representation of a real-world system [VDI-3633, 1996]. The focus of this section is the evaluation of the predictions regarding energy demand using strategies.
There are two types of model-to-system deviations that can occur and which influence the validity of energy demand predictions (Fig. 9.8).

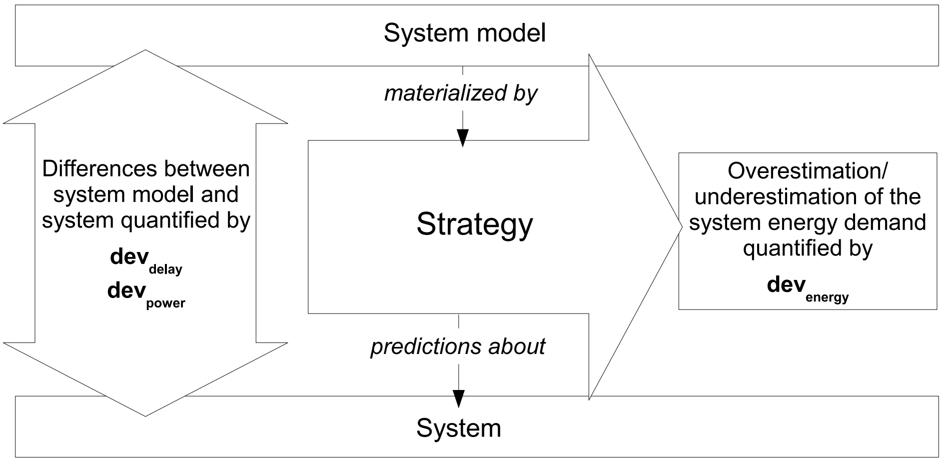


Figure 9.8: Influences $\mathrm{dev}_{\mathrm{delay}}$ and $\mathrm{dev}_{\mathrm{power}}$ on the quality of energy demand prediction $\mathrm{dev}_{\mathrm{energy}}$

The first type is mentioned in Section 9.2: the average deviation of mode delays dev_{delay} (Equ. 9.3). The second type is the average deviation of the input power of modes between the model and the system dev_{power} (Equ. 9.4).

$$\mathrm{dev}_{\mathrm{power}}[\%] = \frac{1}{n} \cdot \sum_{i=1}^{n} \left[\frac{\mathrm{pc}_i(\mathrm{m})_{\mathrm{model}}}{\mathrm{pc}_i(\mathrm{m})_{\mathrm{system}}} - 1 \right] \cdot 100 \tag{9.4}$$

with pc as the average input power of a mode m_i and $[\mathrm{pc}(\mathrm{m}_i)]_{\mathrm{system}} > 0$.

Both types of deviations between model and system influence the accuracy of energy demand predictions $\mathrm{dev}_{\mathrm{energy}}$ (Equ. 9.5). Caused by deviations, the energy demand is overestimated or underestimated by the strategy.

$$\mathrm{dev}_{\mathrm{energy}}[\%] = \left[\frac{\mathrm{eni}_{\mathrm{model}}}{\mathrm{eni}_{\mathrm{system}}} - 1 \right] \cdot 100 \tag{9.5}$$

with $\mathrm{eni}_{\mathrm{system}} > 0$.

If $\mathrm{dev}_{\mathrm{energy}} > 0$, the energy input of the system $\mathrm{eni}_{\mathrm{system}}$ is overestimated by the planned strategy. Otherwise, the energy input $\mathrm{eni}_{\mathrm{system}}$ is underestimated by the planned strategy. A decision maker, needs to define adequacy criteria for $\mathrm{dev}_{\mathrm{energy}}$. An adequacy criterion for $\mathrm{dev}_{\mathrm{energy}}$ might be that the $\mathrm{dev}_{\mathrm{energy}}$ needs to remain in a specific interval (for instance $\mathrm{dev}_{\mathrm{energy}}$ = [-10%, +10%]). The influences on $\mathrm{eni}_{\mathrm{system}}$ (respectively $\mathrm{eni}_{\mathrm{model}}$) is summarized in Equation 9.6. The input power is measured n-times ($\mathrm{pc}_i(\mathrm{m}^j)_{\mathrm{system}}$), so that the pause interval $\mathrm{pa}(\mathrm{sub}_j)^{\mathrm{actual}}$ is devided into n elementary time intervals. Calculating the product of each elementary time interval and the measured input power, the energy demand of a subsystem sub_j can be determined. The overall energy demand of the system $\mathrm{eni}_{\mathrm{system}}$ is the sum over m subsystems.

$$\mathrm{eni}_{\mathrm{system}} = \sum_{i=1}^{n} \frac{\mathrm{pa}(\mathrm{sub}_j)^{\mathrm{actual}}}{\mathrm{n}} \cdot \mathrm{pc}_i(\mathrm{m}^j)_{\mathrm{system}} \tag{9.6}$$

for j = 1, ..., m.

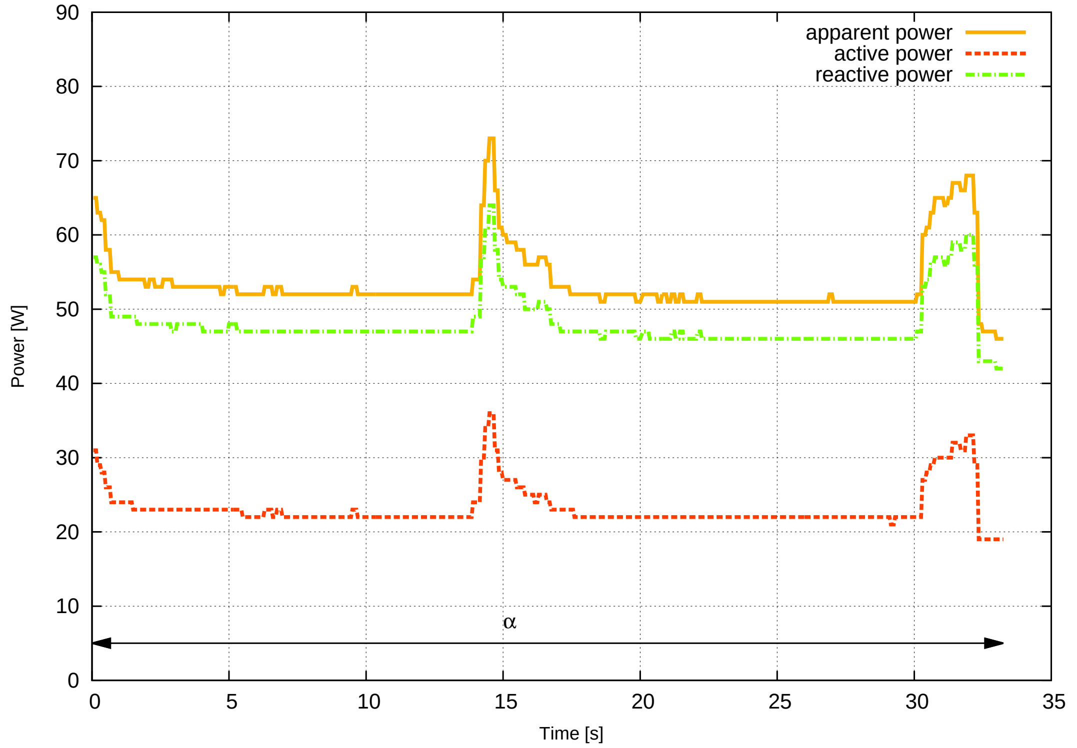
The validity of the energy demand prediction $\mathrm{dev}_{\mathrm{energy}}$ is evaluated in this section. Subsection 9.3.1 justifies the use of constant input powers for modes within unproductive phases based on measurements in Test Bed tb_S. The system model of Test Bed tb_S is validated by executing strategies in the test bed using direct experiments in Subsection 9.3.2. In Subsection 9.3.3, applying simulation-based experiments in Test Bed tb_M enables the separated evaluation of the two types of model-to-system deviations $\mathrm{dev}_{\mathrm{delay}}$ and $\mathrm{dev}_{\mathrm{power}}$.

9.3.1 Abstraction regarding constant input power of modes

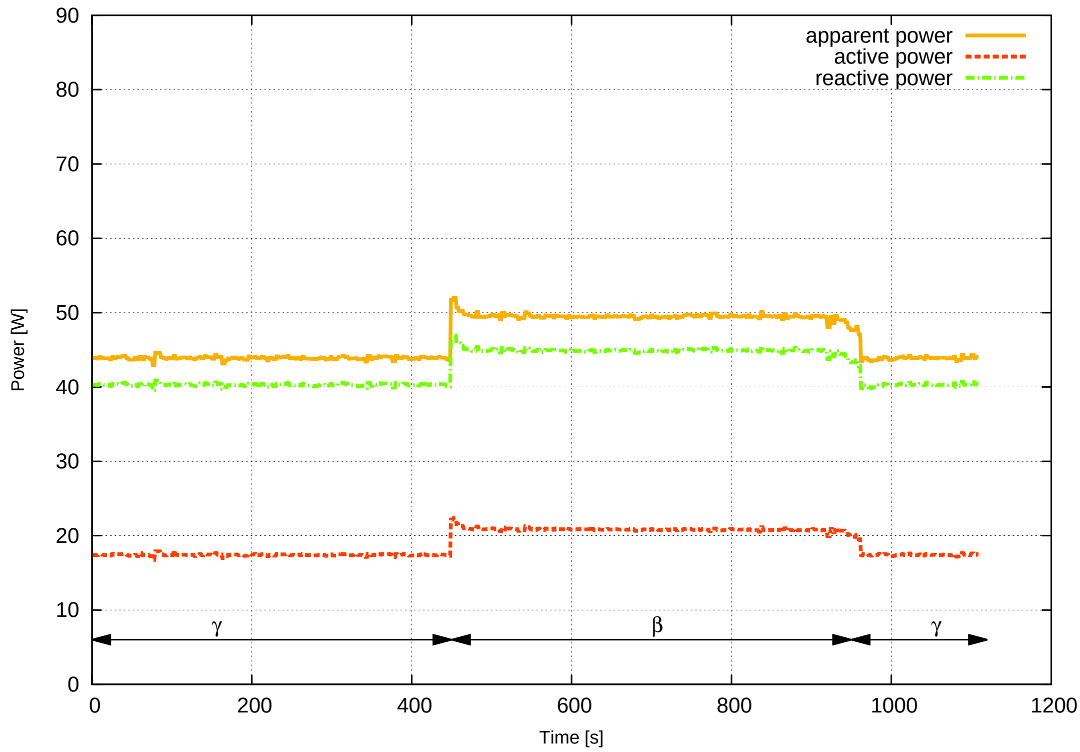
In Chapter 4, it is assumed that input power values of modes are constant (Def. 28, Subsection 4.2.2). This assumption regarding constant input power of modes within unproductive phases is checked by measurements in Test Bed tb_S.
The input power over time of the electric motor in Test Bed tb_S (Subsection 8.3.1) paints the following picture shown in Figure 9.9. The measured input power of the electric motor is plotted over time exhibiting different modes (α as productive mode, Fig. 9.9a and β, γ as unproductive modes, Fig. 9.9b). If effects of deferral of the reactive power are neglected, there exists the following connection between apparent power, active power, and reactive power (Equ. 9.7).

$$\mathrm{pc}_{apparent} = \sqrt{(\mathrm{pc}_{active})^2 + (\mathrm{pc}_{reactive})^2} \tag{9.7}$$

with $\mathrm{pc}_{apparent}$ as apparent power, pc_{active} as active power, and $\mathrm{pc}_{reactive}$ as reactive power.
The input power of the two idling modes β and γ demonstrate stability (Fig. 9.9b) whereas the input power of the productive mode α varies over time (Fig. 9.9a). The variation of input power during productive phases of an automation system depends on the individual production process.



(a) Productive phase



(b) Unproductive phase

Figure 9.9: Productive and unproductive phases of the electric motor of Test Bed tb_S

Based on the measurement of the electric motor, the input power of modes within unproductive phases can be assumed as approximately constant over time. Figure 9.9 (b) shows slightly increased input powers while changing from γ to β mode which is caused by switch-on currents (time point t_1 = 450 [s] in Fig. 9.9b). Switch-on currents while changing from one unproductive mode to another unproductive mode can cause a short increase of the input power. This short increase of input power is approximated using constant input power values of modes presented in Subsection 9.3.2.

9.3.2 Accuracy of energy demand prediction for Test Bed tb_S

Validating the system model of Test Bed tb_S (Appendix A.1), five different strategies are executed in Test Bed tb_S which have been checked for feasibility in Subsection 9.2.1. Each strategy is generated for a specific pause interval (Tab. 9.9). Combined with each strategy, the predicted energy demand eni_{model} and the actual energy demand eni_{system} after strategy execution is available. Using Equation 9.5, the objective is to quantify the overestimation respectively the underestimation dev_{energy} of the actual energy demand.

Serving as example[2], the planned and executed strategy for Scenario *Ver.tbS.3* is graphically presented as a course of input power over time for a 60 [s] unproductive phase (Fig. 9.10). The succession of modes is illustrated as double arrows with the name of the mode annotated showing that the planned and actual strategy comprise the same succession of modes.

The planned strategy has an energy demand of 4.840 [J] whereas the measured energy demand after execution of the strategy is 4.904 [J]. This results in an underestimation of the actual energy demand of dev_{energy} = -1,3%. The underestimation of the actual energy demand is related to dev_{delay} = -6,8% and dev_{power} = -0,1%.

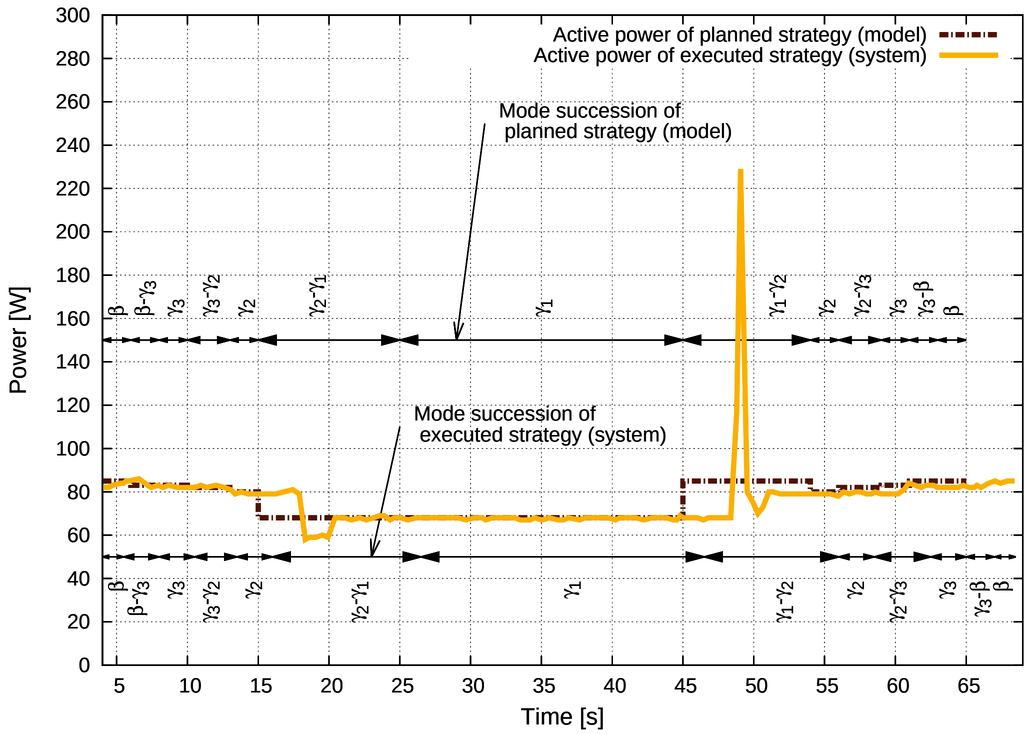


Figure 9.10: *Ver.tbS.3*: Planned (model) and actual (system) input power in Test Bed tb_S

One important aspect related to the execution of strategies are switch-on currents. Figure 9.10 illustrates this fact at time point t = *49 [s]* while switching from mode γ_1 to mode γ_2 caused by activation of *P1.2* (electric motor). This peak is approximated in the model by computing the

[2]The input power over time for Scenarios *Ver.tbS.1* and *Ver.tbS.2* can be found in Appendix C.3.1.

average input power of the system between *t = 46 [s]* and *t = 56 [s]*. This results in an average input power of *pc = 85 [W]* for mode $\gamma_1\_\gamma_2$ which is used in the system model (compare system model in Fig. A.1).

Table 9.11 illustrates the estimation of the actual energy demand for the Scenarios *Ver.tbS.1* to *Ver.tbS.5*. The validation tests reveal that in Scenario *Ver.tbS.1* and Scenario *Ver.tbS.2*, the system model matches the system behavior less than in Scenarios *Ver.tbS.3* to *Ver.tbS.5*. This is due to the fact that within short unproductive phases (*Ver.tbS.1*: 10 [s], *Ver.tbS.2*: 30 [s]), model-to-system deviations have bigger effects on the accuracy of energy demand prediction than within longer unproductive phases (for instance *Ver.tbS.4*: 300 [s]).

Scen.	**Pause interval** [s]	$\mathbf{eni_{model}}$ [J]	$\mathbf{eni_{system}}$ [J]	$\mathbf{dev_{energy}}$ [%]	$\mathbf{dev_{delay}}$ [%]	$\mathbf{dev_{power}}$ [%]
Ver.tbS.1	10	840	939	-10,5	-10,7	0,2
Ver.tbS.2	20	2.455	2.723	-9,8	-10,2	0,6
Ver.tbS.3#	60	4.840	4.868 ± 51	-0,6 ± 1,0	-6,4 ± 0,3	0,2 ± 2,0
Ver.tbS.4#	300	20.340	20.530 ± 277	-0,9 ± 1,3	-1,3 ± 0,1	0,2 ± 1,4
Ver.tbS.5#	600	40.140	40.133 ± 106	0,0 ± 0,3	-0,8 ± 0,1	0,9 ± 0,1

Table 9.11: Energy demand of planned and executed strategies, #average, n = 4

In Scenarios *Ver.tbS.3* through *Ver.tbS.5*, the system model has a good accuracy regarding the description of the system behavior.

9.3.3 Accuracy of energy demand prediction for Test Bed $\mathbf{tb}_M$

For Test Bed tb_M, the accuracy of the energy demand prediction is analyzed using the simulation-based environment (Subsection 8.3.2). Mode delay deviations and input power deviations are evaluated towards the impact on the energy demand prediction separately.

Mode delay deviations

Mode delay deviations dev_{delay} between model and system occur, if time constraints in the model do not exactly correspond to the temporal behavior of the system. This type of deviation is analyzed for Test Bed tb_M in different scenarios (Tab. 9.10, Subsection 9.2.2). Figure 9.11 reveals the typical staircase-shaped course of input power of computed strategies[3] (Scenarios *Ver.tbM.0* and *Ver.tbM.7*) for Subsystem sub_9 in Test Bed tb_M. Mode delay deviations dev_{delay} result in a time shift of input power of the model compared to the actual input power of the system. While analyzing mode delay deviations dev_{delay}, there are no deviations regarding the input power dev_{power}.

[3] Input power over time for Subsystems sub_1 to sub_8 can be found in Appendix C.3.2

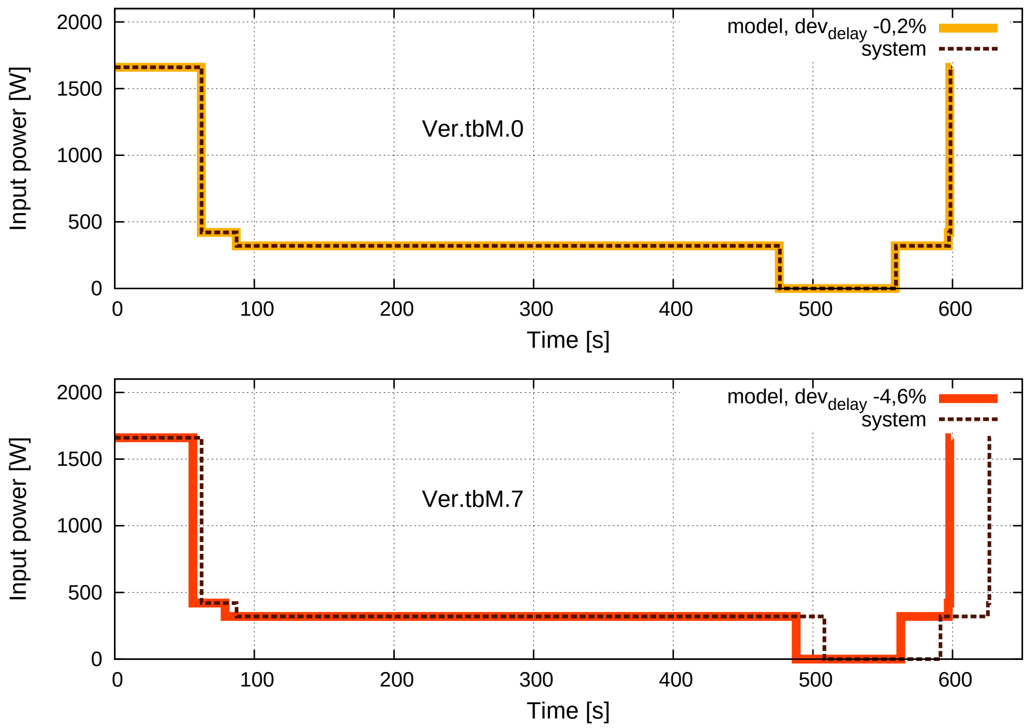


Figure 9.11: *Ver.tbM.0 and Ver.tbM.7*: Planned and actual input power of Subsystem sub_9 plotted over time

In order to link average mode delay deviations dev_{delay} to the accuracy of energy demand prediction dev_{energy} (overestimation/underestimation), Figure 9.12 sums up graphically the results of Table 9.10. For each Scenario in Table 9.10, the mode delay deviation dev_{delay} and its standard deviation as well as the related accuracy of energy demand prediction dev_{energy} is given.

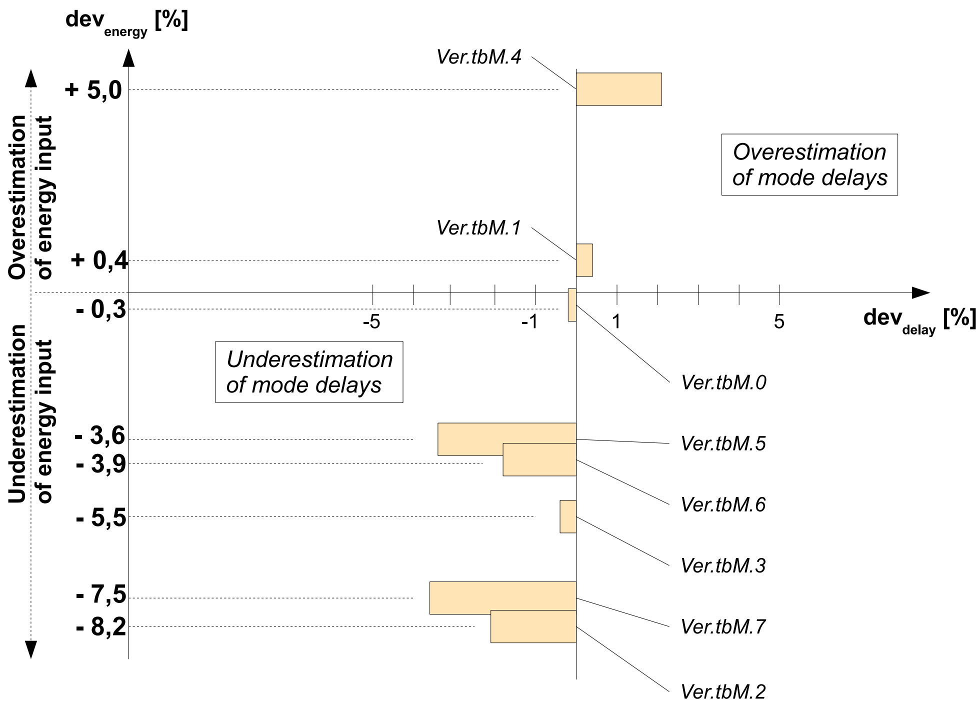


Figure 9.12: *Ver.tbM.x*: Impact on overestimation and underestimation of energy input caused by mode delay deviations dev_{delay}

Figure 9.12 reveals that if the mode delay is overestimated (positive dev_{delay}), the actual energy demand is also overestimated (positive dev_{energy}). If the average mode delay is underestimated (negative dev_{delay}), the actual energy demand is also underestimated (negative dev_{energy}). Comparing Scenario *Ver.tbM.5* with Scenario *Ver.tbM.7*, dev_{delay} is almost the same, but results in different values for dev_{energy}. Mode delay deviations are causally connected to a certain degree of stochastics regarding the estimation of the energy input.

Input power deviations

Input power deviations dev_{power} between model and system occur, if input power values in the system model do not exactly correspond to those of the system. To analyze the impact of input power deviations dev_{power} on the accuracy of energy demand prediction dev_{energy}, Scenarios *Val.tbM.0* to *Val.tbM.6* are used (Tab. C.1, Appendix C.3.3). Figure 9.13 presents the input power over time of Subsystem sub_9 in Test Bed[4] tb_M with dev_{power} = 30%.

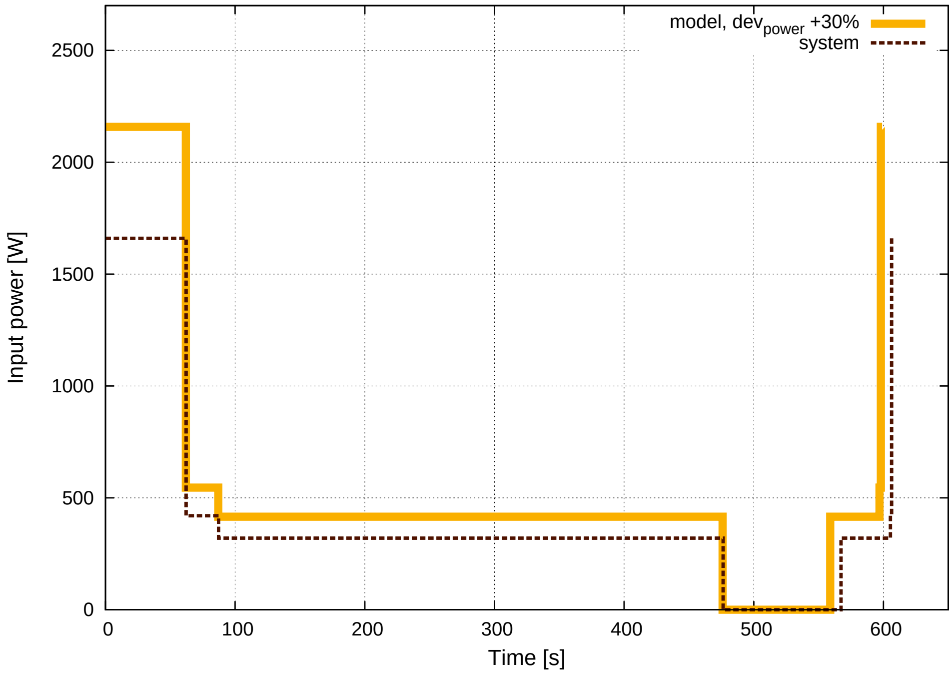


Figure 9.13: *Val.tbM.3*: Modeled input power and actual input power of Subsystem sub_9

[4]Input power over time for Subsystems sub_1 to sub_8 can be found in Appendix C.3.3

Evaluating the influences of modeled input power values of modes on the accuracy of energy demand prediction, Figure 9.14 sums up graphically the results shown in Table C.1 for Scenarios *Val.tbM.0* to *Val.tbM.6*.

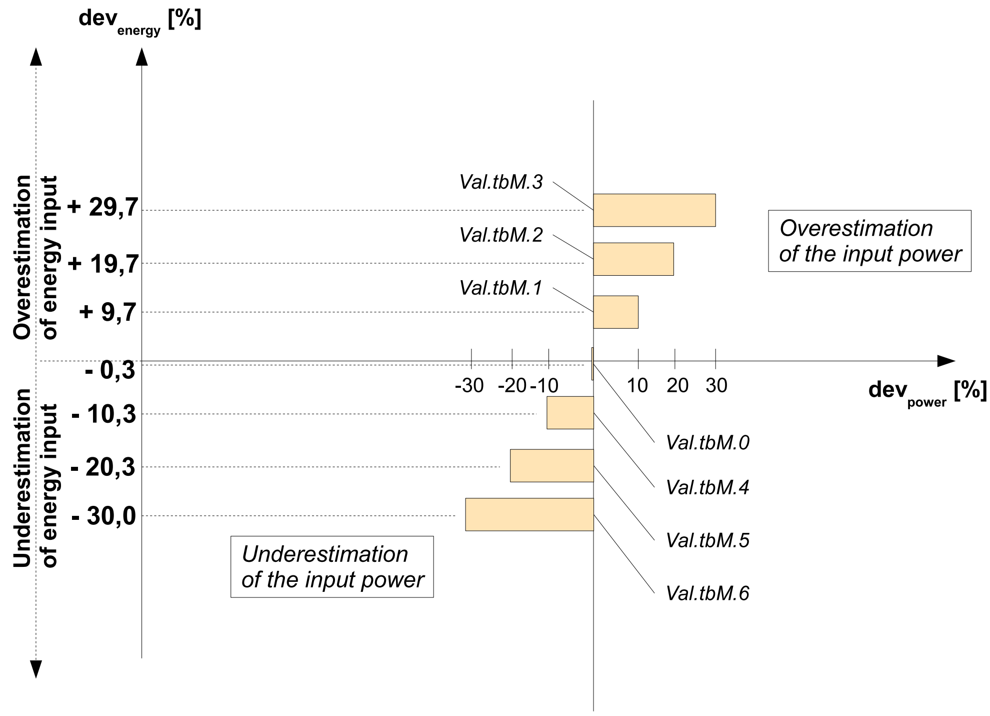


Figure 9.14: *Val.tbM.x*: Impact on overestimation and underestimation of energy demand caused by input power deviations

Figure 9.14 shows that the energy demand of the system is linearly overestimated by an overestimation of the input powers. In case of underestimation of the input power, the energy demand is underestimated. The assumptions in the system model regarding input powers have no impacts on the execution of a strategy, so that this linear causality can be stated.

9.3.4 Summary

Using validation tests, the accuracy of the energy demand prediction by strategies has been evaluated. The parameters dev_{delay} and dev_{power} have been examined regarding their impact on the accuracy of energy demand prediction quantified by dev_{energy}. The accuracy of the prediction[5] of the energy demand dev_{energy} is influenced by modeling assumptions (Conclusion 3).

[5]Evaluation data in Table 9.11 and Table C.1

Conclusion 3 (Overestimation/underestimation of energy demand quantified by dev_{energy})
Overestimation/underestimation of energy demand dev_{energy} is affected by deviations

- *dev_{delay} caused by modeling assumptions:*
  - *transitions between modes being taken instantly*
  - *neglected communication delay between strategy execution and supervision and the test bed*

  The bigger the pause intervals of strategies are, the smaller the effects are on dev_{energy} caused by dev_{delay}. An overestimation (underestimation) of mode delays results in a stochastic overestimation (underestimation) dev_{energy} of actual energy demands.

- *dev_{power} caused by modeling assumptions:*
  - *input power of modes being assumed as constant*
  - *switch-on currents being assumed as constant*

  Input power deviations dev_{power} cause a proportional overestimation respectively underestimation dev_{energy} of the energy demand.

Having analyzed the main parameters influencing the accuracy of energy demand prediction, it can be stated that a system model based on the modeling approach of Chapter 4 provides the possibility to describe the behavior of automation systems in an accurate way if pause intervals have an adequate length. Strategies can be used to calibrate and validate the information contained in the system model incorporating a predefined adequacy criterion.

Besides the influence on accuracy of predicted energy demands, the economic impact of strategies needs to be evaluated in order to quantify the benefit of the approach proposed in Part II. The economic perspective is taken in Section 9.4.

9.4 Reduction of energy demands by strategies

Using strategies for energy demand reduction is a key motivation. Energy efficiency is increased if a strategy is capable of reducing the energy demand during unproductive phases of an automation system (Section 1.3). The amount of energy that can be reduced by strategies is evaluated in this section by comparing the strategy-based approach of Part II with the state-of-the-art method which means to leave the automation system in an idle mode.
Subsection 9.4.1 presents results of economic considerations regarding Test Bed tb_S whereas Subsection 9.4.2 deals with economies in Test Bed tb_M.

9.4.1 Economies in Test Bed tb_S

The actual energy demand of the examined strategies in Table 9.9 is used to identify the energy savings potential realizable by the proposed approach of Part II. In Table 9.12 the energy input of Test Bed tb_S with applied strategies eni_{str} and without using strategies eni_{idl} is presented. A negative ens_{pau}^{rel} denotes energy savings, a positive ens_{pau}^{rel} denotes additional energetic effort produced by a strategy. The renouncement of strategies results in leaving the test bed in mode β related to eni_{idl}. Note, that Test Bed tb_S has no dedicated off mode ε.

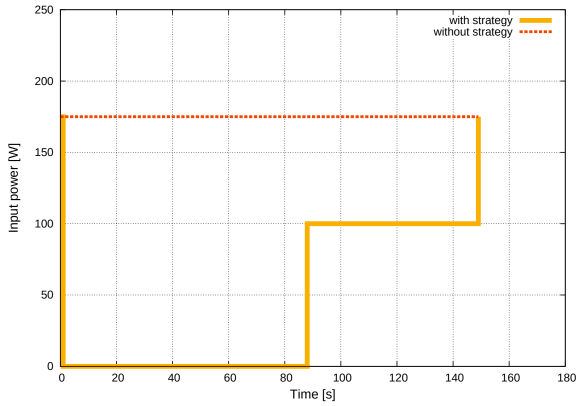
Scen.	Strategy	**Pause [s]**	$\mathbf{eni_{str}}$ **[J]**	$\mathbf{eni_{idl}}$ **[J]**	$\mathbf{ens_{pau}^{rel}}$ **[%]**
Ver.tbS.1	l_1	10	939	856	+9,7
Ver.tbS.2	l_2	30	2.723	2.569	+5,9
Ver.tbS.3	l_3	60	4.868#	5.140	-5,2
Ver.tbS.4	l_4	300	20.530#	25.698	-20,1
Ver.tbS.5	l_5	600	40.133#	51.396	-21,9

Table 9.12: Test Bed tb_S using strategies (energy demand eni_{str}) and idling of Test Bed tb_S (energy demand eni_{idl}), #average, n = 4

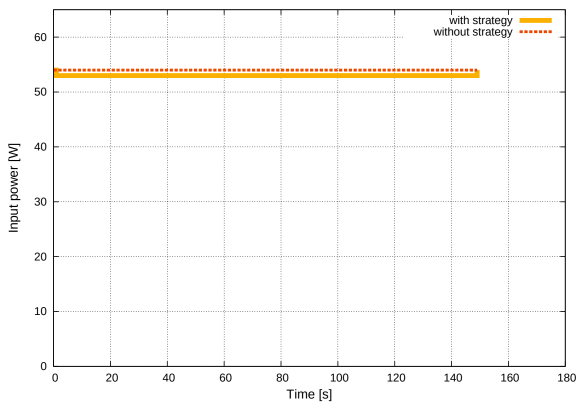
The amount of energy to be applied for Strategy l_1 and l_2 is greater than the energy demand for idling. Consequently, both strategies are not beneficial. This corresponds with the findings in Subsection 9.3.2 where the energy demand prediction by Strategies l_1 and l_2 has a rather weak accuracy caused by deviations dev_{delay} within the short pause interval of 10 [s] respectively 30 [s]. Explained by this weak accuracy, strategies for very short unproductive phases may give wrong indication about the energy savings potential.
Quite the contrary, applying Strategies l_3, l_4, and l_5, the energy demand of Test Bed tb_S can be reduced by |5,2%| (unproductive phase of 60 [s]), |20,1%| (unproductive phase of 300 [s]), and |21,9%| (unproductive phase of 600 [s]). In these scenarios, effects of dev_{delay} and dev_{power} are reduced certifying a good accuracy.

9.4.2 Economies in Test Bed tb_M

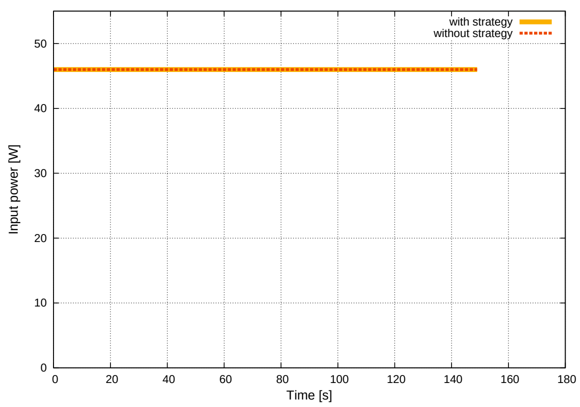
Analyzing the economies in Test Bed tb_M, it is distinguished between long-term ($\geq$ 30 minutes, Tab. C.3) and short-term (< 30 minutes, Tab. C.2) unproductive phases.
To give an example, the input power over time is presented for the nine subsystems of Test Bed tb_M in Figure 9.15 and Figure 9.16. It is distinguished between the input power over time of a strategy (continuous line, *with strategy*) and idling of the subsystem (dotted line, *without strategy*). The example testifies that for this short-term pause interval of 2,5 minutes, the complete shutdown of each subsystem is not feasible. The application of a strategy still corresponds to a relative energy reduction of ens^{rel}_{pau} = -79,7%.



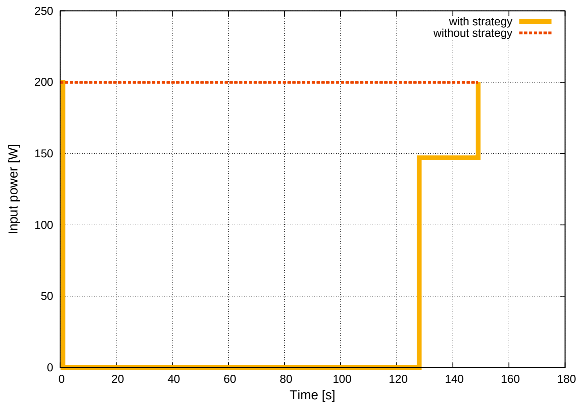
(a) Subsystem sub_1



(b) Subsystem sub_2

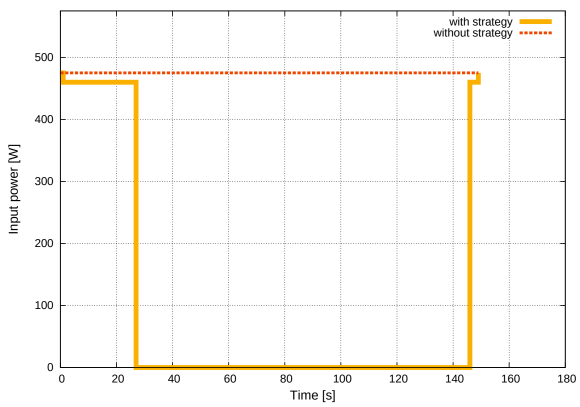


(c) Subsystem sub_3

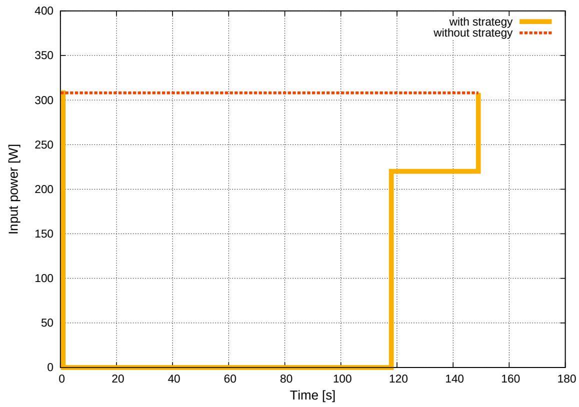


(d) Subsystem sub_4

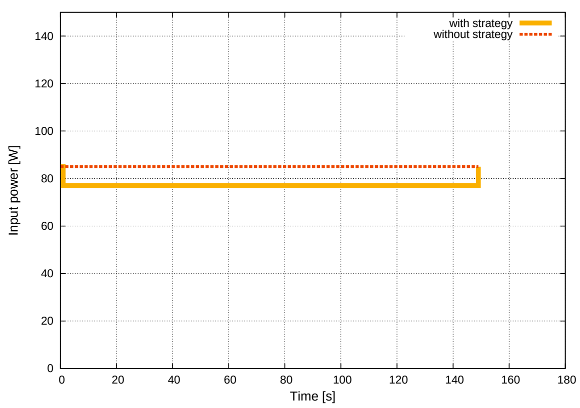
Figure 9.15: *Eco.tbM.1.1*: Input power over time during a 2,5 minutes pause interval, with and without strategies for Subsystems sub_1 to sub_4



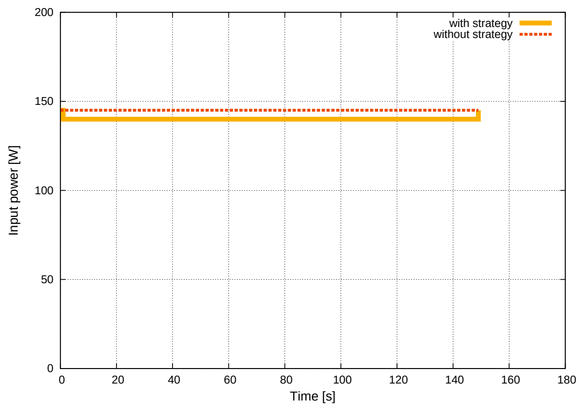
(a) Subsystem sub_5



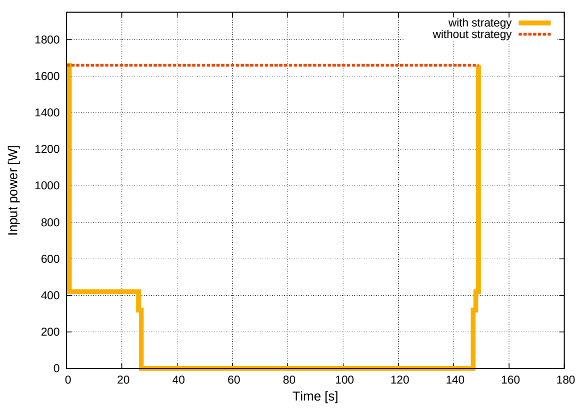
(b) Subsystem sub_6



(c) Subsystem sub_7



(d) Subsystem sub_8



(e) Subsystem sub_9

Figure 9.16: *Eco.tbM.1.1*: Input power over time during a 2,5 minutes pause interval, with and without strategies for Subsystems sub_5 to sub_9

In Figure 9.17, the absolute energy savings potential ens^{abs}_{pau} is illustrated for strategies with different length of unproductive phases. The savings potentials arises from the difference between eni_{idl} and eni_{str}. The energy demand realized by strategies testify a bigger savings potential for long-term pause intervals (Fig. 9.17 (b)). In the considered long-term pause intervals, eni_{str} is uniformly at 55 [kJ] for all intervals since a complete shutdown of Test Bed tb_M is feasible. This is due to the fact that once reaching the complete off mode in the system, input power ceases. In long-term pause intervals, the complete switch-off can always be initiated. In contrast to that, in short-term pause intervals, the complete-off mode ε can not always be reached (Fig. 9.15 and Fig. 9.16), so that this results in different degrees of energy savings. The relative energy savings within a short-term pause interval is between ens^{rel}_{pau} = 79,7% (2,5 minutes interval) and ens^{rel}_{pau} = 94,5% (20 minutes interval) of the reference value eni_{idl} (Tab. C.2). In the context of long-term pause intervals, ens^{rel}_{pau} is between ens^{rel}_{pau} = 98,2% (30 minutes interval) and ens^{rel}_{pau} = 99,8% (420 minutes interval) of eni_{idl}.

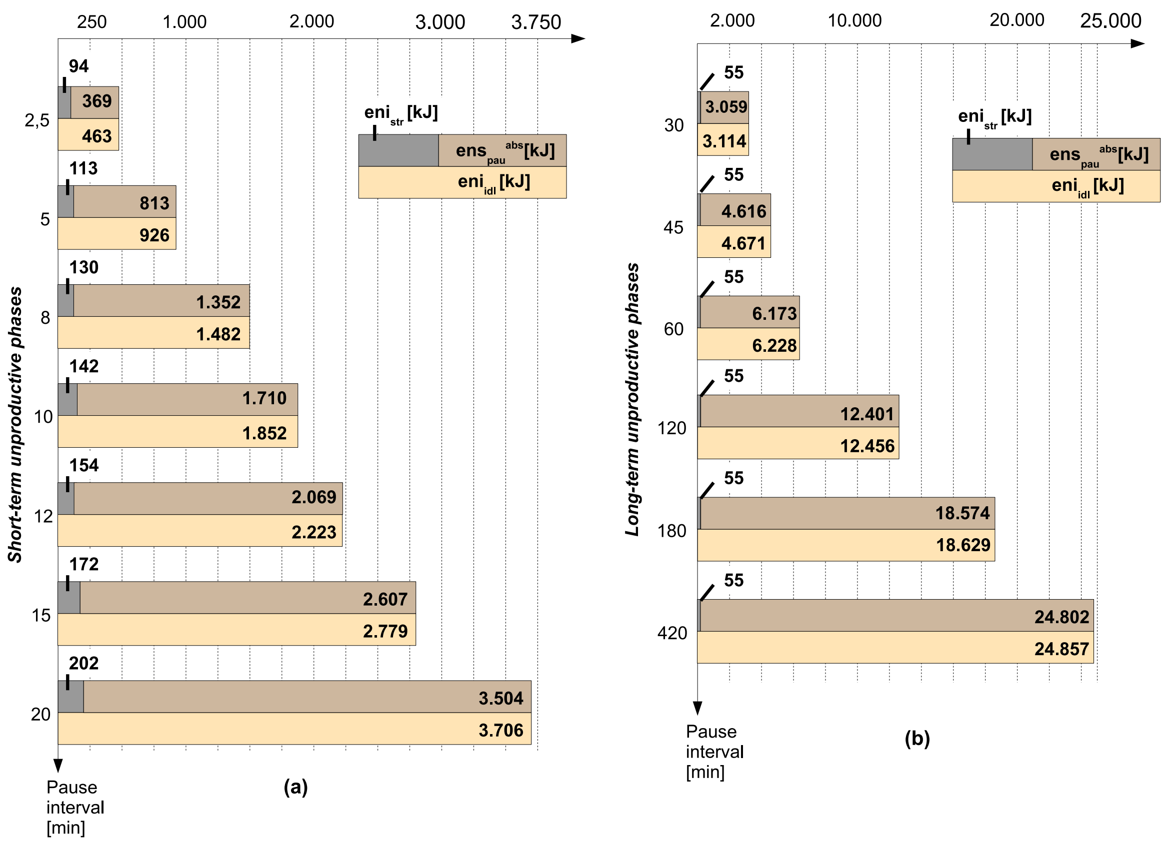


Figure 9.17: Energy savings ens^{abs}_{pau} in Test Bed tb_M for (a) short-term unproductive phases and (b) long-term unproductive phases

9.4.3 Summary

In this section, the economic impact of strategy usage has been considered. The energy savings for two typical automation systems have been analyzed (Subsection 9.4.1 and Subsection 9.4.2). With regard to the analyzed scenarios, the energy savings realizable in Test Bed tb_S are between 4,8% (Scenario *Ver.tbS.3*) and 21,6% (Scenario *Ver.tbS.5*) regarding unproductive phases. Bigger energy savings are realizable depending on the considered pause intervals.
The scenarios for Test Bed tb_M have revealed realizable energy savings between 79,7% (Scenario *Eco.tbM.1.1*) and 99,8% (Scenario *Eco.tbM.2.6*) regarding unproductive phases. Since Test Bed tb_S has no dedicated off mode ε, the relative energy savings are less than in Test Bed tb_M. Nevertheless, the reduction of the energy demand using strategies can be quantified and an essential amount of energy can be saved using the approach proposed in Part II (Conclusion 4).

Conclusion 4 (Reduction of the energy demand in unproductive phases using strategies)
Depending on the available length of the unproductive phase and the internal behavior of the system, significant energy savings are technically and fully automated realizable using the approach of Part II.

Chapter 10

Summary, conclusion and outlook

Summary and conclusion

In the recent years, energy-efficiency has been evolved into one of the top political and social issues in Germany and Europe [EU-Directive, 2012] which affects industrial production [DENA, 2013]. Providing an approach to reduce energy demands in unproductive phases, this thesis gives an answer to more energy efficiency in industrial automation. The approach of this thesis establishes sound insights into the energetical behavior of automation systems within unproductive phases. Moreover, the approach comprises a proposition for the technical realization so that the theoretical energy savings potentials can actually be raised.
The key findings of this thesis are summarized in this section reflecting the research objectives and scientific contribution introduced in Section 1.3.

System model for energy-centric description of automation systems

In Chapter 4, an automaton-based system model has been introduced enabling the formal description of structural aspects and energetical as well as temporal behavior of automation subsystems. This system model provides modeling power to describe automation systems in a natural way and serves as information basis for the derivation of strategies. With respect to technical practicability, the modes of the automaton-based model can be transformed to IEC-61131-3 templates (Chapter 7). In this way, the proposed approach supports the implementation phase. The applicability and realization of the modeling approach has been exemplified using two test beds.
In order to back up the engineering of automation systems, energetical aspects need to be considered in early phases of the design process. The proposed automaton-based approach efficiently supports the energetical engineering of automation systems. The modeling approach enables a graphical and natural description of the structural and behavioral aspects of the automation system. During engineering, subsystem dependencies are made explicit as part of

the model. For this, the automaton-based system model provides a natural way to describe the structural and energetical aspects. The model is meant to provide means for analytical analysis of the automation system using computerized strategies.

Computerized strategies for unproductive phases

The system model provides the information basis for deriving energy-optimal strategies in order to address unproductive phases (Chapter 5). Strategies are a specification of the succession of modes respecting structural and temporal facts of the system model. Applying parametrization (initial and target mode of subsystems and a given unproductive phase), optimal strategies are computed and are executed fully automated in the automation system (Chapter 7). In this way, the proposed strategy-based approach contributes to the research objectives raised in the introductory Section 1.3.

Identification of optimal strategies

The identification of optimal strategies is efficiently supported by the procedure proposed in Chapter 6. The procedure uses the structural knowledge (modularity) of the automation system to investigate most promising strategies in the set of alternative strategies, first.
That way, the computational runtime required for the identification of the optimal strategy within a given unproductive phase is accelerated (Section 9.1). The comparison with the complete enumeration approach proves that the optimal strategy is identified by the procedure based on the bounded investigation of the set of strategies. Compared to an optimization model without using structural knowledge, the proposed selective procedure significantly reduces the computational runtime to identify the optimal strategy. Consequently, it has been shown that the proposed procedure in this thesis efficiently supports the identification of optimal strategies for industrial settings.

Specification of feasible strategies

Strategies need to be executed in the way they are specified. This comprises the guarantee that the succession of modes is executed in the system as intended while structural and temporal facts of the subsystems are considered. Model-to-system deviations complicate the execution of computed strategies. Therefore, a robustness modification of computed strategies before execution has been proposed (Section 7.4).
In Section 9.2, the feasibility of strategies has been shown by direct experiments in Test Bed tb_S and simulation-based experiments in Test Bed tb_M. The feasibility of strategies in the system influenced by mode delay deviations has been checked. Despite of model-to-system deviations, in the evaluated scenarios, every planned strategy has been actually feasible.

Model validation using strategies

The guarantee of feasibility is a necessity, but not sufficient to reduce the energy demand within unproductive phases. The validity of the system model respectively the accuracy of energy demand predictions needs to be ensured. Mode delay deviations and input power deviations have been evaluated regarding their impact on energy demand predictions. Realizing strategies in the test beds, it has been shown that the proposed approach provides appropriate means to calibrate and validate the energetical and temporal aspects of automation systems in a system model.

Reduction of energy demands by strategies

A key motivation of using strategies within unproductive phases is the reduction of the energy demand. In general, automation subsystems are left in idling modes within unproductive phases. As it is motivated in Section 1.2, this state-of-the-art method to deal with unproductive phases does not completely exploit the energy savings potentials of these phases. Therefore, the energy demand is intended to be reduced by proposing strategies. As key element of the approach of this thesis, the energy demand of a strategy to be executed can be quantified and can be compared to other alternatives, for instance system idling. Additionally, a precises specification of the temporal succession of modes is provided by the strategy-based approach of this thesis.

The economic impact is shown to be significant (Section 9.4). Using the approach of this thesis enables the quantification of the energy savings potentials for different pause intervals paired with a concrete strategy proposition in which way this potential can be exploited.

Outlook

The approach of this thesis can be extended in several ways. Therefore, this section provides suggestions in which way the approach of this thesis might be enriched.

Modeling guidelines and model libraries

Based on the conceptual elements of this approach (Section 4.1), a methodology for model generation can be developed. Setting up models for automation systems could stick to different procedures. A top-down approach might be practical for brownfield plants which are already existent. In this case, analyzing the structure of the automation system, the number and the composition of subsystems can be derived. Taking a closer look inside the control and functions of a subsystem, a specific number of modes can be identified with specific transitions between those modes. Measuring the input power of a mode and the times to changeover between modes reveals the energetical and temporal information necessary to annotate the automaton-based system model.

In contrast, the generation of models for greenfield plants which are engineered from scratch might follow a bottom-up approach. Considering the functions and the hardware internals of a subsystem, energetically distinguishable modes and respective transitions can be defined which need to be mapped to the control of the subsystem. Information about the planned subsystem interactions enable the derivation of subsystem dependencies to generate the system model.

Providing plant engineers with model libraries, the modeling effort can be reduced. These model libraries enable the reuse of subsystem models in similar production settings.

Automatic information feedback to the system model

As soon as strategies are executed in the system, computerized strategies can be used to calibrate the system model (Section 9.3). Since the subsystem behavior might change over time because of component aging and environmental impacts, the proposed approach can be enriched by automatic feedback of information from the system to the system model. Enriching the proposed approach during system runtime, there are enhancements conceivable. In order to make strategy predictions more precise and realistic, modeled information can be updated using current system knowledge and measurements regarding the measured input power of modes and the measured transitional times between modes. In form of a closed loop, current system data can be perpetually ascribed to the model.

Linking the approach to related planning levels

The implemented approach of this thesis addresses the energetical description of automation systems. The knowledge materialized in strategies can be useful for related planning levels. For instance, manufacturing execution systems or production planning systems might incorporate the information contained in system models of this approach. The detailed planning of production breaks which is basically conducted by manufacturing execution systems or production planning systems might use the knowledge about the energetical effects of unproductive phases to optimize the length of planned production breaks. The interactions of the proposed approach with related planning levels might lever further energy savings potentials.

Application of the approach for partial-load operation

The applicability of the proposed approach to partial-load operation within productive phases might be focus of further investigations. Partial-load operation of automation systems might be beneficial if the energy input to production is reduced while the system is running in a partial-load mode and the quality of the production output is not affected. Today, the final proof of energetical effectiveness of partial-load operation is missing. In this context, the proposed approach might be extended to address the energetical evaluation of partial-load operation within productive phases. For this purpose, the effects on production output need to be included into the system model introduced in this thesis. While computing strategies, the effects of partial-load operation are required to be incorporated to estimate the impacts, for instance on production throughput and production scheduling.

Determination of reasons for strategy recomputation during strategy execution

Caused by system behavior that has not been considered in the system model, it might happen that strategies are not feasible as it is planned. In this case, during execution of a strategy, a requirement for replanning a strategy based on new facts might occur. For instance, a subsystem might not behave like it is represented by the system model because a mode within a subsystem cannot be activated. To reach a given target mode within each subsystem, the dynamic reaction during runtime needs to be provided. There are two aspects that need to be considered to realize strategy recomputation during strategy execution.

First, key performance indicators are required to evaluate and define relevant reasons for recomputation of strategies during strategy execution. The available strategy which can be computed by the approach of this thesis can serve as basis to measure significant deviations as recomputation reasons.

Secondly, having identified a reason for recomputation of a strategy during execution, an appropriate infrastructure needs to implement the recomputed strategy. Implementing an automation system as peer-to-peer network might enrich the proposed approach concerning strategy execution. The centrally planned strategy ensures the optimality of the strategy whereas the peer-to-peer network provides means to detect errors during execution in a dynamic way. This could prevent a black-out of the automation system by local error recovery. That way, the central planning approach can be enriched by a decentral control of strategy execution.

Bibliography

Abele, E. and C. Eisele (2010). "Energieeffiziente Produktionsmaschinen durch Simulation in der Produktentwicklung". In: *Zeitschrift für wirtschaftlichen Fabrikbetrieb* 105(2010). Carl Hansen Verlag, München, pages 980–983 (cited on page 4).

ACPI (2011). *Advanced Configuration & Power Interface Specification 5.0*. Hewlett-Packard Corporation, Intel Corporation, Microsoft Corporation, Phoenix Technologies Ltd., Toshiba Corporation. URL: http://acpi.info/DOWNLOADS/ACPIspec50.pdf (visited on 04/22/2013) (cited on page 30).

Agora Energiewende (2013). *12 Insights on Germany's Energiewende*. Discussion paper. Stiftung Mercator, European Climate Foundation (cited on page 4).

Agricola, A. Cl., S. Joest, M. Czernie, R. Heuke, D. Kalinowska, S. Peters, J. Perner, and D. Bothe (2012). *Steigerung der Energieeffizienz mit Hilfe von Energieeffizienz-Verpflichtungssystemen*. Report. Deutsche Energie-Agentur GmbH, Frontier Economics Ltd. (cited on page 3).

Alfaro, Luca de and Thomas A. Henzinger (2001). "Interface automata". In: *Proc. of the 8th European software engineering conference held jointly with 9th ACM SIGSOFT international symposium on Foundations of software engineering, New York, USA*. ESEC/FSE-9. ACM, pages 109–120 (cited on page 40).

Alpern, B. and F. B. Schneider (1985). "Defining liveness". In: *Information Processing Letters* 21(4). Elsevier, pages 181–185 (cited on page 45).

Alur, R. and D. Dill (1990). "Automata for modeling real-time systems". In: *Automata, Languages and Programming*. Volume LNCS 443. Springer, pages 322–335 (cited on pages 43, 46).

Alur, R. and D. Dill (1994). "A theory of timed automata". In: *Theoretical Computer Science* 126(2), pages 183–235 (cited on pages 42, 65).

Alur, R., S. La Torre, and G. J. Pappas (2001). "Optimal paths in weighted timed automata". In: *Hybrid Systems: Computation and Control*. Volume LNCS 2034. Springer, pages 49–62 (cited on pages 47, 65).

Alur, R. and M. Parthasarathy (2004). “Decision Problems for Timed Automata: A Survey”. In: *Formal Methods for the Design of Real-Time Systems*. Volume LNCS 3185. Springer, pages 1–24 (cited on page 46).

Ambühl, C., M. Mastrolilli, N. Mutsanas, and O. Svensson (2011). “On the Approximability of Single-Machine Scheduling with Precedence Constraints”. In: *Mathematics of Operations Research* 36(4). INFORMS, pages 653–669 (cited on page 49).

Andradóttir, S. (1998). *Handbooks of Simulation: Principles, Methodology, Advances, Applications, and Practice*. Edited by J. Banks. John Wiley & Sons. Chapter Simulation optimization (cited on page 25).

Arpinen, T., E. Salminen, T. D. Hämäläinen, and M. Hännikäinen (2012). “MARTE profile extension for modeling dynamic power management of embedded systems”. In: *Journal of Systems Architecture* 58(5). Elsevier, pages 209–219 (cited on pages 29, 30).

Artigues, A and C. Haït (2011). “A hybrid CP/MILP method for scheduling with energy costs”. In: *European Journal of Industrial Engineering* 5(4). Inderscience Publishers, pages 471–489 (cited on pages 24, 51).

Baehre, D., M. Swat, P. Steuer, and K. Trapp (2011). “Energy Consumption: One Criterion for the Sustainable Design of Process Chains”. In: *The 9th Global Conference on Sustainable Manufacturing, St. Petersburg, Russia*. Edited by Günther Seliger. Springer, pages 164–170 (cited on page 20).

Beck, A. and P. Göhner (2011). “Modeling Knowledge for User-Centric Energy Cost Analysis of Industrial Automation Systems”. In: *23rd International Symposium on Information, Communication and Automation Technologies, Sarajevo, Bosnia and Herzegovina*, pages 1–8 (cited on page 26).

Beck, A. and P. Göhner (2012). “Analysis Levels for Improved User-Centric Energy Cost Analysis”. In: *7th Iberian Conference on Information Systems and Technologies, Madrid, Spain*, pages 1–6 (cited on page 26).

Behrmann, G. (2003). “Data Structures and Algorithms for the Analysis of Real Time Systems”. PhD thesis. Department of Computer Science, Aalborg University (cited on page 45).

Behrmann, G., A. Fehnker, T. Hune, K. Larsen, P. Pettersson, and J. Romijn (2001a). “Efficient Guiding Towards Cost – Optimality in UPPAAL”. In: *Tools and Algorithms for the Construction and Analysis of Systems*. Volume LNCS 2031. Springer, pages 174–188 (cited on page 47).

Behrmann, G., A. Fehnker, T. Hune, K. Larsen, P. Pettersson, J. Romijn, and F. Vaandrager (2001b). “Minimum-cost reachability for priced timed automata”.

In: *Hybrid Systems: Computation and Control*. Volume LNCS 2034. Springer, pages 147–161 (cited on pages 46, 47, 65).

Behrmann, G., K. G. Larsen, and J. I. Rasmussen (2006). "Priced Timed Automata: Algorithms and Applications". In: *Formal Methods for Components and Objects*. Volume LNCS 3657. Springer, pages 162–182 (cited on page 65).

Bellman, R. (1957). *Dynamic Programming*. Princeton University Press (cited on pages 45, 89).

Bellman, R. (1958). "On a routing problem". In: *Quarterly of Applied Mathematics* 16(1), pages 87–90 (cited on page 45).

Benders, J. F. (1962). "Partitioning procedures for solving mixed-variables programming problems". In: *Numerische Mathematik* 4, pages 238–252 (cited on page 53).

Bengtsson, J., K. Larsen, F. Larsson, P. Pettersson, and W. Yi (1996). "UPPAAL – a Tool Suite for Automatic Verification of Real-Time Systems". In: *Hybrid systems III*. Volume LNCS 1066. Springer, pages 232–243 (cited on page 109).

Benini, L., A. Bogliolo, and G. De Micheli (2000). "A Survey of Design Techniques for System–Level Dynamic Power Management". In: *IEEE Transactions on Very Large Scale Integration Systems* 8(3), pages 299–316 (cited on page 30).

Bérard, B., F. Cassez, S. Haddad, D. Lime, and O. H. Roux (2005). "Comparison of Different Semantics for Time Petri Nets". In: *Automated Technology for Verification and Analysis*. Volume LNCS 3707. Springer, pages 293–307 (cited on page 44).

Berger, R. (2009). *Global Automation Industry Study 2015*. Report. Roland Berger Strategy Consultants (cited on page 4).

Berger, R. (2010). *Automation – Time to find your true north*. Report. Roland Berger Strategy Consultants (cited on page 4).

Berger, U., D. Wolff, and D. Kulus (2012). "Einsatz digitaler Techniken zur Steigerung der Energieeffizienz in der Automobilindustrie". In: *zwf-online* 107. Carl Hanser Verlag, München, pages 587–590 (cited on page 25).

Berthomieu, B. and M. Menasche (1983). "An Enumerative Approach For Analyzing Time Petri Nets". In: *Proceedings IFIP, Paris, France*. Edited by R. E. A. Mason. Elsevier, pages 41–46 (cited on page 45).

Beyer, D. (2002). "Formale Verifikation von Realzeit-Systemen mittels Cottbus Timed Automata". PhD thesis. Cottbus: Faculty of Mathematics, Science, and Computer Science, Cottbus University of Technology (cited on page 47).

Beyer, D., C. Lewerentz, and A. Noack (2003).
"Rabbit: A Tool for BDD-Based Verification of Real-Time Systems".
In: *Computer-Aided Verification*. Volume LNCS 2725. Springer, pages 122–125
(cited on page 109).

BMBF (Aug. 2011). *Bundeskabinett verabschiedet 6. Energieforschungsprogramm*.
Bundesministerium für Bildung und Forschung.
URL: http://www.bmbf.de/press/3136.php (visited on 04/22/2013) (cited on page 3).

BMU (2011). *Einfluss der Umwel- und Klimapolitik auf die Energiekosten der Industrie – mit Fokus auf die EEG-Umlage*. Report.
Berlin: Bundesministerium für Umwelt, Naturschutz und Reaktorsicherheit
(cited on page 5).

BMU (2012). *Federal Environment Minister Röttgen launches Renewable Energy Platform*.
Federal Ministry for the Environment, Nature Conservation and Nuclear Safety.
URL: http://www.bmu.de/english/current_press_releases/pm/48836.php (visited on 12/04/2012) (cited on page 3).

BMWi (2010). *Energieeffizienz – Made in Germany – Energieeffizienz im Bereich Industrie, Gebäudeanwendungen und Verkehr*. Report.
Berlin: Bundesministerium für Wirtschaft und Technologie (BMWi) (cited on page 12).

BMWi (2011). *2. Nationaler Energieeffizienz-Aktionsplan (NEEAP) der Bundesrepublik Deutschland*.
Report. Berlin: Bundesministerium für Wirtschaft und Technologie (BMWi), Bundesamt für Wirtschaft und Ausfuhrkontrolle (BAFA), Bundesstelle für Energieeffizienz (BfEE)
(cited on page 7).

Bolton, W. (2009). *Programmable Logic Controllers*. Elsevier (cited on page 94).

Bouyer, P., E. Brinksma, and K. G. Larsen (2004). "Staying Alive as Cheaply as Possible".
In: *Hybrid Systems: Computation and Control*. Volume LNCS 2993. Springer, pages 203–218
(cited on page 29).

Bouyer, P., T. Brihaye, V. Bruyère, and J.-F. Raskin (2007).
"On the optimal reachability problem on weighted timed automata".
In: *Formal Methods in System Design* 31(2). Kluwer Academic Publishers, pages 135–175
(cited on page 29).

Bouyer, P., U. Fahrenberg, K. G. Larsen, N. Markey, and J. Srba (2008).
"Infinite Runs in Weighted Timed Automata with Energy Constraints". In: *Proc. of the 6th International Conference on Formal Modeling and Analysis of Timed Systems, Saint Malo, France*,
pages 33–47 (cited on page 29).

Bozga, M., C. Daws, O. Maler, A. Olivero, S. Tripakis, and S. Yovine (1998).
"Kronos: A model-checking tool for real-time systems".
In: *Formal Techniques in Real-Time and Fault-Tolerant Systems*. Volume LNCS 1486. Springer,
pages 298–302 (cited on page 109).

Bradley, S. P. (1977). "Large-Scale Systems". In: *Applied Mathematical Programming*. Addison Wesley, pages 363–409 (cited on pages 31, 51, 52).

Brost, M., D. Rosenfeld, and F. Vorholz (2012). "Lüge auf der Stromrechnung". In: *Die Zeit*(35) (August, 23 2012), pages 17–19 (cited on page 5).

Buchan, D. (2012). *The Energiewende - Germany's gamble*. The Oxford Institute for Energy Studies. URL: http://www.oxfordenergy.org/wpcms/wp-content/uploads/2012/06/SP-261.pdf (visited on 04/22/2013) (cited on page 3).

Büchi, R. (1962). "On a decision method in restricted second-order arithmetic". In: *International Congress on Logic, Methodology, and Philosophy of Science*. Stanford University Press, pages 1–12 (cited on page 43).

Bunse, K., M. Vodicka, P. Schönsleben, M. Brülhart, and F. O. Ernst (2010). "Integrating energy efficiency performance in production management – gap analysis between industrial needs and scientific literature". In: *Journal of Cleaner Production* 19(6-7). Elsevier, pages 667–679 (cited on page 11).

Castro, P. M. and I. E. Grossmann (2005). "New continuous-time MILP model for the short-term scheduling of multistage batch plants". In: *Industrial & engineering chemistry research* 44, pages 9175–9190 (cited on page 50).

Castro, P. M., I. Harjunkoski, and I. E. Grossmann (2009). "A new continuous-time scheduling formulation for continuous plants under variable electricity cost". In: *Industrial and Engineering Chemistry Research* 48(14). American Chemical Society, pages 6701–6714 (cited on page 20).

Castro, P. M., I. Harjunkoski, and I. E. Grossmann (2011). "Optimal scheduling of continuous plants with energy constraints". In: *Computers & Chemical Engineering* 35(2). Elsevier, pages 372–387 (cited on page 20).

Chen, T. and C. Sin (1990). "A state-of-the-art review of parallel-machine scheduling research". In: *European Journal of Operational Research* 47(3). Elsevier, pages 271–291 (cited on page 49).

Chen, Y.-K. (2012). "Challenges and Opportunities of Internet of Things". In: *17th Asia and South Pacific Design Automation Conference, Sidney, Australia*, pages 383–388 (cited on page 31).

Chen, Z.-L. and W. B. Powell (1999). "Solving parallel machine scheduling problems by column generation". In: *Journal on Computing* 11(1). INFORMS, pages 78–94 (cited on page 52).

Chiotellis, S., N. Weinert, and G. Seliger (2010). "Simulation-based, energy-aware production planning". In: *Sustainable Production and Logistics in Global Networks*. Edited by Kuhland Sihn. Vienna, Austria: Neuer Wissenschaftlicher Verlag, pages 964–971 (cited on page 20).

Chong, E. K. P. and S. H. Zak (2013). *An Introduction to Optimization*. 4th edition. John Wiley & Sons (cited on page 47).

Conti, J. and P. Holtberg (2011). *International Energy Outlook 2011*. Report. Washington D. C.: U. S. Energy Information Administration (cited on page 3).

Cook, W. J., W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver (1998). *Combinatorial Optimization*. 2nd edition. John Wiley & Sons (cited on page 48).

DENA (2013). *Umfrage zur Energieeffizienz bei Entscheidungsträgern aus Unternehmen in Industrie und Gewerbe*. Report, 250 Unternehmen befragt, Erhebungszeitraum: 27.11.2012 bis 07.12.2012, Durchführung: mindline Energy GmbH. Pressemitteilung vom 02.04.2013. Deutsche Energie-Agentur. URL: http://www.stromeffizienz.de/fileadmin/user_upload/presse/medienmaterial/dateien/Umfrage-Investitionen-Energieeffizienz-Hintergrundinfo.pdf (visited on 04/25/2013) (cited on page 149).

Deng, G. (2007). "Simulation-based Optimization". PhD thesis. Madison: University of Wisconsin (cited on page 25).

Devoldere, T., W. Dewulf, W. Deprez, B. Willems, and J. R. Duflou (2007). "Improvement Potential for Energy Consumption in Discrete Part Production Machines". In: *Advances in Life Cycle Engineering for Sustainable Manufacturing Businesses*. Springer, pages 311–316 (cited on page 28).

Devoldere, T., W. Dewulf, W. Deprez, and J. R. Duflou (2008). "Energy Related Life Cycle Impact and Cost Reduction Opportunities in Machine Design: The Laser Cutting Case". In: *Proc. of the 15th CIRP International Conference on Life Cycle Engineering, Sidney, Australia*, pages 412–419 (cited on page 28).

DFG. *FOR1088'ECOMATION'*. Institute for Control Engineering of Machine Tools and Manufacturing, Stuttgart. URL: http://www.isw.uni-stuttgart.de/forschung/oeffentliche-projekte-laufend/dfg-forschergruppe-ecomation/ (visited on 12/12/2012) (cited on page 3).

Dierks, H. (2006). "Time, Abstraction and Heuristics". Habilitation. Oldenburg: Department of Computer Science, University of Oldenburg (cited on pages 47, 109).

Dietmair, A. and A. Verl (2009a). "A generic energy consumption model for decision making and energy efficiency optimisation in manufacturing". In: *International Journal of Sustainable Engineering* 2(2). Taylor & Francis, pages 123–133 (cited on page 28).

Dietmair, A. and A. Verl (2009b). "Energy Consumption Forecasting and Optimization for Tool Machines". In: *Modern Machinery Science Journal* 2009(3), pages 63–67 (cited on pages 8, 28).

Dietmair, A., A. Verl, and P. Eberspaecher (2009c). "Predictive Simulation for Model Based Energy Consumption Optimisation in Manufacturing System and Machine Control".

In: *Proc. of the 19th Flexible Automation and Intelligent Manufacturing Conference, Middlesbrough, UK*. Edited by F. Nabhani. Curran Associates Inc., pages 226–233 (cited on page 28).

Dietmair, A., A. Verl, and P. Eberspächer (2011). "Model-based energy consumption optimisation in manufacturing system and machine control". In: *International Journal of Manufacturing Research* 6(4). Inderscience Publishers, pages 380–401 (cited on page 23).

Dill, D. L. (1990). "Timing assumptions and verification of finite-state concurrent systems". In: *Automatic Verification Methods for Finite State Systems*. Volume LNCS 407. Springer, pages 197–212 (cited on page 45).

DIN-EN-ISO-50001 (2011). *Energy management systems - Requirements with guidance for use (ISO 50001:2011)* (cited on pages 6, 7).

DIN-IEC-60050 (2012). *International Electrotechnical Vocabulary*. International Electronical Commission. URL: http://www.electropedia.org/iev/iev.nsf/welcome?openform (cited on page 38).

Domschke, W. and A. Drexl (2011). *Einführung in Operations Research*. 8th edition. Springer (cited on pages 52, 53).

Draganescu, F., M. Gheorghe, and C. V. Doicin (2003). "Models of machine tool efficiency and specific consumed energy". In: *Journal of Materials Processing Technology* 141(1). Elsevier, pages 9–15 (cited on page 30).

Duflou, J. R., J. W. Sutherland, D. Dornfeld, C. Herrmann, J. Jeswiet, S. Kara, M. Hauschild, and K. Kellens (2012). "Towards energy and resource efficient manufacturing: A process and systems approach". In: *CIRP Annals - Manufacturing Technology* 61(2). Elsevier, pages 587–609 (cited on page 6).

EC (2009). *Office for Official Publications of the European Communauties, ICT and Energy Efficiency - The Case for Manufacturing*. Report. Luxembourg: European Commission (cited on page 11).

Elefterie, K.-A., M. Findeis, A. Wisdorf, J. Pottharst, and G. Pfeiffer (2012). *Milliarden einsparen dank höherer Energieeffizienz*. URL: http://www.europarl.europa.eu/pdfs/news/expert/infopress/20120907IPR50808/20120907IPR50808_de.pdf (visited on 02/13/2013) (cited on page 4).

EU-Directive (2012). *Directive 2012/27/EU of the European Parliament and of the Council, COM(2011)0370 - C7-0168/2011 - 2011/0172(COD)*. Official Journal of the European Union. URL: http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2012:315:0001:0056:EN:PDF (visited on 12/04/2012) (cited on pages 4, 149).

Eurostat. *Eurostat database*. Environment and energy – energy statistics, prices – Electricity, industrial consumers.

URL: http://epp.eurostat.ec.europa.eu/portal/page/portal/statistics/search_database (visited on 04/05/2013) (cited on page 5).

FGG (2012). *Gesetz über die friedliche Verwendung der Kernenergie und den Schutz gegen ihre Gefahren*. Bundesregierung Deutschland. URL: http://www.gesetze-im-internet.de/atg/ (visited on 11/29/2012) (cited on page 3).

Flauger, J. (2012). "Energiewende absurd". In: *Handelsblatt*(233) (November, 30 2012), pages 24–25 (cited on page 3).

Florea, A., J. Montemayor, C. Postelnicu, and J. Lastra (2012a). "A cross-layer approach to energy management in manufacturing". In: *IEEE 10th International Conference on Industrial Informatics, Beijing, China*, pages 304–308 (cited on page 19).

Florea, A., C. Postelnicu, and J. Lastra (2012b). "A Holistic View on Energy Efficiency in Manufacturing". In: *International Congress on Informatics, Environment, Energy and Applications, Singapore*. Volume 38, pages 53–57 (cited on page 20).

Floudas, C. A. and X. Lin (2004). "Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review". In: *Computers & Chemical Engineering* 28. Elsevier, pages 2109–2129 (cited on page 50).

Forrester, J. W. (1968). *Principles of Systems*. Wright-Allen Press (cited on page 38).

Fromherz, M. P. J. (2001). "Constraint-based Scheduling". In: *American Control Conference, Arlington, USA*. Volume 4, pages 3231–3244 (cited on page 98).

Fu, M. (2002). "Optimization for simulation: Theory vs. practice". In: *Journal on Computing* 14(3). INFORMS, pages 192–215 (cited on page 25).

Gadonneix, P. (2010). *Energy Efficiency: A Recipe for Success*. Report. London: World Energy Council (cited on page 3).

GCT (2010). *Innovationsallianz Green Carbody Technologies*. URL: http://www.greencarbody.de/ (visited on 02/13/2013) (cited on page 3).

George, A. (1973). "Nested dissetion of a regular finite element mesh". In: *SIAM Journal of Numerial Analysis* 2(10). Society for Industrial and Applied Mathematics, pages 345–363 (cited on page 51).

Gomory, R. E. (1958). "Outline of an Algorithm for Integer Solutions to Linear Programs". In: *Bulletin of the American Mathematical Society* 64(5), pages 275–278 (cited on page 53).

Graham, R. L., E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan (1979). "Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey". In: *Discrete Optimization II* 5. Elsevier, pages 287–326 (cited on page 48).

Greenfield, D. (2009). *2010 Global Automation Industry Outlook*. 12, pages 18–21. URL: http://electronics.wesrch.com/paper-details/pdf-EL1GP9000NODC-global-automation-survey (visited on 02/26/2013) (cited on page 4).

Günter, A. (1991). "Flexible Kontrolle in Expertensystemen zur Planung und Konfigurierung in technischen Domänen". PhD thesis. Universität Hamburg (cited on page 84).

Gutowski, T., J. Dahmus, and A. Thiriez (2006). "Electrical Energy Requirements for Manufacturing Processes". In: *13th CIRP International Conference on Life Cycle Engineering*. CIRP International (cited on page 20).

Haag, H., J. Siegert, T. Bauernhansl, and E. Westkämper (2012). "An Approach for the Planning and Optimization of Energy Consumption in Factories Considering the Peripheral Systems". In: *Leveraging Technology for a Sustainable World, Berkeley, USA*. Springer, pages 335–339 (cited on page 22).

Harel, D. (1987). "Statecharts: A Visual Formalism For Complex Systems". In: *Science of Computer Programming* 8(3). Elsevier, pages 231–274 (cited on page 60).

Harjunkoski, I. and I. E. Grossmann (2001). "A decomposition approach for the scheduling of a steel plant production". In: *Computers and Chemical Engineering* 25(11–12). Elsevier, pages 1647–1660 (cited on page 53).

Harjunkoski, I. and Grossmann. I. E. (2002). "Decomposition Techniques for Multistage Scheduling Problems Using Mixed-Integer and Constraint Programming Methods". In: *Computers & Chemical Engineering* 26(11). Elsevier, pages 1533–1552 (cited on pages 51, 98).

Heilala, J., S. Vatanen, H. Tonteri, J. Montonen, S. Lind, B. Johansson, and J. Stahre (2008). "Simulation-Based Sustainable Manufacturing System Design". In: *Proc. of the 2008 Winter Simulation Conference, Miami, USA*, pages 1922–1930 (cited on page 21).

Henzinger, T. A., X. Nicollin, J. Sifakis, and S. Yovine (1994). "Symbolic model checking for real-time systems". In: *Information and Computation* 111(2). Elsevier, pages 193–244 (cited on page 45).

Herrmann, C., L. Bergmann, S. Thiede, and A. Zein (2007). "Framework for Integrated Analysis of Production Systems". In: *Advances in Life Cycle Engineering for Sustainable Manufacturing Businesses*. Springer, pages 195–200 (cited on page 22).

Herrmann, C. and S. Thiede (2009). "Process chain simulation to foster energy efficiency in manufacturing".

In: *CIRP Journal of Manufacturing Science and Technology* 1(4). Elsevier, pages 221–229 (cited on page 21).

Herrmann, C., S. Thiede, and T. Heinemann (2011a). "A Holistic Framework for Increasing Energy and Resource Efficiency in Manufacturing". In: *Advances in Sustainable Manufacturing*. Springer, pages 265–271 (cited on page 22).

Herrmann, C., S. Thiede, S. Kara, and J. Hesselbach (2011b). "Energy oriented simulation of manufacturing systems, Concept and application". In: *CIRP Annals, Manufacturing Technology* 60. Elsevier, pages 45–48 (cited on page 22).

Hornberger, M. (2009). "Den Energieeinsatz in der Produktion optimieren". In: *Intelligenter Produzieren* 2009(5), pages 34–35 (cited on page 9).

Hübner, I. (2011). "Erkenntnisse der Profienergy-Studie". In: *openautomation* 11(3), pages 1–3 (cited on pages 7, 8, 27).

IBM (2010). *Modeling with IBM ILOG CPLEX CP Optimizer – Practical Scheduling Examples*. White paper. IBM Software Group (cited on page 98).

IEA (2004). *Energy Statistics Manual*. Report. Paris: International Energy Agency, Organisation for Economic Co-Operation and Development, EUROSTAT (cited on page 6).

IEA (2007). *Energy Policies of IEA Countries - Germany 2007 Review*. Report. Paris: International Energy Agency (cited on page 3).

IEA (2011). *Clean energy Progress Report*. Report. Paris: International Energy Agency (cited on page 3).

IEA (2012a). *Energy Management Programmes for Industry*. Report. Paris: International Energy Agency (cited on page 6).

IEA (2012b). *Energy Technology Perspectives 2012 - Pathways to a Clean Energy System*. Executive Summary. International Energy Agency (cited on page 4).

IEA (2012c). *Tracking Clean Energy Progress*. Report. Paris: International Energy Agency (cited on page 3).

IEC (2003). *Programmable controllers - Part 3: Programming languages*. Standard. URL: http://webstore.iec.ch/webstore/webstore.nsf/Artnum_PK/29664 (visited on 12/17/2012) (cited on page 94).

IEC (2011). *TC65: Industrial-process measurement, control and automation*. International Electrotechnical Commission (cited on page 32).

Inan, K. and P. Varaiya (1988). "Finitely Recursive Process Models for Discrete Event Systems". In: *IEEE Transactions on Automatic Control* 33(7), pages 626–639 (cited on page 40).

Irani, S. and K. R. Pruhs (2005). "Algorithmic problems in power management". In: *SIGACT News* 36(2). ACM, pages 63–76 (cited on page 30).

Jackson, J. R. (1955). *Scheduling a Production Line to Minimize Maximum Tardiness* (cited on page 49).

Johnson, S. M. (1954).
"Optimal two and three-stage production schedules with setup times included".
In: *Naval Research Logistics Quarterly* 1, pages 81–68 (cited on page 49).

Junge, M. (2007). "Simulationsgestützte Entwicklung und Optimierung einer energieeffizienten Produktionssteuerung".
PhD thesis. Kassel: Fachbereich Maschinenbau, Universität Kassel (cited on page 21).

Kahlenborn, W., S. Kabisch, J. Klein, I. Richter, and S. Schürmann (2012). *DIN EN 16001: Energiemanagementsysteme in der Praxis – Ein Leitfaden für Unternehmen und Organisationen*.
Report. Berlin: Bundesministerium für Umwelt, Naturschutz und Reaktorsicherheit (BMU) (cited on page 7).

Kessler, S. and S. Rakitsch (2008). *Aspects of energy efficient use of port installations*.
fml - Institute for materials handling, material flow, logistics (TU München).
URL: http://www.fml.mw.tum.de/fml/images/Publikationen/Kessler_Rakitsch%20-%20Unistock_1.pdf (visited on 04/21/2013) (cited on page 5).

Knafla, F. (2010). "Messung des Energieverbrauchs direkt an der Maschine".
In: *Energy 2.0-Kompendium*. publish-industry Verlag GmbH, München, pages 68–70.
URL: http://www.energy20.net/pi/index.php?StoryID=317&articleID=166961 (visited on 04/22/2013) (cited on page 7).

Kögler, A. (2012). "Strompreise im Fokus der Wende".
In: *Energie & Management* (July, 15 2012), page 6 (cited on page 5).

Kolbe, T. H. (2000). "Identifikation und Rekonstruktion von Gebäuden in Luftbildern mittels unscharfer Constraints".
PhD thesis. Vechta: Hochschule, Institut für Umweltwissenschaften (cited on page 84).

Kondili, E., C. Pantelides, and R. W. H. Sargent (1993).
"A general algorithm for short-term scheduling of batch operations – I. MILP formulation".
In: *Computers & Chemical Engineering* 17(2). Pergamon Press, pages 211–227 (cited on page 50).

Krägenow, T. (2012). "Kosten der Energiewende steigen auf 21,5 Milliarden Euro pro Jahr".
In: *Energie & Management* (June, 1 2012), page 5 (cited on page 5).

Kulus, D., D. Wolff, S. Ungerland, and S. Dreher (2011).
"Energieverbrauchssimulation als Werkzeug der Digitalen Fabrik".
In: *zwf-online* 106. Carl Hanser Verlag, pages 585–589 (cited on pages 8, 25).

Laroussinie, F. and K. G. Larsen (1995).
"Compositional Model Checking of Real Time Systems".
In: *Proc. of the 6th International Conference on Concurrency Theory, Philadelphia, USA*. Springer, pages 27–41 (cited on page 39).

Larsen, K., G. Behrmann, E. Brinksma, A. Fehnker, T. Hune, P. Pettersson, and J. Romijn (2001).
"As Cheap as Possible: Efficient Cost-Optimal Reachability for Priced Timed Automata".

In: *Computer Aided Verification*. Volume LNCS 2102. Springer, pages 493–505 (cited on pages 45, 47).

Larsen, K. G., P. Pettersson, and W. Yi (1995). "Model-Checking for Real-Time Systems". In: volume LNCS 965. Berlin, Heidelberg: Springer, pages 62–88 (cited on page 108).

Laub, M. (2013). "Netzwerke gegen die Stromfresser". In: *Handelsblatt*(67) (April, 8 2013) (cited on page 3).

Lawler, E. L. (1978). "Sequencing Jobs to Minimize Total Weighted Completion Time Subject to Precedence Constraints". In: *Annals of Discrete Mathematics* 2. Elsevier, pages 75–90 (cited on page 49).

Lawler, E. L., J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys (1989). *Sequencing and Scheduling: Algorithms and Complexity* (cited on page 49).

Lunze, J. (2006). *Ereignisdiskrete Systeme – Modellierung und Analyse dynamischer Systeme mit Automaten, Markovketten und Petrinetzen*. München: Oldenbourg Verlag (cited on pages 33, 37, 41).

Marsan, M. A. (1990). "Stochastic petri nets: An elementary introduction". In: *Advances in Petri Nets 1989*. Volume LNCS 424. Springer, pages 1–29 (cited on page 45).

McKinsey (2009). *Wettbewerbsfaktor Energie – Neue Chancen für die deutsche Wirtschaft*. Report. McKinsey & Company, Inc. (cited on page 4).

McKinsey (2010). *Energy Efficiency: A compelling global resource*. Report. McKinsey & Company, Inc. (cited on page 3).

Mealy, G. H. (1955). "A method for synthesizing sequential circuits". In: *Bell System Technical Journal* 34(5), pages 1045–1079 (cited on pages 42, 61).

Mechs, S., J. P. Müller, S. Lamparter, and J. Peschke (2012a). "Networked priced timed automata for energy-efficient factory automation". In: *American Control Conference, Montréal, Canada*, pages 5310–5317 (cited on page 62).

Mechs, S., S. Lamparter, and J. P. Müller (2012b). "On Evaluation of Alternative Switching Strategies for Energy-Efficient Operation of Modular Factory Automation Systems". In: *Proc. of the 17th IEEE Conference on Emerging Technologies and Factory Automation, Kraków, Poland*, pages 1–8 (cited on page 78).

Mechs, S., S. Lamparter, and J. Peschke (2012c). "Steigerung der Energieeffizienz in Automatisierungssystemen durch Start-Stopp-Automatik". In: *Automation 2012, 13. Branchentreff der Mess- und Automatisierungstechnik, Baden-Baden, Germany*. Volume VDI-Berichte 2171. Düsseldorf: VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik, pages 251–254 (cited on page 69).

Mechs, S., S. Lamparter, J. Peschke, and J. P. Müller (2013a). "Efficient Identification of Energy-Optimal Switching and Operating Sequences for Modular Factory Automation Systems". In: *26th International Conference on Industrial*

Engineering & Other Applications of Applied Intelligent Systems, Amsterdam, The Netherlands. accepted for publication (cited on page 85).

Mechs, S., S. Lamparter, J. Peschke, and J. P. Müller (2013b). "Start-Stopp-Automatik für Nicht-Produktivphasen – Höhere Energieeffizienz in Automatisierungssystemen". In: *atp edition*. accepted for publication (cited on page 91).

Méndez, C. A., J. Cerdá, I. E. Grossmann, and M. Harjunkoski I. Fahl (2006). "State-of-the-art review of optimization methods for short-term scheduling of batch processes". In: *Computers & Chemical Engineering* 30. Elsevier, pages 913–946 (cited on pages 50, 98).

Merlin, P. M. (1974). "A study of the recoverability of computing systems". PhD thesis. Irvine: Department of Information and Computer Science, University of California (cited on page 44).

Migliore, M., V. Martorana, and F. Sciortino (1990). "An algorithm to find all paths between two nodes in a graph". In: *Journal of Computational Physics* 87(1). Academic Press Professional, San Diego, pages 231–236 (cited on page 70).

Mittelbach, K. (2010). *Energie intelligent erzeugen, verteilen und nutzen*. Report. Version 2. vollständig überarbeitete Auflage. Weißbuch Energie-Intelligenz. Zentralverband Elektrotechnik- und Elektroindustrie e. V. (ZVEI) (cited on page 13).

Mitten, L. G. (1970). "Branch-And-Bound Methods: General Formulation and Properties". In: *Operations Research* 18(1). INFORMS, pages 24–34 (cited on page 51).

Möhring, R. H. and F. J. Radermacher (1984). "Substitution Decomposition for Discrete Structures and Connections with Combinatorial Optimization". In: *North-Holland Mathematics Studies* 95. Elsevier, pages 257–356 (cited on page 51).

Molloy, M. K. (1981). "On the Integration of Delay and Throughput Measures in Distributed Processing Models". PhD thesis. UCLA (cited on page 44).

Montanari, U. (1971). *Networks of Constraints: Fundamental Properties and Applications to Picture Processing*. Computer Science Department, Carnegie Mellon University, pages 1–54 (cited on page 83).

Moore, E. F. (1956). "Gedanken-Experiments on sequential machines". In: *Automata Studies*. Edited by C. Shannon and J. McCarthy. Princeton University Press, pages 129–153 (cited on pages 42, 61).

Mouret, S., I. E. Grossmann, and P. Pestiaux (2011). "Time representations and mathematical models for process scheduling problems". In: *Computers & Chemical Engineering* 35(6). Elsevier, pages 1038–1063 (cited on page 50).

Mouzon, G. and M. B. Yildirim (2008). "A framework to minimise total energy consumption and total tardiness on a single machine". In: *International Journal of Sustainable Engineering* 1(2), pages 105–116 (cited on page 28).

Müller, A. (2012). "Virtuelles Kraftwerk auf dem Weg". In: *Energie & Management* (June, 1 2012). Energie & Management Verlagsgesellschaft mbH, Herrsching, Germany, page 18 (cited on pages 8, 19).

Müller, E., J. Engelmann, T. Löffler, and J. Strauch (2009). *Energieeffiziente Fabriken planen und betreiben*. Heidelberg: Springer (cited on pages 4, 7, 22).

Müller, E. and T. Löffler (2010). *Energy Efficiency at Manufacturing Plants - A Planning Approach*, pages 1–8 (cited on page 22).

Müller-Merbach, H. (1973). *Operations Research*. 3rd edition. Franz Vahlen (cited on page 53).

Natkin, S. (1980). "Les Réseaux de Petri Stochastiques et leur Application à l'Evaluation des Systèmes Informatiques". PhD thesis. Paris: CNAM (cited on page 44).

Neher, J. (2009). "Energiedatenerfassung in der Produktion - Technologien und Möglichkeiten". In: *Fraunhofer IPA Workshop, Stuttgart, Germany*. Edited by Fraunhofer IPA. F 182 (cited on pages 9, 10).

Neugebauer, R. (2008). *Energieeffizienz in der Produktion – Untersuchung zum Handlungs- und Forschungsbedarf*. Report. Berlin: Bundesministerium für Bildung und Forschung (cited on pages 8, 9, 11).

Newman, S. T., A. Nassehi, R. Imani-Asrai, and V. Dhokia (2012). "Energy efficient process planning for CNC machining". In: *CIRP Journal of Manufacturing Science and Technology* 5(2). Elsevier, pages 127–136 (cited on page 28).

Niebert, P., S. Tripakis, and S. Yovine (2000). "Minimum-Time Reachability for Timed Automata". In: *IEEE Mediteranean Control Conference, Lisbon, Portugal* (cited on page 46).

Niemann, K.-H. (2012). "Energiemanagement in Automatisierungssystemen – Stand der Technik und künftige Anforderungen". In: *AUTOMATION 2012, 13. Branchentreff der Mess- und Automatisierungstechnik, Baden-Baden, Germany*. Edited by VDI Wissensforum GmbH. Düsseldorf, pages 99–102 (cited on page 6).

Nolde, K. and M. Morari (2010). "Electrical load tracking scheduling of a steel plant". In: *Computers and Chemical Engineering* 34(11). Elsevier, pages 1899–1903 (cited on page 22).

NYT (May 2012). *Merkel Pays a Price for Her Energy Policy Shift*. New York Times. URL: http://www.nytimes.com/2012/05/29/world/europe/29iht-letter29.html (visited on 11/29/2012) (cited on page 3).

OMG (2010). *OMG Unified Modeling Language, Infrastructure, Version 2.3*. URL: http://www.omg.org/spec/UML/2.3/ (visited on 04/22/2013) (cited on page 29).

Pahl, G., W. Beitz, J. Feldhusen, and K.-H. Grote (2007). *Konstruktionslehre - Grundlagen*. 7th edition. Berlin, Heidelberg, New York: Springer (cited on page 35).

Pantelides, C. C. (1994).
"Unified Frameworks for the Optimal Process Planning and Scheduling". In: *Proceedings of the Second Conference on Foundations of Computer Aided Operations, Austin, USA*. Edited by D.W.T. Rippin and J. Hale, pages 253–274 (cited on page 20).

Papadimitriou, C. H. and K. Steiglitz (1982).
Combinatorial Optimization: Algorithms and Complexity. Englewood Cliffs (cited on page 48).

Petri, C. A. (1962). "Kommunikation mit Automaten".
PhD thesis. Institut für Instrumentelle Mathematik (cited on page 44).

Pinto, J. M. and I. E. Grossmann (1998).
"Assignment and sequencing models for the scheduling of process systems".
In: *Annals of Operations Research* 81(0). Kluwer Academic Publishers, pages 433–466 (cited on page 48).

PLCSim (2012). *SIMATIC S7-PLCSIM – Software test without controller, Siemens AG*.
URL: www.automation.siemens.com/mcms/simatic-controller-software/en/step7/simatic-s7-plcsim/Pages/Default.aspx (visited on 12/30/2012) (cited on page 113).

PNO (2010). *THE PROFIenergy PROFILE – Increasing the Energy Efficiency of Automation Systems using Smart Energy Management over PROFINET*. Report.
Karlsruhe: PROFIBUS Nutzerorganisation e. V. (PNO) (cited on page 27).

PNO (2011). *Assessing PROFIenergy's potential – Quantifying the energy saving possibilities of PI's PROFIenergy profile for PROFINET and assessing its deployment opportunities*. Report.
Karlsruhe: PROFIBUS Nutzerorganisation e. V. (PNO) (cited on page 27).

Putz, M., A. Schlegel, S. Lorenz, E. Franz, and S. Schulz (2011).
"Gekoppelte Simulation von Material- und Energieflüssen in der Automobilfertigung".
In: *14. Tage des Betriebs- und Systemingenieurs am 24. November 2011*.
Wissenschaftliche Schriftenreihe des Instituts für Betriebswissenschaften und Fabriksysteme 17, TU Chemnitz, pages 135–144 (cited on page 23).

QuartzNET (2012). *Quartz.NET – Enterprise Job Scheduler for .NET Platform*. Terracotta, USA.
URL: http://quartznet.sourceforge.net/ (visited on 12/30/2012) (cited on page 101).

Rager, M. (2006).
"Energieorientierte Produktionsplanung – Analyse, Konzeption und Umsetzung".
PhD thesis. Universität Augsburg (cited on page 21).

Rahimifard, S., Y. Seow, and T. Childs (2010).
"Minimising Embodied Product Energy to support energy efficient manufacturing".
In: *CIRP Annals - Manufacturing Technology* 59(1). Elsevier, pages 25–28 (cited on page 22).

Ramchandani, C. (1974). "Analysis of asynchronous concurrent systems by timed Petri nets".
PhD thesis. Cambridge: Massachusetts Institute of Technology (cited on page 44).

Rasmussen, J., K. G. Larsen, and K. Subramani (2004).
"Resource optimal scheduling using priced timed automata".

In: *Tools and Algorithms for the Construction and Analysis of Systems*. Volume LNCS 2988. Springer, pages 220–235 (cited on page 29).

Reinhart, G., F. Geiger, F. Karl, and M. Weidemann (2011). "Handlungsfelder zur Realisierung energieeffizienter Produktionsplanung- und steuerung". In: *Zeitschrift für wirtschaftlichen Fabrikbetrieb* 106. Carl Hanser Verlag, München, pages 596–600 (cited on pages 20, 25).

Reklaitis, G. V. (2000). "Overview of Planning and Scheduling Technologies". In: *Latin American Applied Research* 30, pages 285–293 (cited on page 48).

ResCom (2012). *Resource Conservation by Context-Activated M2M-Communication*. URL: http://www.res-com-projekt.de/index.php/home_EN.html (visited on 02/13/2013) (cited on page 3).

Rinaudo, S., G. Gangemi, A. Calimera, A. Macii, and M. Poncino (2011). "Moving to Green ICT: From stand-alone power-aware IC design to an integrated approach to energy efficient design for heterogeneous electronic systems". In: *Design, Automation and Test in Europe*, pages 1–2 (cited on page 31).

Rong, P. and M. Pedram (2006). "Determining the optimal timeout values for a power-managed system based on the theory of Markovian processes: offline and online algorithms". In: *Design, Automation and Test in Europe, Leuven, Belgium*. European Design and Automation Association, pages 1128–1133 (cited on page 30).

Salem, A., C. Anagnostopoulos, and G. Rabadi (2000). "A branch-and-bound algorithm for parallel machine scheduling problems". In: *Harbour, Maritime & Multimodal Logistics Modeling and Simulation Workshop, Portofino, Italy*. Society for Computer Simulation International (SCS) (cited on page 52).

Santos, J. P., Marta Oliveira, F. G. Almeida, J. P. Pereira, and A. Reis (2011). "Improving the environmental performance of machine-tools: influence of technology and throughput on the electrical energy consumption of a press-brake". In: *Journal of Cleaner Production* 19(4). Elsevier, pages 356–364 (cited on pages 9, 10).

Schermer, G. (2012). "Die Stromnachfrage aktiv managen". In: *Energie & Management* 19 (October, 1 2012). Energie & Management Verlagsgesellschaft mbH, Herrsching, Germany, page 10 (cited on page 8).

Schlechtendahl, J., P. Eberspächer, H. Haag, A. Verl, and E. Westkämper (2013). "Framework for Controlling Energy Consumption of Machine Tools". In: *EcoProduction and Logistics*. Edited by P. Golinska. 2013th edition. Berlin, Heidelberg: Springer, pages 155–168 (cited on page 27).

Schnellnhuber, H. J., D. Messner, C. Leggewie, R. Leinfelder, N. Nakicenovic, S. Rahmsdorf, S. Schlacke, J. Schmid, and R. Schubert (Apr. 2012). *Welt im Wandel – Gesellschaftsvertrag für eine Grosse Transformation*.

URL: http://www.wbgu.de/fileadmin/templates/dateien/veroeffentlichungen/hauptgutachten/jg2011/wbgu_jg2011_ZfE.pdf (visited on 04/22/2013) (cited on page 3).

Segala, R. (1995). "Modeling and verification of randomized distributed real-time systems". PhD thesis (cited on page 44).

Seow, Y. and S. Rahimifard (2011). "A framework for modelling energy consumption within manufacturing systems". In: *CIRP Journal of Manufacturing Science and Technology* 4(3). Elsevier, pages 258–264 (cited on page 22).

Shanon, C. E. (1938). "A symbolic analysis of relay and switching circuits". In: *Transactions of the AIEE* 57(12), pages 713–723 (cited on pages 25, 40).

Shorin, D. and A. Zimmermann (2010). "Model-based Development of Energy-Efficient Automation Systems". In: *55th IWK, International Scientific Colloquium, Ilmenau, Germany*. Edited by Rector of the Ilmenau University of Technology. URL: http://www.tu-ilmenau.de/en/sse/forschung/projekte/mdeas/ (visited on 02/13/2013) (cited on page 29).

Shorin, D., A. Zimmermann, and P. Paciel (2012). "Transforming UML State Machines into Stochastic Petri Nets for Energy Consumption Estimation of Embedded Devices". In: *2nd IFIP Conference on Sustainable Internet and ICT for Sustainability, Pisa, Italy*, pages 1–6 (cited on page 29).

SiemensAG (2011). *SINUMERIK Ctrl-Energy - Maximum energy efficiency for your machine tools*. Siemens AG Industry Sector, Motion Control Systems. URL: https://c4b.gss.siemens.com/resources/articles/e20001-a1440-p610-x-7600.pdf (visited on 02/13/2013) (cited on page 10).

Šimunić, T., L. Benini, P. Glynn, and G. De Micheli (2001). "Event-Driven Power Management". In: *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems* 20(7), pages 840–857 (cited on page 30).

Srivastava, M. B., A. P. Chandrakasan, and R. B. Brodersen (1996). "Predictive System Shutdown and Other Architectural Techniques for Energy Efficient Programmable Computation". In: *IEEE Transactions on Very Large Scale Integration Systems* 4(1), pages 42–55 (cited on page 30).

Swaminathan, V. and K. Chakrabarty (2003). "Energy-Conscious, Deterministic I/O Device Scheduling in Hard Real-Time Systems". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 22(7), pages 847–858 (cited on page 30).

Thiede, S. (2012). *Energy Efficiency in Manufacturing Systems*. Edited by C. Herrmann. Springer. Chapter State of research, pages 51–88 (cited on page 25).

Thorelli, L.-E. (1966). "An algorithm for computing all paths in a graph". In: *BIT Numerical Mathematics* 6(4). Kluwer Academic Publishers, pages 347–349 (cited on page 70).

Tuttle, M. R. (1984). "Hierarchical correctness proofs for distributed algorithms". supervisor: Lynch, N. A. Master's thesis. Massachusetts Institute of Technology (cited on page 40).

Ullman, J. D. (1975). "NP-complete scheduling problems". In: *Journal of Computer and System Sciences* 10(3). Academic Press, pages 384–393 (cited on page 49).

VDI-3633 (1996). *Simulation von Logistik-, Materialfluß- und Produktionssystemen - Begriffsdefinitionen* (cited on page 133).

VDI-4602 (2007). *Energy Management* (cited on page 6).

Vijayaraghavan, A. and D. Dornfeld (2010). "Automated energy monitoring of machine tools". In: *CIRP Annals - Manufacturing Technology* 59(1). Elsevier, pages 21–24 (cited on page 26).

Waez, T. B., J. Dingel, and K. Rudie (2011). *Timed Automata for the Development of Real-Time Systems*. Report. School of Computing, Department of Electrical and Computer Engineering, Queen's University, ON (cited on page 42).

Wagner, H.-F. (Feb. 2010). *Energieforschungsprogramme 1974 bis heute*. Welt der Physik. URL: http://www.weltderphysik.de/gebiete/technik/unsortiert/energie/geschichte-der-energieforschung/energieforschung-ab-1974/ (visited on 04/22/2013) (cited on page 3).

Watts, D. J. and H. Strogatz (1998). "Collective dynamics of small-word networks". In: *nature* 393 (June, 4 1998). Nature Publishing Group, pages 440–442 (cited on page 117).

Weck, M. and C. Brecher (2005). *Werkzeugmaschinen - Maschinenarten und Anwendungsbereiche*. 6th edition. Berlin, Heidelberg: Springer (cited on pages 34, 35).

Weinert, N. (2010). "Vorgehensweise für Planung und Betrieb energieeffizienter Produktionssysteme". PhD thesis. TU Berlin (cited on page 26).

Wellenreuther, G. and D. Zastrow (2008). *Automatisieren mit SPS – Theorie und Praxis*. Vieweg + Teubner (cited on page 94).

Westerkamp, D. (2009). *Automation 2020 - Bedeutung und Entwicklung der Automation bis zum Jahr 2020 - Thesen und Handlungsfelder*. Report. Version 2. Düsseldorf: VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik (cited on page 7).

Wettmann, R. W. (Aug. 2012). *Le tournant énergétique en Allemagne - Un projet en crise?* URL: http://library.fes.de/pdf-files/bueros/paris/09368.pdf (visited on 04/22/2013) (cited on page 3).

Wolff, D., D. Kulus, and S. Dreher (2012).
"Simulating Energy Consumption in Automotive Industries". In:
Use Cases of Discrete Event Simulation. Springer, pages 59–86 (cited on page 24).

Yang, L., J. Deuse, and M. Droste (2011).
"Energy Efficiency at Energy Intensive Factory – A Facility Planning Approach".
In: *IEEE 18th International Conference on Industrial Engineering and Engineering Management, Changchun, China*. Volume 1, pages 699–703 (cited on page 24).

Yildirim, M. B. and G. Mouzon (2012).
"Single-Machine Sustainable Production Planning to Minimize Total Energy Consumption and Total Completion Time Using a Multiple Objective Genetic Algorithm".
In: *IEEE Transaction on Engineering Management* 59(4), pages 585–597 (cited on page 28).

Zeigler, B. P. (1976). *Theory of modelling and simulation*. USA: Elsevier Science (cited on page 40).

Zeigler, B. P., H. Preahofer, and T. G. Kim (2000). *Theory of modelling and simulation*.
2nd edition. San Diego: Academic Press (cited on page 40).

Zeigler, B. P., H. Praehofer, and T. G. Kim (2005). *Theory of Modeling and Simulation - Integrating discrete event and continuous complex dynamic systems*. 2nd edition.
Amsterdam: Academic Press (cited on page 38).

Zimmermann, A. and M. Knoke (2007). *TimeNET 4.0: A Software Tool for the Performability Evaluation with Stochastic and Colored Petri Nets*.
URL: http://www.tu-ilmenau.de/sse/timenet/ (visited on 04/22/2013) (cited on page 29).

Index

S

T

U

V

W

Appendix A

Test Bed $\mathbf{tb}_S$

A.1 System model of Test Bed $\mathbf{tb}_S$

The automaton-based system model of Test Bed tb_S is presented in Figure A.1.

A.2 Meaning of operating modes

The meaning of operating modes in Test Bed tb_S is explained in Table A.1. It is distinguished between ε (electric off), β for working (available), γ_i (partially available), and α (productive mode) of the components.

Component	**Modes of the system** ε	γ_1	γ_2	γ_3	β	α
CU	ε	β	β	β	β	α
I/O	ε	β	β	β	β	α
P1.1	ε	ε	β	β	β	α
P1.2	ε	ε	γ	β	β	α
P3.1	ε	ε	ε	ε	β	α
P3.2	ε	ε	ε	ε	β	α
P2.1	ε	ε	ε	ε	β	α
P2.2	ε	ε	ε	ε	β	α

Table A.1: Modes of the system and their reference to the components of Test Bed tb_S

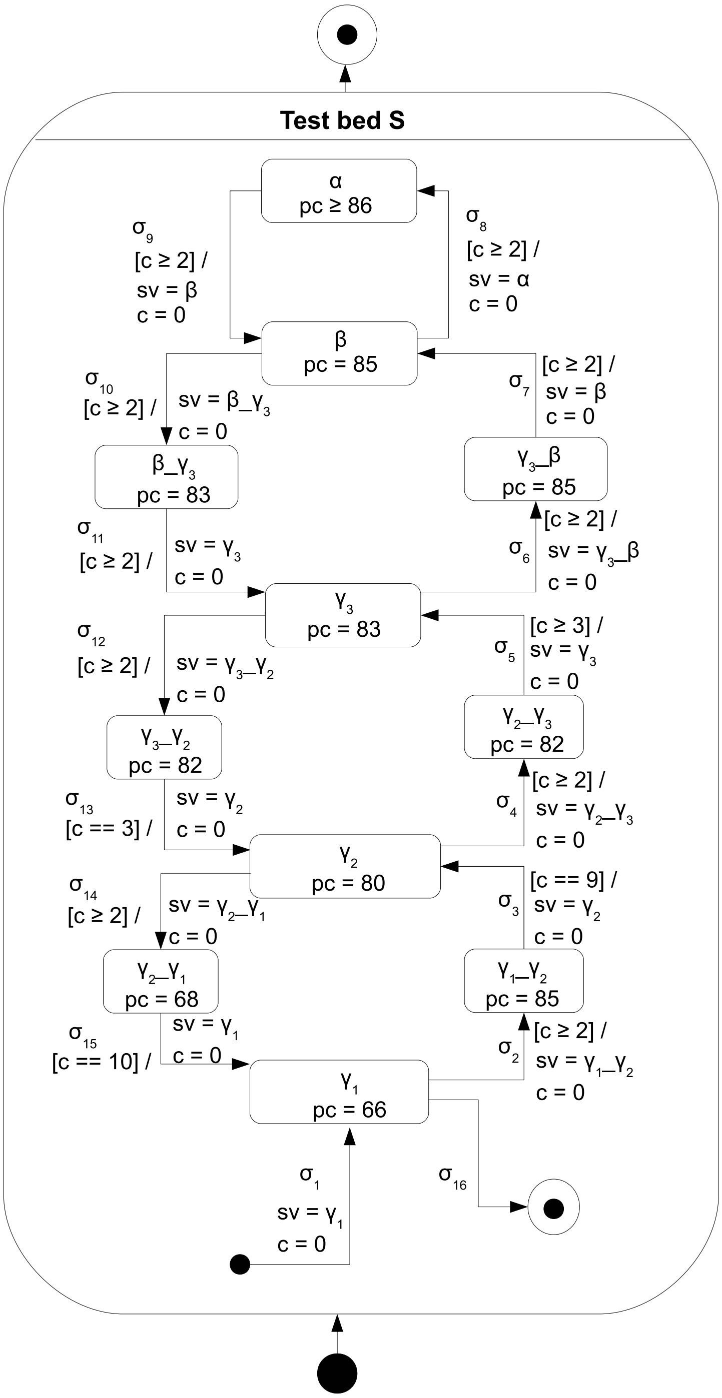


Figure A.1: Automaton-based system model of Test Bed tb_S

Appendix B

Test Bed tb_M

B.1 Subsystems of the test bed

B.1.1 Subtasks

Test Bed tb_M consists of nine subsystems with specific automation subtask (Tab. B.1).

Subsystem ID	Automation subtask
1	Filling bottles
2	Capping bottles
3	Uncapping bottles
4	Discharging/recycling bottles
5	Picking bottles
6	Commissioning bottles
7	Checking the content of bottles
8	Checking the content of bottles
9	Transporting bottles

Table B.1: Test bed subsystems and automation subtasks

B.1.2 Models of the subsystems

The timed networked automation subsystems of Test Bed tb_M are illustrated in the following Figures B.1, B.2, B.3, B.4, B.5, B.6, B.7, B.8, B.9. The automation subsystems are orthogonal to each other.

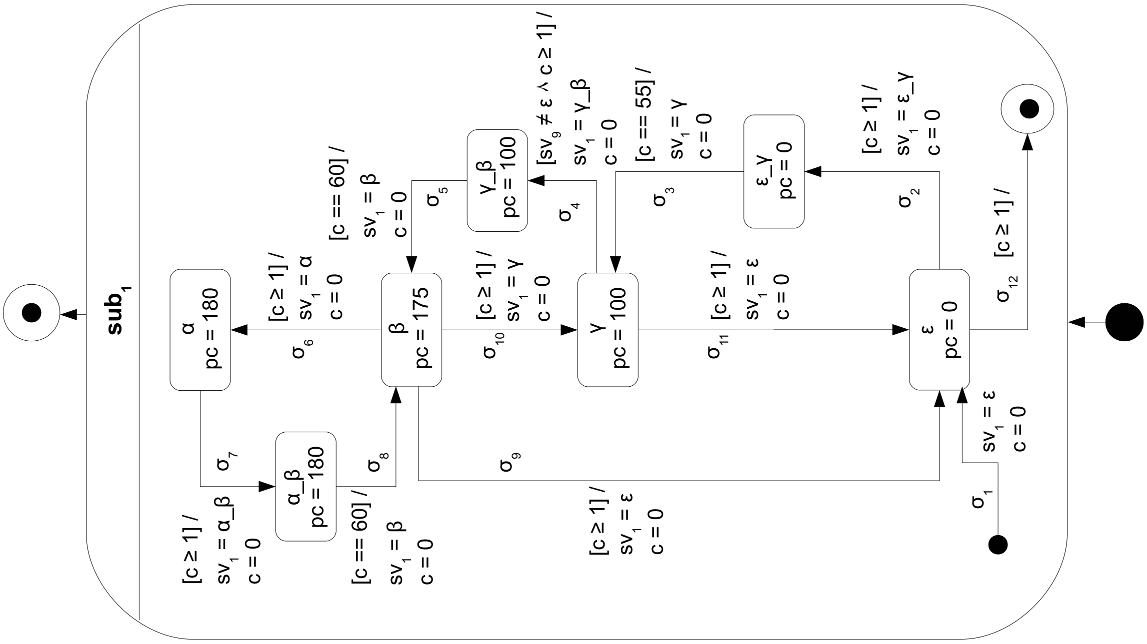


Figure B.1: Timed networked automation subsystem sub_1

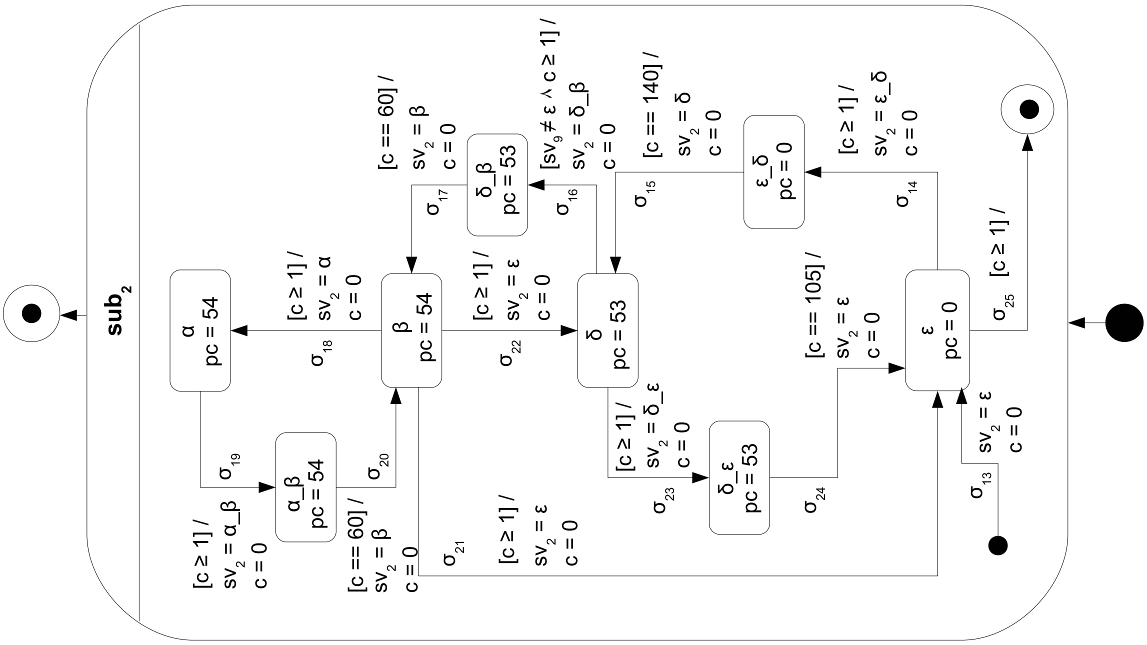


Figure B.2: Timed networked automation subsystem sub_2

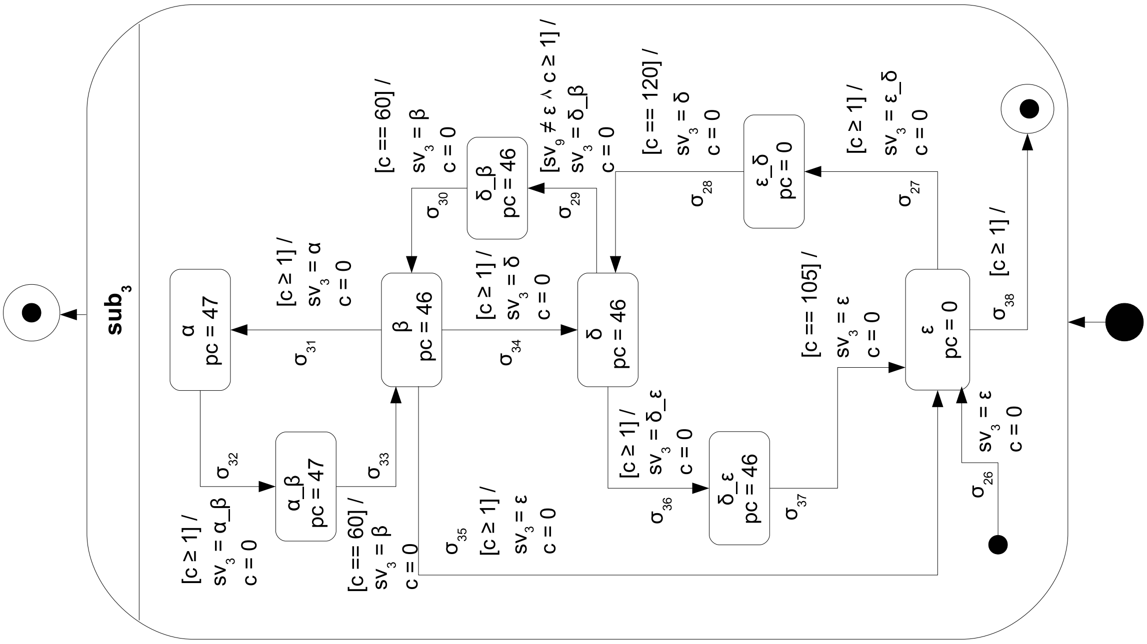


Figure B.3: Timed networked automation subsystem sub_3

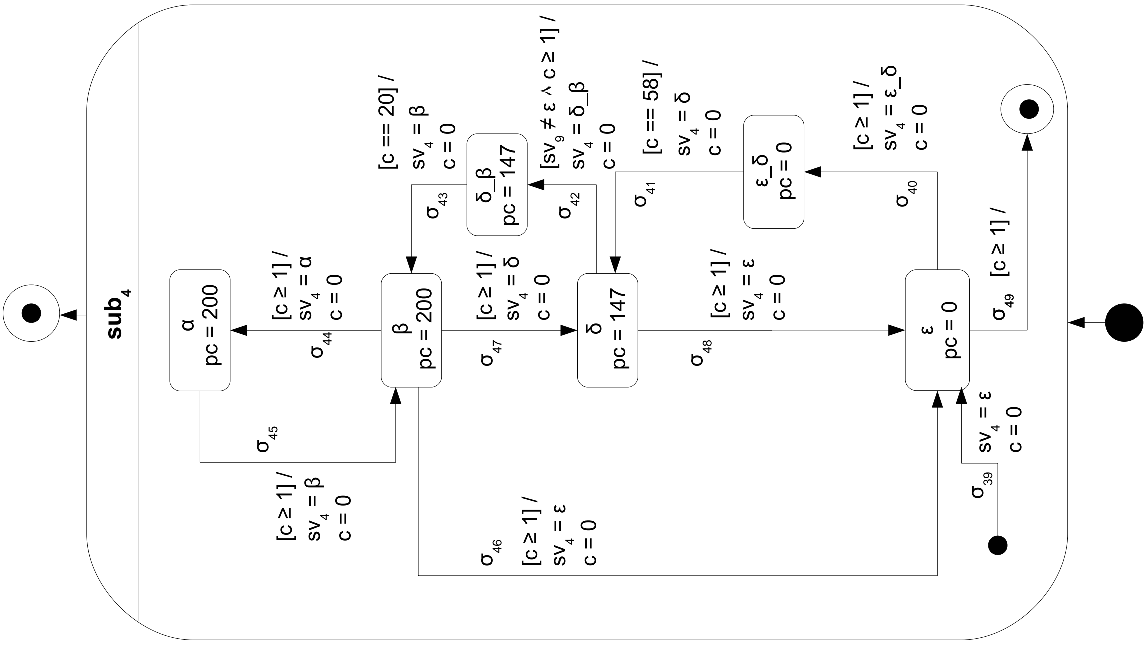


Figure B.4: Timed networked automation subsystem sub_4

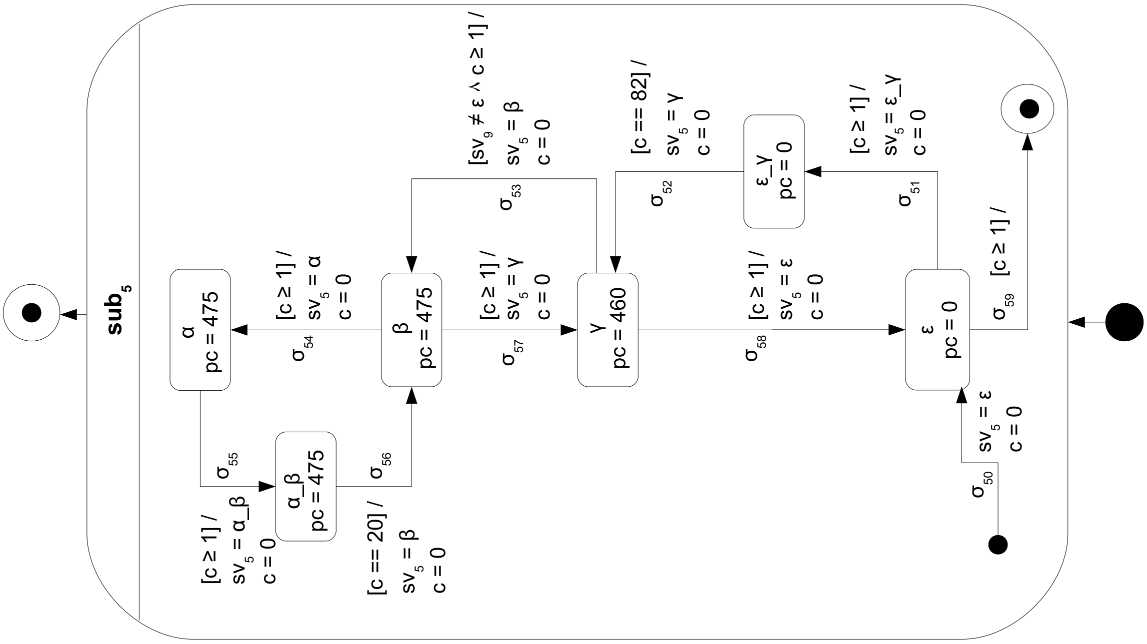


Figure B.5: Timed networked automation subsystem sub_5

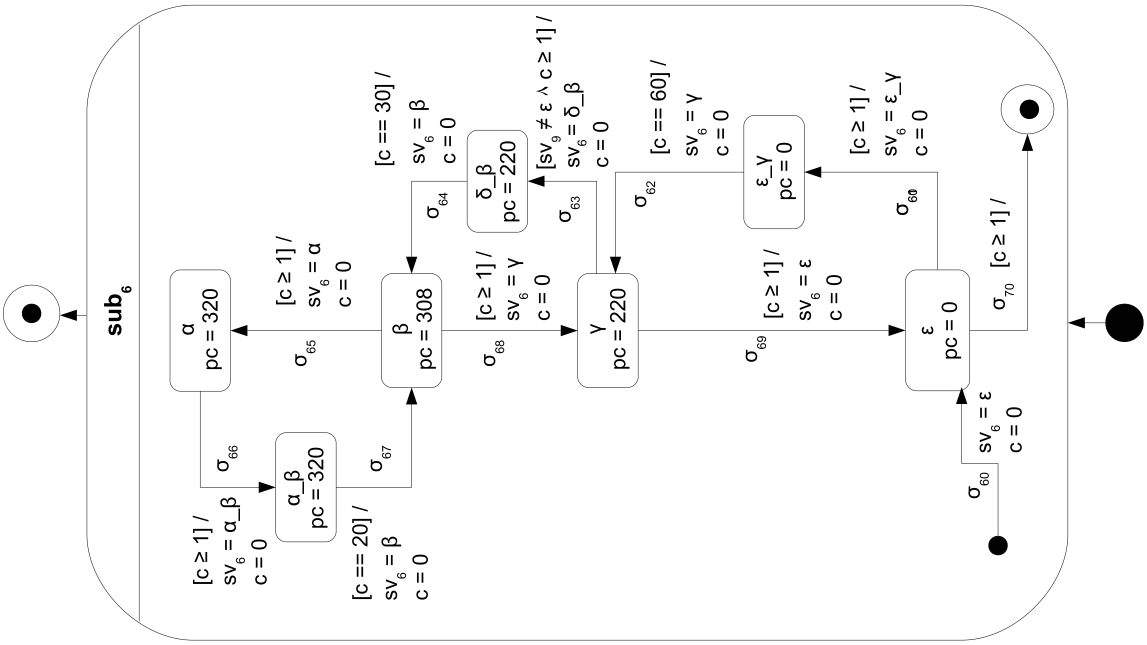


Figure B.6: Timed networked automation subsystem sub_6

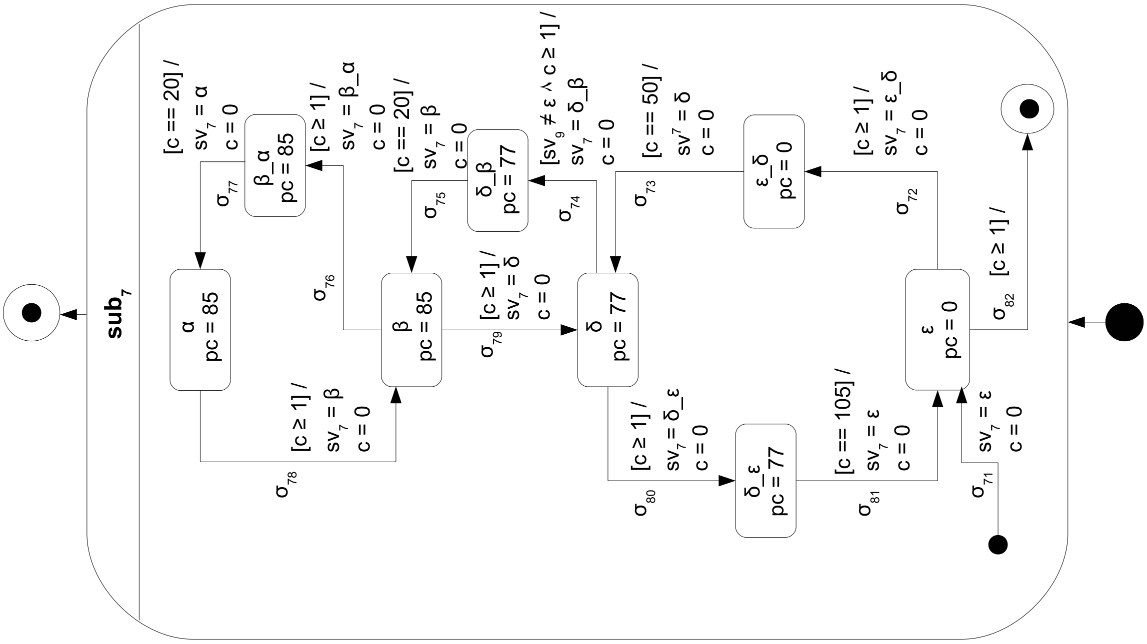


Figure B.7: Timed networked automation subsystem sub_7

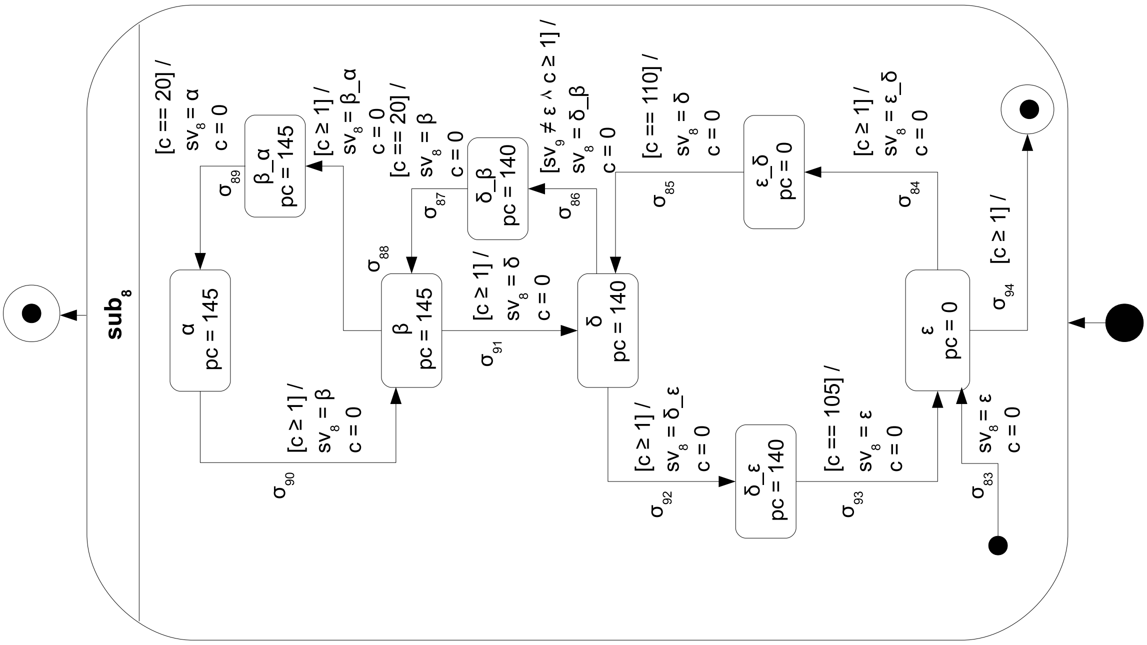


Figure B.8: Timed networked automation subsystem sub_8

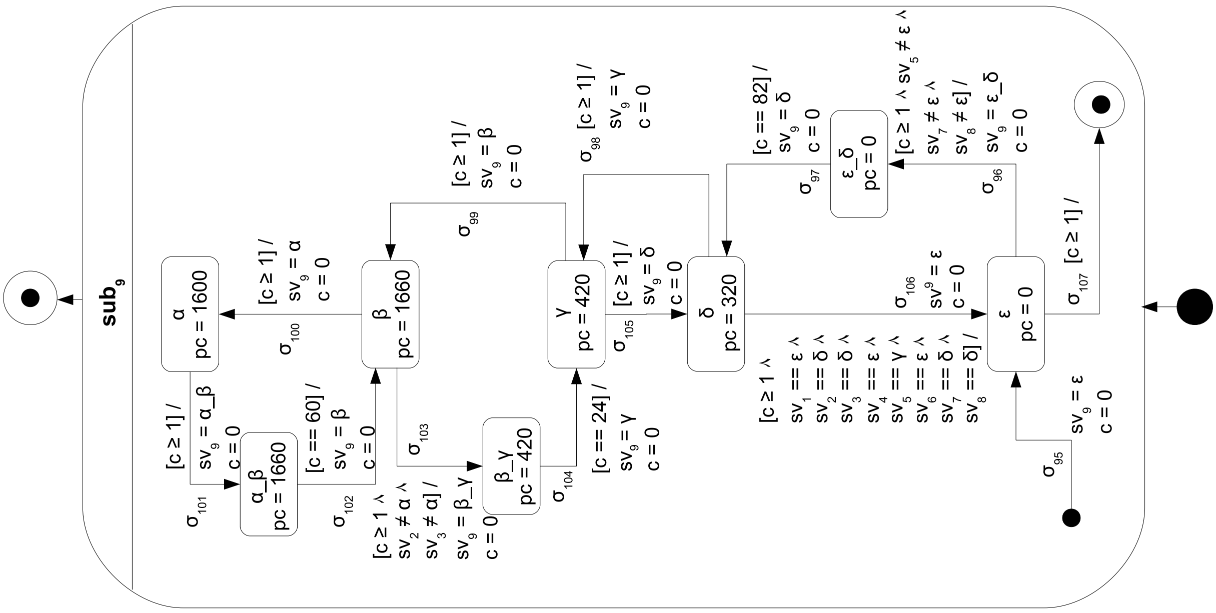


Figure B.9: Timed networked automation subsystem sub_9

B.2 Input power of Test Bed tb_M

The input power of Test Bed tb_M is measured for different operating modes of its nine subsystems (Fig. B.10). Each subsystem has a specific set of operating modes (α, β, γ_i, δ, ε) with specific input power.

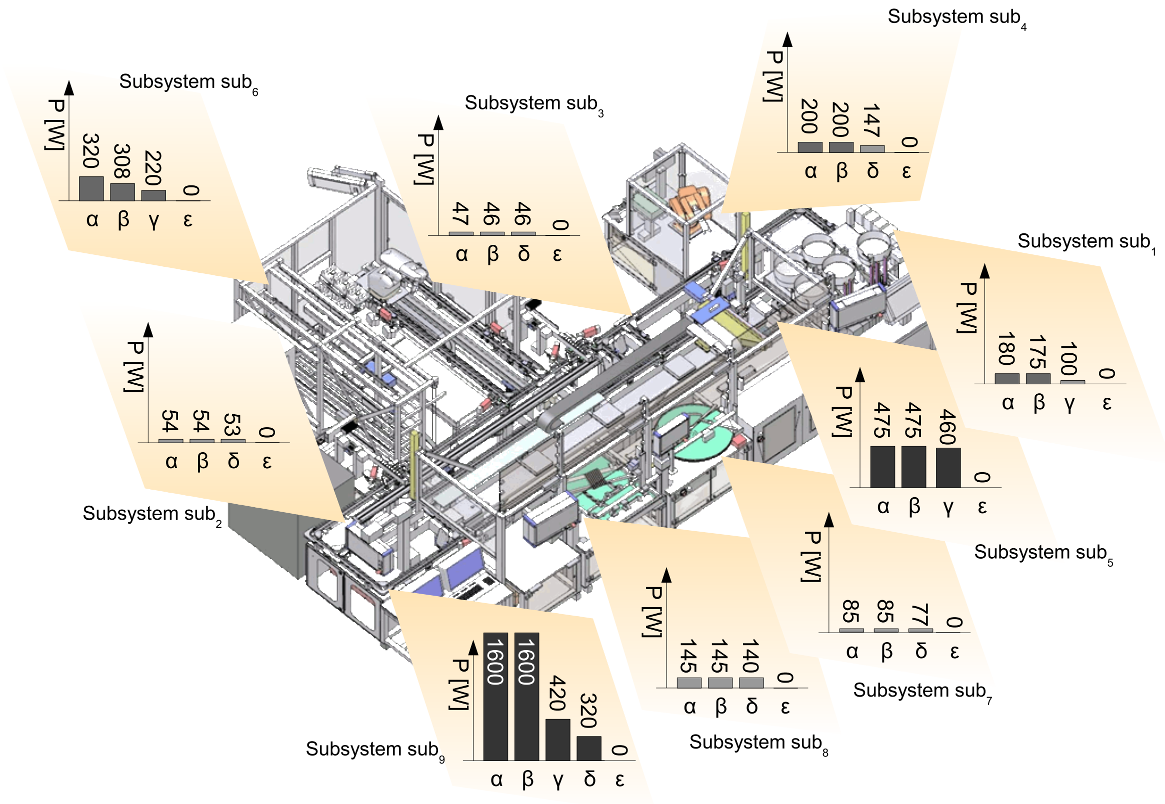


Figure B.10: Measured input power (apparent power) for different operating modes of the test bed

B.3 Idling of Test Bed tb_M

Idling of the test bed means a specific combination of operating modes in the automation subsystems. The test bed has a 54,6% idling input power of the input power during full load operation (Tab. B.2 and Tab. B.3).

Subsystem ID	Operating mode	Symbol	Input power, idling [W]
1 (Filling)	stand-by	γ_i	100
2 (Capping)	ready-for-production	β	54
3 (Uncapping)	ready-for-production	β	46
4 (Discharging)	ready-for-production	β	200
5 (Picking)	stand-by	γ_i	460
6 (Commissioning)	stand-by	γ_i	220
7 (Checking 1)	ready-for-production	β	85
8 (Checking 2)	ready-for-production	β	145
9 (Transportation)	stand-by	γ_i	420

Table B.2: Modes of subsystems while test bed is idling

Subsystem ID	Operating mode	Symbol	Input power, full load [W]
1 (Filling)	production	α	180
2 (Capping)	production	α	54
3 (Uncapping)	production	α	47
4 (Discharging)	production	α	200
5 (Picking)	production	α	475
6 (Commissioning)	production	α	320
7 (Checking 1)	production	α	85
8 (Checking 2)	production	α	145
9 (Transportation)	production	α	1660

Table B.3: Modes of subsystems while test bed is in full load operation

Appendix C

Evaluation of the approach

C.1 Identification of optimal strategies

Figure C.1 shows a subsystem with 10 modes and 12 transitions. Figure C.2 presents a subsystem with 19 modes and 24 transitions.

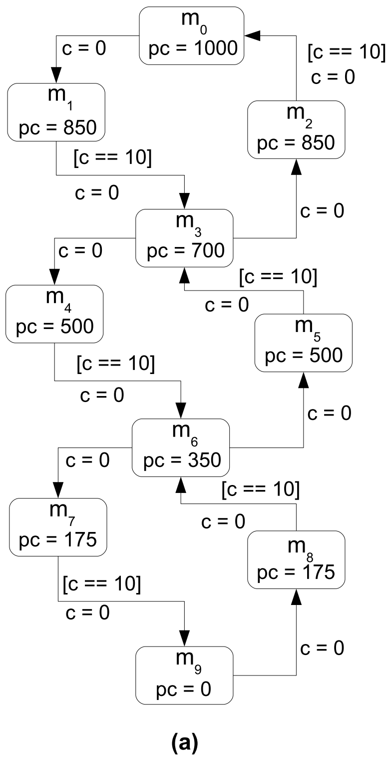


Figure C.1: Subsystem with 10 modes and 12 transitions

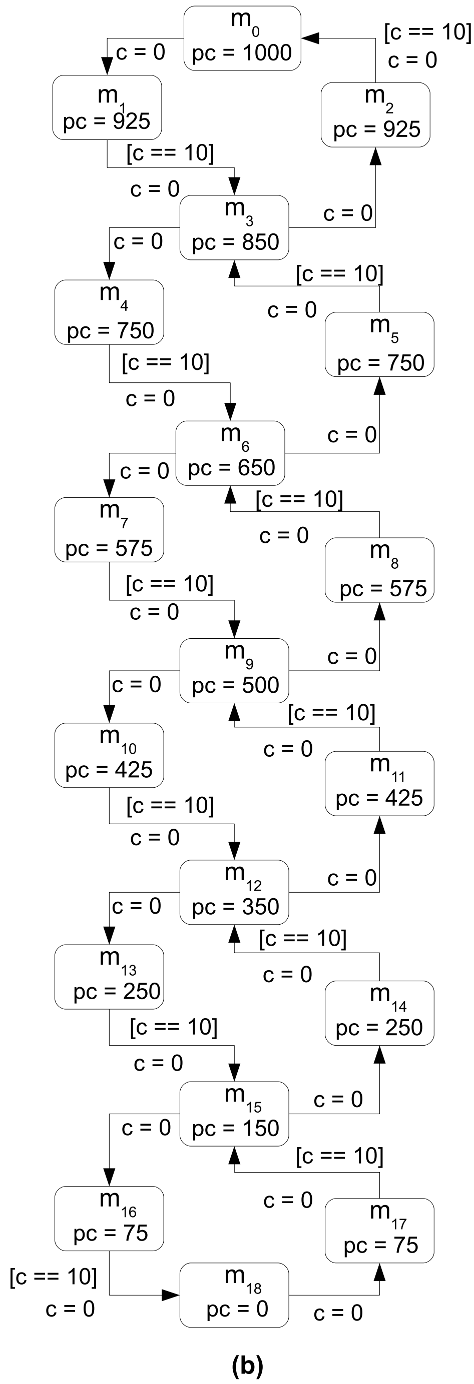


Figure C.2: Subsystem with 19 modes and 24 transitions

C.2 Specification of feasible strategies

Schedules for planned and actual strategies (Scenarios Ver.tbM.1, Ver.tbM.5, Ver.tbM.7) are illustrated as strategies in Figures C.3, C.4, and C.5.

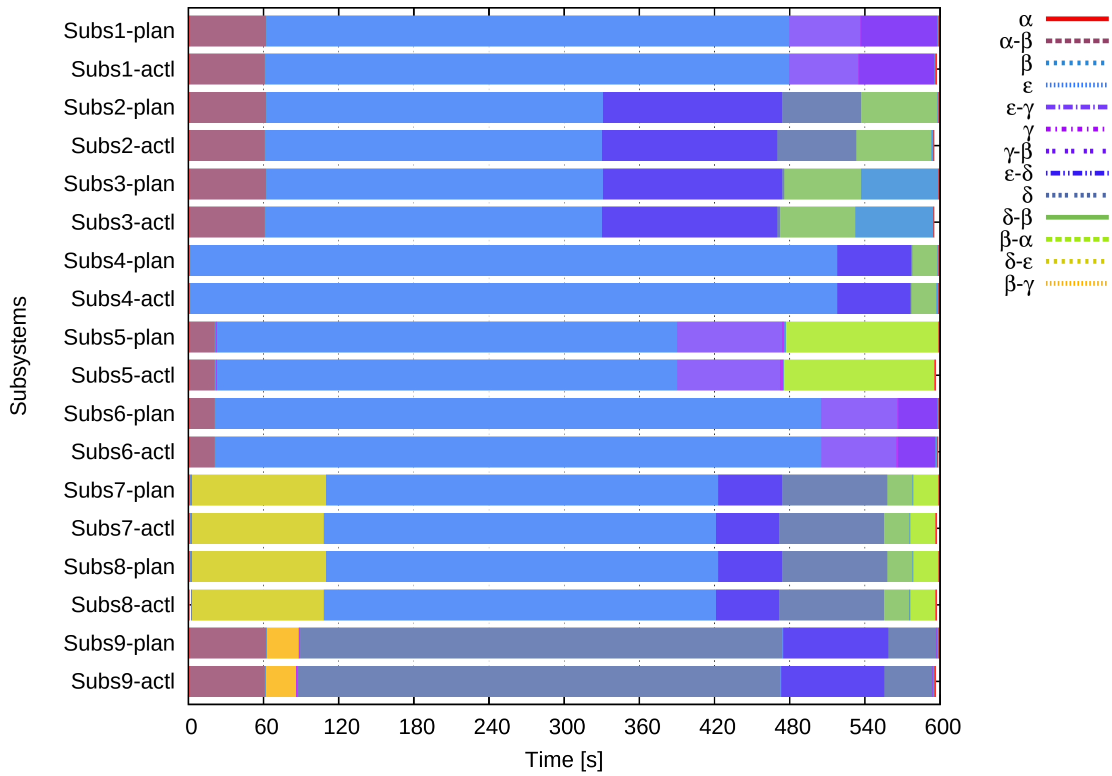


Figure C.3: *Ver.tbM.1*: Planned and executed strategy ($\text{dev}_{\text{delay}}$ = +0,4%)

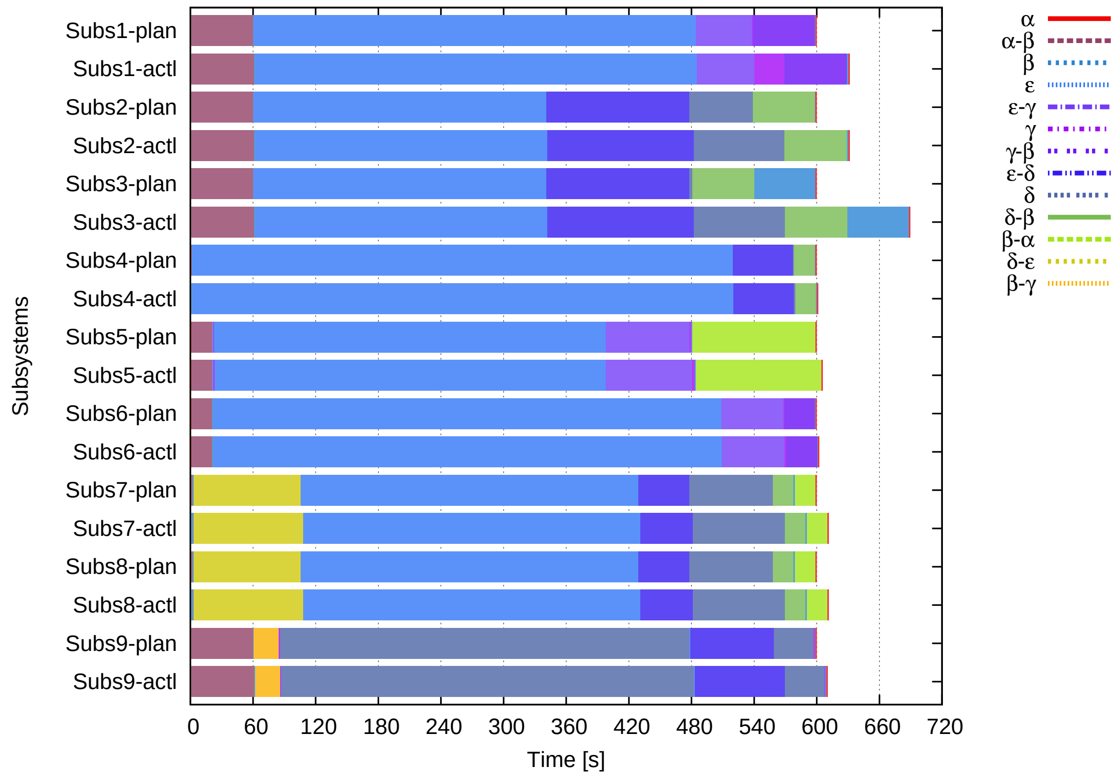


Figure C.4: *Ver.tbM.5*: Plan and executed strategy (dev$_{\text{delay}}$ = -3,4%)

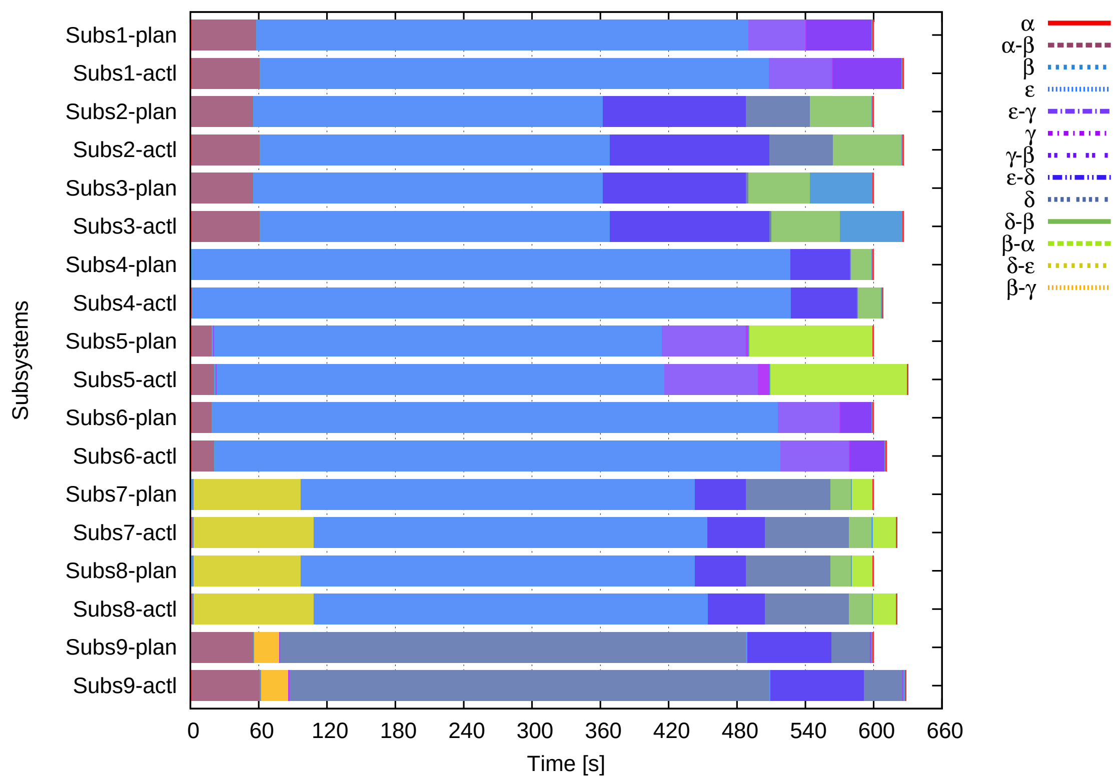


Figure C.5: *Ver.tbM.7*: Plan and executed strategy (dev$_{\text{delay}}$ = -3,6%)

C.3 Model validation using strategies

C.3.1 Accuracy of energy demand prediction for Test Bed tb_S

Planned strategies are executed in Test Bed tb_S to check if the predicted energy demand corresponds to the actual energy demand after executing the strategy. In Figure C.6, Test Bed tb_S is initially in mode β and the test bed returns to mode β at the end of the pause interval (10 [s]). The planned strategy has an energy demand of 840 [J] whereas the actual strategy has an energy demand of 939 [J]. This results in an underestimation of the actual energy demand of dev_{energy} = -10,5 %

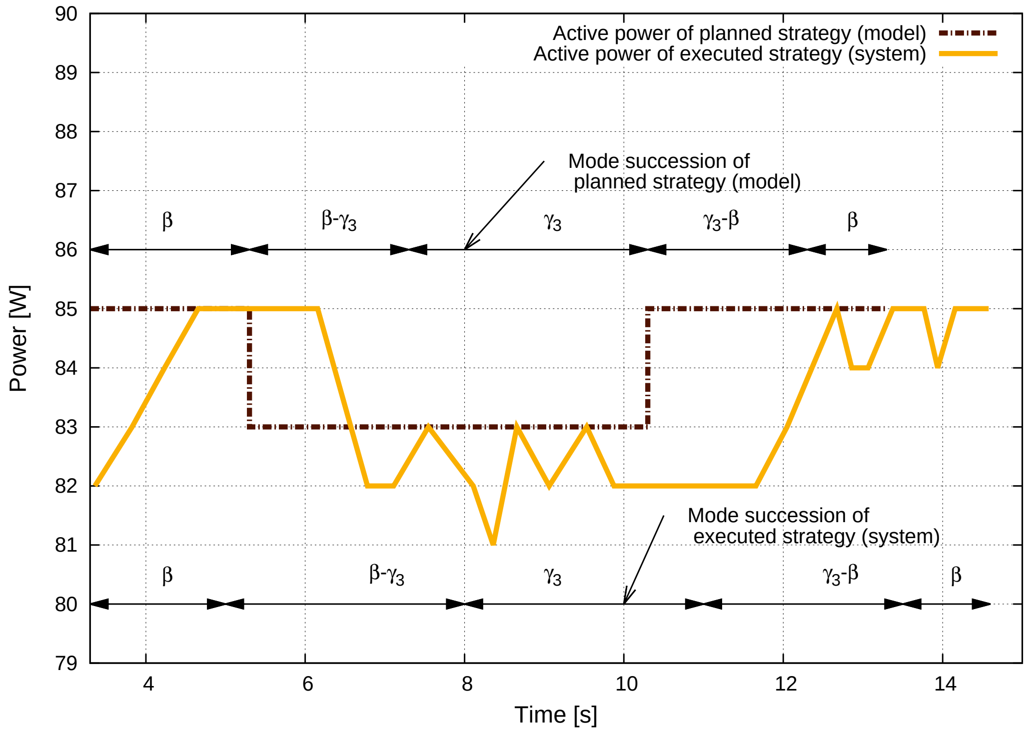


Figure C.6: *Ver.tbS.1*: Planned (model) and actual (system) input power over time in Test Bed tb_S

In Figure C.7, Test Bed tb_S has an initial and target mode β within a 30 [s] pause interval. The planned strategy has an energy demand of 2.455 [J] whereas the actual strategy has an energy demand of 2.723 [J]. This results in an underestimation of the actual energy demand of dev_{energy} = -9,8%

C.3.2 Mode delay deviations in Test Bed tb_M

Figure C.8 and Figure C.9 show the results for dev_{delay} = -0,2% deviation (upper figures) compared to a subsystem-specific dev_{delay} deviation (lower figures) between model and subsystem.

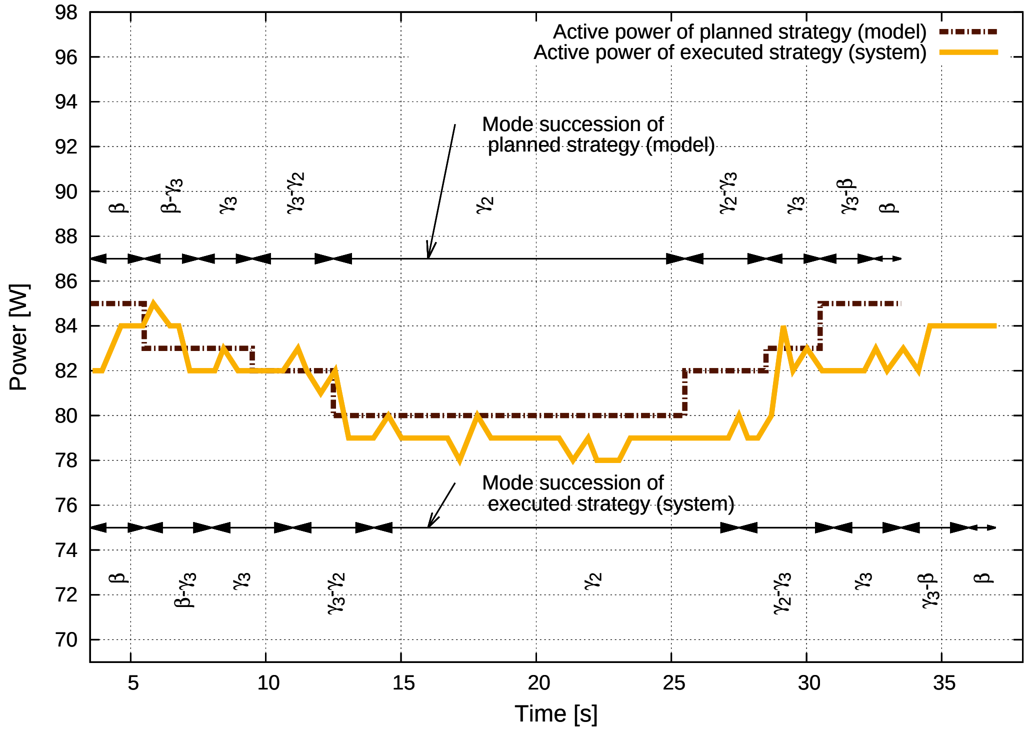
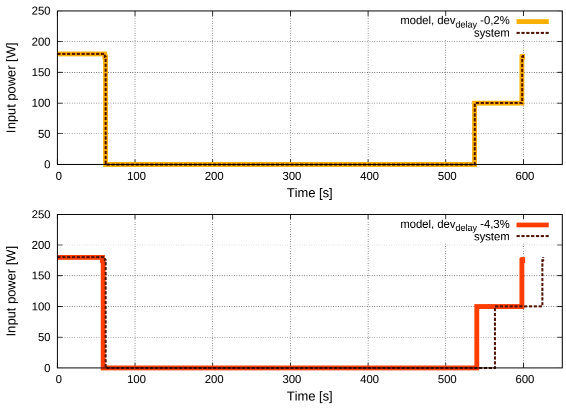
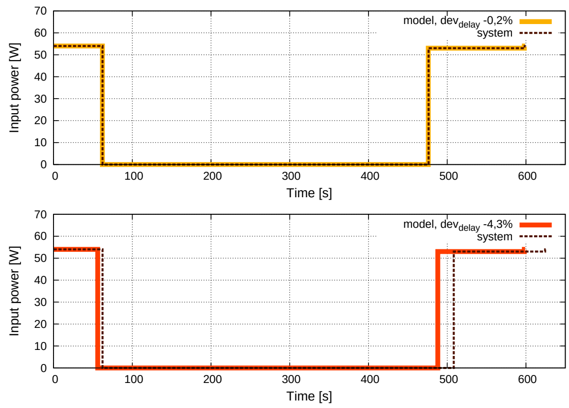


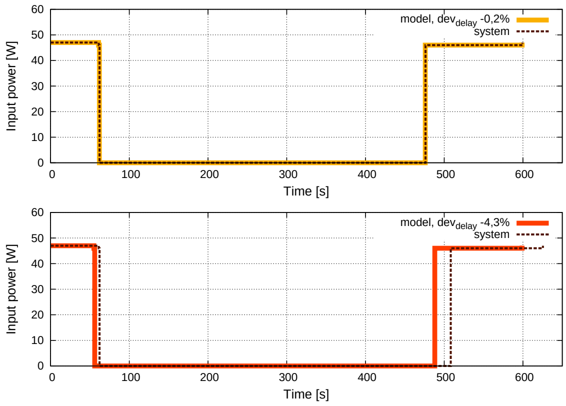
Figure C.7: *Ver.tbS.2*: Planned (model) and actual (system) input power over time in Test Bed tb_S



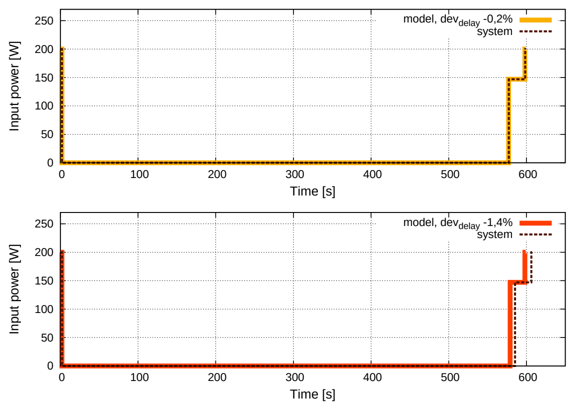
(a) Subsystem sub_1



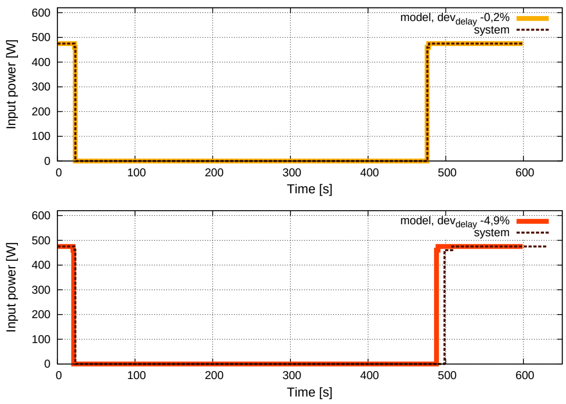
(b) Subsystem sub_2



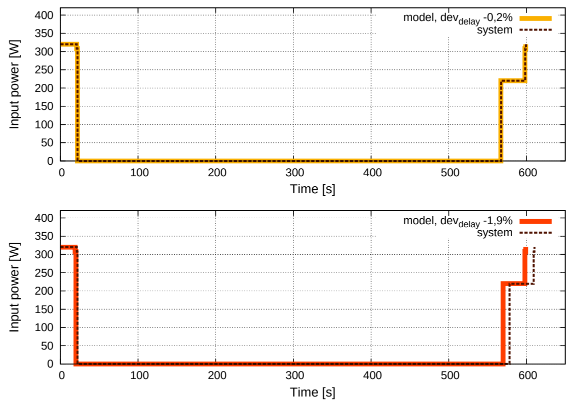
(c) Subsystem sub_3



(d) Subsystem sub_4

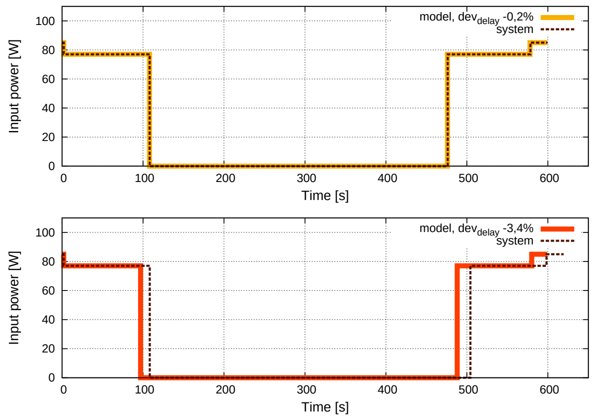


(e) Subsystem sub_5

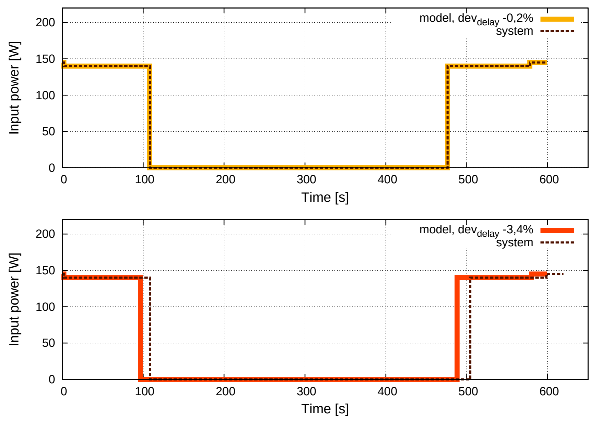


(f) Subsystem sub_6

Figure C.8: *Ver.tbM.0 and Ver.tbM.7*: Planned and actual input power for Subsystems sub_1 to sub_6 plotted over time, (dev_{delay} = -0,2% and subsystem-specific deviation dev_{delay})



(a) Subsystem sub_7



(b) Subsystem sub_8

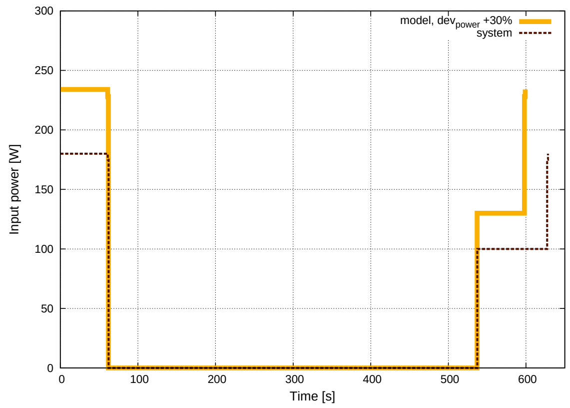
Figure C.9: *Ver.tbM.0 and Ver.tbM.7*: Planned and actual input power for Subsystems sub_7 and sub_8 plotted over time, (dev_{delay} = -0,2% and subsystem-specific deviation dev_{delay})

C.3.3 Input power deviations in Test Bed tb_M

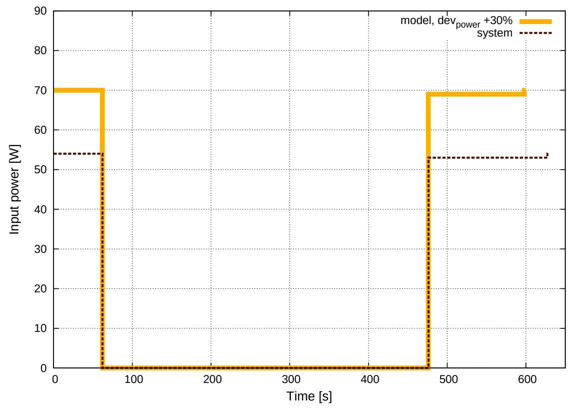
Input power deviations dev_{power} are checked for their effects on the accuracy of prediction of the energy demand dev_{energy} in Table C.1.

Scen.	**Model/ System**	**Input**			**Output**	
		Initial	Final	Input power deviation: dev_{power}	Energy input [kJ] Model/ System	Planned strategy feasible? (●/○)
Val.tbM.0 (= Ver.tbM.0)	Model System	$\alpha^{(9)}$ $\alpha^{(9)}$	$\alpha^{(9)}$ $\alpha^{(9)}$	0	428,7 429,8	●
Val.tbM.1	Model System	$\alpha^{(9)}$ $\alpha^{(9)}$	$\alpha^{(9)}$ $\alpha^{(9)}$	+10	471,8 429,8	●
Val.tbM.2	Model System	$\alpha^{(9)}$ $\alpha^{(9)}$	$\alpha^{(9)}$ $\alpha^{(9)}$	+20	514,4 429,8	●
Val.tbM.3	Model System	$\alpha^{(9)}$ $\alpha^{(9)}$	$\alpha^{(9)}$ $\alpha^{(9)}$	+30	557,3 429,8	●
Val.tbM.4	Model System	$\alpha^{(9)}$ $\alpha^{(9)}$	$\alpha^{(9)}$ $\alpha^{(9)}$	-10	385,7 429,8	●
Val.tbM.5	Model System	$\alpha^{(9)}$ $\alpha^{(9)}$	$\alpha^{(9)}$ $\alpha^{(9)}$	-20	342,7 429,8	●
Val.tbM.6	Model System	$\alpha^{(9)}$ $\alpha^{(9)}$	$\alpha^{(9)}$ $\alpha^{(9)}$	-30	300,0 429,8	●

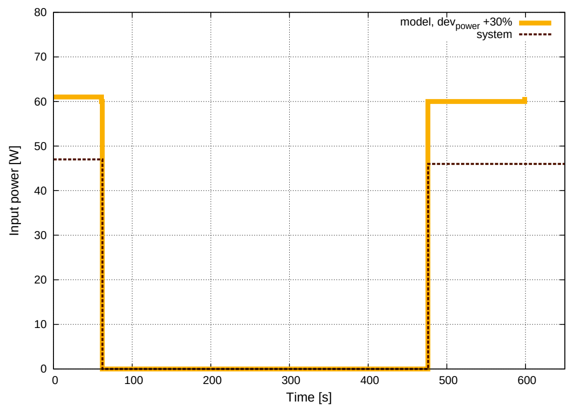
Table C.1: *Val.tbM.x*: Scenario overview for deviation tests



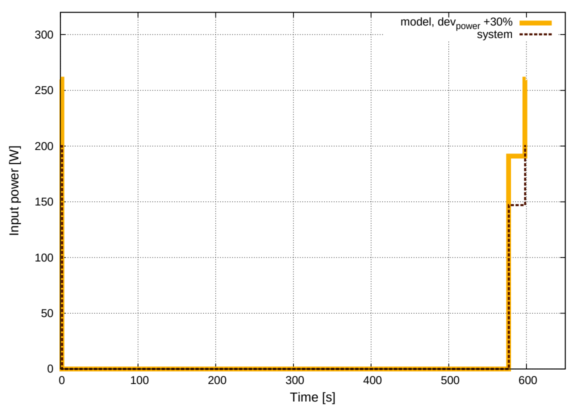
(a) Subsystem sub_1



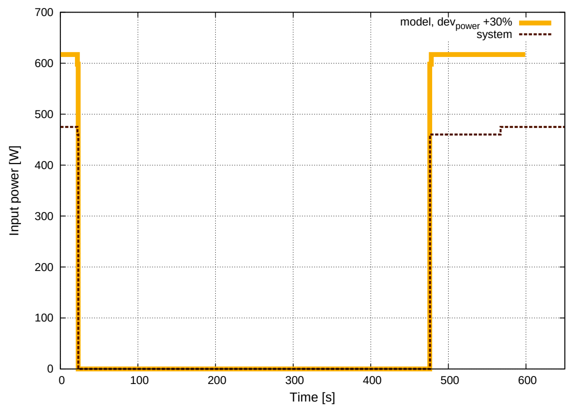
(b) Subsystem sub_2



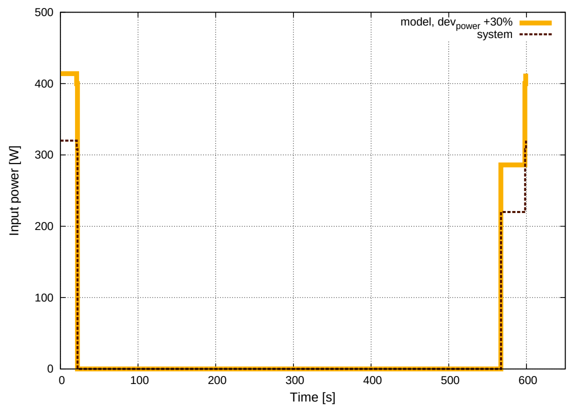
(c) Subsystem sub_3



(d) Subsystem sub_4

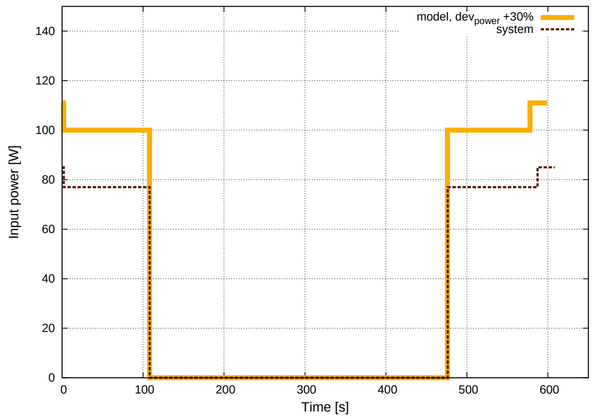


(e) Subsystem sub_5

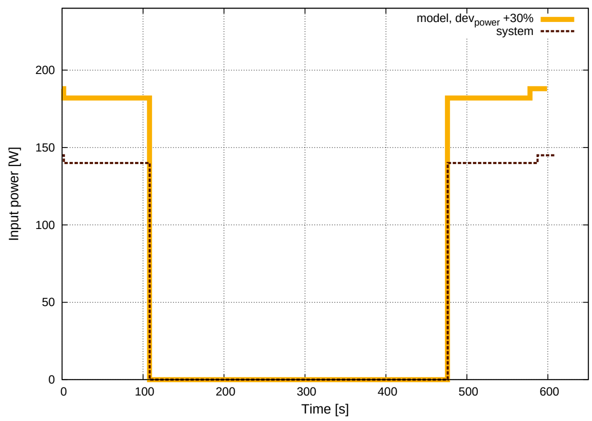


(f) Subsystem sub_6

Figure C.10: *Val.tbM.3*: Modeled input power and actual input power of Subsystems sub_1 to sub_6, (dev_{power} = +30%)



(a) Subsystem sub_7



(b) Subsystem sub_8

Figure C.11: *Val.tbM.3*: Modeled input power and actual input power of Subsystems sub_7 and sub_8, (dev_{power} = +30%)

C.4 Reduction of energy demands by strategies

C.4.1 Short-term pause intervals in Test Bed tb_M

Table C.2 shows the resulting energy savings of the test bed for short-term pause intervals. Initial and target modes of the subsystems are β each.

Scenario	**Input**		**Output**			
	Initial mode = Target mode	Pau [min]	eni_{str} [kJ]	eni_{idl} [kJ]	ens^{abs}_{pau} [kJ]	ens^{rel}_{pau} [%]
Eco.tbM.1.1	$\beta^{(9)}$	2,5	94	463	369	-79,7
Eco.tbM.1.2	$\beta^{(9)}$	5	113	926	813	-87,8
Eco.tbM.1.3	$\beta^{(9)}$	8	130	1.482	1.352	-91,2
Eco.tbM.1.4	$\beta^{(9)}$	10	142	1.710	1.852	-91,7
Eco.tbM.1.5	$\beta^{(9)}$	12	154	2.069	2.223	-92,6
Eco.tbM.1.6	$\beta^{(9)}$	15	172	2.779	2.607	-93,8
Eco.tbM.1.7	$\beta^{(9)}$	20	202	3.504	3.706	-94,2

Table C.2: Variation of the length of unproductive phases for evaluation of the energy savings potential within short-term pause intervals

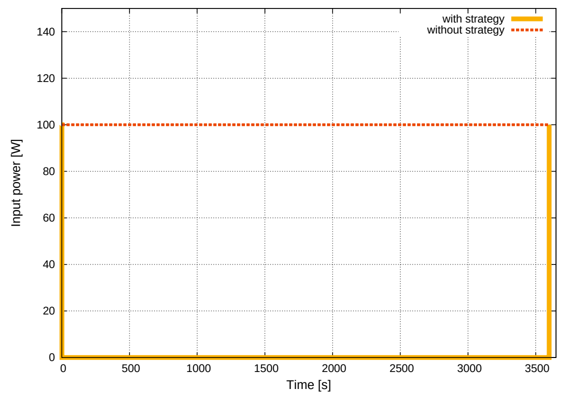
C.4.2 Long-term pause intervals in Test Bed tb_M

Table C.3 illustrates the results for long-term pause intervals of the Test Bed. The initial and target modes of the subsystems are the idling modes given in Section B.3.

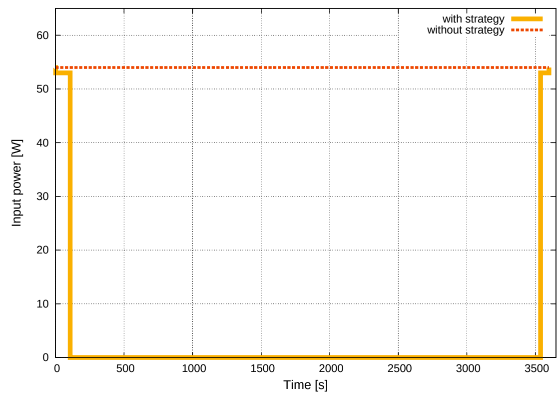
Scenario	**Input**		**Output**			
	Initial mode = Target mode	Pau [min]	eni_{str} [kJ]	eni_{idl} [kJ]	ens^{abs}_{pau} [kJ]	ens^{rel}_{pau} [%]
Eco.tbM.2.1	$<\gamma,\beta,\beta,\beta,\gamma,\gamma,\beta,\beta,\gamma>$	30	55	3.114	3.059	-98,2
Eco.tbM.2.2	$<\gamma,\beta,\beta,\beta,\gamma,\gamma,\beta,\beta,\gamma>$	45	55	4.671	4.616	-98,8
Eco.tbM.2.3	$<\gamma,\beta,\beta,\beta,\gamma,\gamma,\beta,\beta,\gamma>$	60	55	6.228	6.173	-99,1
Eco.tbM.2.4	$<\gamma,\beta,\beta,\beta,\gamma,\gamma,\beta,\beta,\gamma>$	120	55	12.456	12.401	-99,5
Eco.tbM.2.5	$<\gamma,\beta,\beta,\beta,\gamma,\gamma,\beta,\beta,\gamma>$	180	55	18.629	18.574	-99,7
Eco.tbM.2.6	$<\gamma,\beta,\beta,\beta,\gamma,\gamma,\beta,\beta,\gamma>$	420	55	24.857	24.802	-99,8

Table C.3: Variation of the length of unproductive phases for the evaluation of the energy savings potential within long-term pause intervals

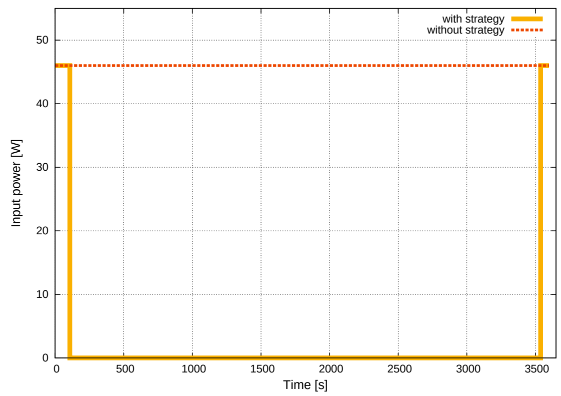
Scenario *Eco.tbM.2.3* is visualized in Figure C.12 and Figure C.13.



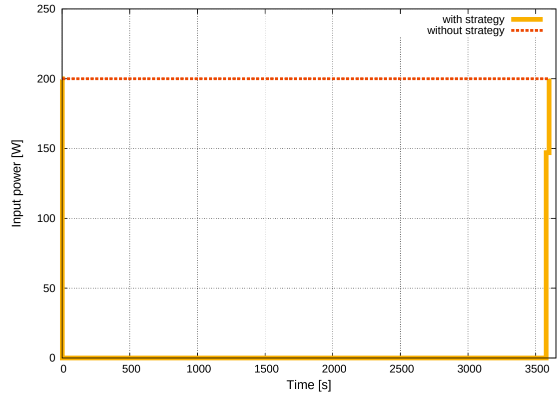
(a) Subsystem sub_1



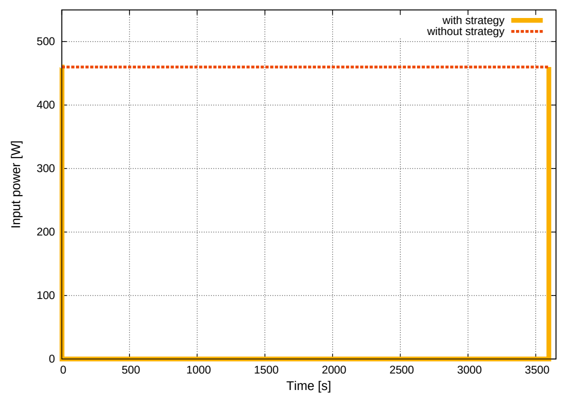
(b) Subsystem sub_2



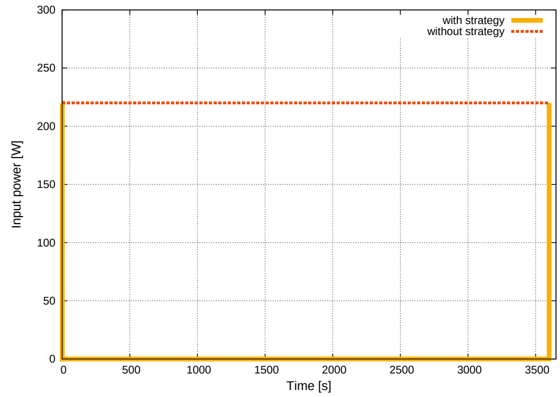
(c) Subsystem sub_3



(d) Subsystem sub_4

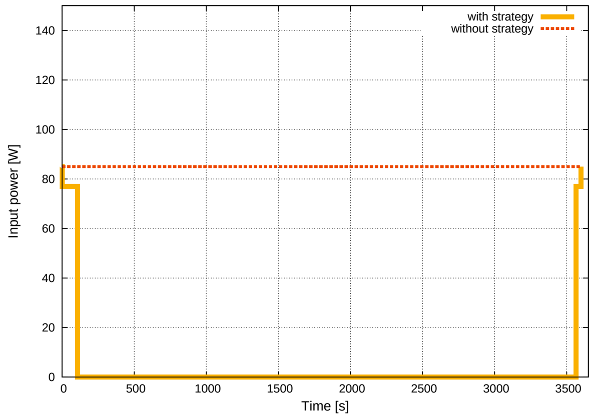


(e) Subsystem sub_5

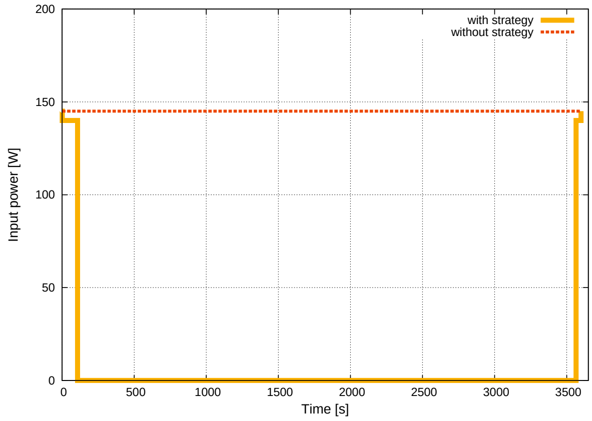


(f) Subsystem sub_6

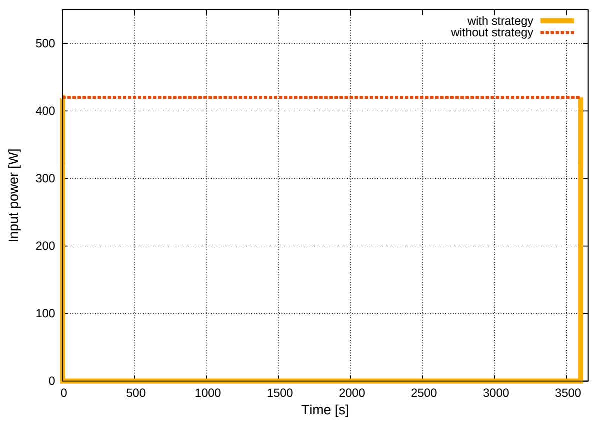
Figure C.12: *Eco.tbM.2.3*: Input power over time during a 60 minutes pause interval, with and without strategies for Subsystems sub_1 to sub_6



(a) Subsystem sub_7



(b) Subsystem sub_8



(c) Subsystem sub_9

Figure C.13: *Eco.tbM.2.3*: Input power over time during a 60 minutes pause interval, with and without strategies for Subsystems sub_7 to sub_9

Curriculum vitae

Dipl.-Wirtsch.-Ing. Sebastian Gabriel Mechs

born in Schweinfurt on December, 21 1982, German citizenship

Education and Services

05/2010-05/2013	Industry-sponsored Ph. D. in Computer Science and Automation Technology, Clausthal University of Technology, Department of Informatics & Siemens Corporate Technology, Munich
10/2003-10/2009	Diploma in Industrial Engineering and Management, Friedrich-Alexander-Universität, Erlangen-Nürnberg Thesis: "Development of a modular, agent-based system for fault detection in production plants"
09/2006-05/2007	Business studies abroad, Ecole de Management Strasbourg, Business School, Strasbourg, France
07/2002-04/2003	Compulsory community service, Leopoldina hospital
09/1993-05/2002	Diploma from German secondary school qualifying for university admission, Walther-Rathenau-Gymnasium

Industrial and Professional Experience

11/2008-03/2009	Student trainee in Industrial Communication Technology, Siemens AG, Industrial Automation, Nürnberg
06/2008-10/2008	Internship in Industrial Communication Technology, Siemens Energy & Automation, Norcross (Georgia), USA
09/2007-06/2008	Student trainee in Industrial Communication Technology, Siemens AG, Industrial Automation, Nürnberg
08/2005-09/2005	Internship in Customer Relationship Management, Generali Group Ochsenfurt
05/2003-06/2003	Internship in Quality Management, FAG Kugelfischer AG & Co. KG Schweinfurt

Munich, May 2013

Conference and journal papers

- Mechs, S. and Lamparter, S. and Müller, J. P.
  On Evaluation of Alternative Switching Strategies for Energy-Efficient Operation of Modular Factory Automation Systems, In: Proceedings of the 17th IEEE Conference on Emerging Technologies and Factory Automation, Kraków, Poland, pp. 1-8, 2012

- Mechs, S. and Lamparter, S. and Peschke, J.
  Steigerung der Energieeffizienz in Automatisierungssystemen durch Start-Stopp-Automatik, In: Automation 2012, 13. Branchentreff der Mess- und Automatisierungstechnik, Baden-Baden, Germany, VDI-Berichte 2171, VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik, pp. 251–254, 2012

- Mechs, S. and Müller, J. P. and Lamparter, S. and Peschke, J.
  Networked priced timed automata for energy-efficient factory automation, In: American Control Conference, Montréal, Canada, pp. 5310–5317, 2012

- Mechs, S. and Lamparter, S. and Peschke, J. and Müller, J. P.
  Efficient Identification of Energy-Optimal Switching and Operating Sequences for Modular Factory Automation Systems, In: 26th International Conference on Industrial Engineering & Other Applications of Applied Intelligent Systems, Amsterdam, The Netherlands, Lecture Notes in Artificial Intelligence, Springer, accepted for publication, 2013

- Mechs, S. and Lamparter, S. and Peschke, J. and Müller, J. P.
  Start-Stopp-Automatik für Nicht-Produktivphasen – Höhere Energieeffizienz in Automatisierungssystemen, In: atp edition, Deutscher Industrieverlag, München, Germany, accepted for publication, 2013

Patent applications

- Mechs, S. and Lamparter, S. and Peschke, J.
  Formal Model for Controlling and Analyzing Timed and Energetic Switching Behavior in Automation Systems, submitted 2012-05-08, international file,
  application number: *PCT/EP2012/058452*

- Mechs, S. and Lamparter, S. and Peschke, J.
  Verfahren zur effizienten Generierung von durchführbaren Schaltstrategien und Betriebsstrategien für Automatisierungs(sub)systeme zur Optimierung des energetischen Betriebsverhaltens, submitted 2012-09-17, national file, application number: *102012216575.4*

Datum: 08.05.2013
Dipl.-Wirtsch.-Ing. Sebastian Gabriel Mechs

EIDESSTATTLICHE ERKLÄRUNGEN

Hiermit erkläre ich an Eides Statt, dass ich die bei der Fakultät für Mathematik/Informatik und Maschinenbau der Technischen Universität Clausthal eingereichte Dissertation selbständig und ohne unerlaubte Hilfe angefertigt habe.
Die benutzten Hilfsmittel sind vollständig angegeben.

....
Unterschrift (Dipl.-Wirtsch.-Ing. Sebastian Gabriel Mechs)

Hiermit erkläre ich an Eides Statt, dass ich bisher noch keinen Promotionsversuch unternommen habe.

....
Unterschrift (Dipl.-Wirtsch.-Ing. Sebastian Gabriel Mechs)