Automation Architecture for Demand Response on Aqueous Parts Cleaning Machines

Vom Fachbereich Maschinenbau
an der Technischen Universität Darmstadt
zur
Erlangung des Grades eines Doktor-Ingenieurs (Dr.-Ing.)
genehmigte

DISSERTATION

vorgelegt von

Daniel Fuhrländer-Völker, M. Sc.

aus Frankfurt am Main

Berichterstatter:	Prof. Dr.-Ing. Matthias Weigold
Mitberichterstatter:	Prof. Dr.-Ing. Reiner Anderl
Tag der Einreichung:	18.04.2023
Tag der mündlichen Prüfung:	20.06.2023

Darmstadt 2023

D17

Fuhrländer-Völker, Daniel: Automation Architecture for Demand Response on Aqueous Parts Cleaning Machines
Darmstadt, Technical University Darmstadt
Year of publication of the dissertation on TUprints: 2023
URN: urn:nbn:de:tuda-tuprints-242591
Day of the oral examination: 20.06.2023

Foreword of the Editor

To reduce greenhouse gas emissions, the share of renewable energies in electricity production is increasing worldwide. However, renewable energy generation fluctuates as it depends on the weather. At the same time, the industrial sector has a large share of the total electrical energy consumption and current research shows that it is possible to adapt this energy consumption to the fluctuating renewable energy generation by means of demand response. Production equipment such as aqueous parts cleaning machines contain untapped potential for energy flexibilisation and thus for supporting demand response. The use of demand response in the industrial sector is made possible by the transformation of industrial automation from hierarchically structured to service-oriented architectures.

This thesis uses these new structures and develops an automation architecture to implement demand response on aqueous parts cleaning machines. The first part is a demand response potential analysis for aqueous parts cleaning machines that determines the potential of the machines. Then, an automation architecture is developed that enables the execution of demand response. This consists of an object-oriented automation programme and a data model for the data exchange between the machine control level and the IT level. Finally, a demand response control algorithm is presented that executes demand response measures and is integrated into an IT framework.

Previous research is mostly limited to the development and simulative validation of demand response algorithms. Only few approaches show the implementation of demand response on real production plants, but then do not describe in a structured way how the plants were adapted to enable the execution of the algorithms. The demand response automation architecture developed in this thesis fills this research gap by describing how the automation of production plants must be designed to enable demand response in industry. Thus, it is transferable and scalable to be used on different machines of different sizes.

In this thesis, the method is applied to a real aqueous parts cleaning machine and not only validated theoretically or in simulation. The application consists of the object-oriented demand response automation program implemented on the machine's PLC and the demand response data model in the form of an OPC UA data model. The implemented demand response automation architecture is validated in a field test.

Darmstadt, July 2023 Prof. Dr.-Ing. Matthias Weigold

Acknowledgements

This thesis was developed in the Energy Technologies and Applications in Production (ETA) research group which is part of the institute of Production Management, Technology and Machine Tools (PTW) at the Technical University of Darmstadt. I would like to extend special thanks to Prof. Dr.-Ing. Matthias Weigold, head of the PTW, for the trusting cooperation and support during my research work as well as the feedback on my research. I would also like to thank Prof. Dr.-Ing. Reiner Anderl for taking on the role of the co-lecturer.

The research of this work was carried out as part of the German research projects PHI-Factory, Synergie and KI4ETA, funded by the Federal Ministry for Economic Affairs and Climate Action (BMWK) and the Federal Ministry of Education and Research (BMBF). I would like to thank all research and industry partners with whom I have worked in these projects for the discussions and their input to this thesis.

I would like to thank my colleagues at the PTW and especially the members of the ETA research group for their contribution to the success of this work through the intensive discussions, motivating words and above all, for supporting me over the last few months such that I could devote myself on the completion of this thesis. In particular, I would like to thank Niklas Panten, Philipp Schraml and Nina Raue (née Strobel) who's research inspired me for this thesis. Furthermore, I want to thank Fabian Borst, Lukas Theisinger and Heiko Ranzau for the valuable discussions on automation systems and optimisation and Jonathan Magin for his input on aqueous parts cleaning machines. Special thanks also go to my student assistant Florian Körner for his support regarding Tikz graphics.

I owe special thanks to Ghada Elserafi and Martin Lindner for the cooperation and intensive research exchange in the last years and especially their constructive feedback on this thesis, which contributed greatly to this work's quality. Moreover, I would like to thank Nicholas Mackinnon for his spelling and grammar correction.

My special gratitude goes to the Diss Booster Team. I would like to thank Thomas Kohne for motivating me in the final phase of the thesis and for fertile discussions on research concept and thesis structure. I would like to thank Benedikt Grosch for the close collaboration in the past years, for many inspiring discussions on our research and for helping finding all the errors in my code until it finally worked. I would like to thank you both for being there for me in difficult situations and for all the fun we had together.

I would like to express my heartfelt thanks to my family on whose help and support I can always rely. I would especially like to thank my parents, who taught me that I can achieve anything and who always believe in me.

Finally, I want to thank my wife Mona for valuable discussions, constructive and extensive feedback on this thesis and unconditional and endless support during the last years. I could have not done this without you.

Darmstadt, July 2023 Daniel Fuhrländer-Völker

Abstract

The share of fluctuating renewable energy in the electricity grid is increasing strongly, primarily in industrialised countries. The industrial sector accounts for a large share of the total electrical energy consumption and current research shows that it is possible to adapt this energy consumption to the fluctuating renewable energy generation using demand response (DR). Especially aqueous parts cleaning machines have a high DR potential. However, only a few approaches exist that show how DR measures can be implemented on real production machines.

This work develops a method that enables the execution of DR measures on aqueous parts cleaning machines. The so-called demand response automation architecture design (DRAAD) method consists of a DR potential analysis, a DR automation architecture including a DR automation program and a DR data model, as wall as a DR control algorithm.

The DR potential analysis analyses the technical DR potential of the machine components for the DR method *store energy inherently* and of the cleaning process for the DR method *interrupt process*. The DR potential analysis uses only the machine documentation and simple calculations, such that it can be carried out by employees of the machine manufacturer.

The framework for the DR automation architecture is a cyber-physical production system. This consists of the physical aqueous parts cleaning machine, its digital twin, external elements such as the energy market and a cyber-physical interface representing the communication in the cyber-physical production system. The digital twin includes the DR automation program, the DR data model and the DR process model, which is used by the DR control algorithm, in the digital master. The digital twin also includes a digital shadow and digital services.

The object-oriented DR automation program implements sensor, actuator, and system objects as well as functions that enable the execution of the DR methods *store energy inherently* and *interrupt process*. In addition to DR functions, functional safety functions are included. The communication between DR automation program and DR control algorithm is modelled in the DR data model. This includes all data points needed for the calculation (observing) and execution (controlling) of the two DR actions.

The DR control algorithm, a model predictive control algorithm, minimises the energy cost of the aqueous parts cleaning machine based on varying energy prices. Both DR measures are implemented and the approach is scalable and transferable to different aqueous parts cleaning machines.

The DRAAD method is applied and validated on the aqueous parts cleaning machine MAFAC KEA in the ETA research factory. The DR potential analysis of the machine results in a DR power potential of 87 % of the machine's rated power for *store energy inherently* and a DR energy power potential of 99 % of the energy consumption of the reference cleaning process for *interrupt process*. In the field test, a power change of 49 % and an energy shift of 82 % can be retrieved.

Keywords: Cyber-physical production system, digital twin, data model, energy-flexibility, model predictive control

Zusammenfassung

Weltweit steigt der Anteil erneuerbarer Energien im Stromnetz stark an. Die Erzeugung erneuerbarer Energien schwankt, da sie vom Wetter abhängig ist. Der industrielle Sektor hat einen großen Anteil am gesamten elektrischen Energieverbrauch und aktuelle Forschung zeigt, dass es möglich ist, diesen Energieverbrauch mittels Demand Response (DR) an die schwankende erneuerbare Energieerzeugung anzupassen. Vor allem wässrige Bauteilreinigungsanlagen besitzen ein hohes DR Potential, da ihr Reinigungstank als Energiespeicher genutzt und der Reinigungsprozess unterbrochen werden kann. Allerdings existieren nur wenige Ansätze, die zeigen, wie DR Maßnahmen an realen Produktionsanlagen implementiert werden können.

In dieser Arbeit wird eine Methode entwickelt, die die Ausführung von DR Maßnahmen auf wässrigen Bauteilreinigungsanlagen ermöglicht. Die sogenannten Demand Response Automationsarchitektur Design (DRAAD) Methode umfasst eine DR Potentialanalyse, eine DR Automationsarchitektur, bestehend aus einem DR Automationsprogramm und einem DR Datenmodell, sowie einen DR Regelungsalgorithmus.

In der DR Potentialanalyse wird das technische DR Potential der Maschinenkomponenten für die DR Maßnahme *Energie inhärent speichern* sowie des Reinigungsprozesses für die DR Maßnahme *Prozess unterbrechen* ermittelt. Die DR Potentialanalyse nutzt nur die Maschinendokumentation und einfache Rechnungen, sodass eine Durchführung durch Mitarbeitende des Maschinenbauunternehmens ermöglicht wird.

Der Rahmen für die DR Automationsarchitektur ist ein cyber-physisches Produktionssystem, bestehend aus der physischen wässrigen Bauteilreinigungsanlage, ihrem digitalen Zwilling, der einen digitalen Master, digitalen Schatten und digitale Services umfasst, externen Elementen wie dem Energiemarkt sowie einem cyber-physischen Interface, dass die Kommunikation im cyber-physischen Produktionssystem repräsentiert. Der Digitale Master beinhaltet das DR Automationsprogramm, das DR Datenmodell sowie das DR Optimierungsmodell, welches durch den DR Regelungsalgorithmus genutzt wird.

Das objekt-orientierte DR Automationsprogramm enthält Sensor-, Aktor-, und Systemobjekte sowie Funktionen, die die Speicherung der Energie in Maschinenkomponenten mit einem hohen DR Potential sowie das Unterbrechen des Reinigungsprozesses ermöglichen. Neben DR Funktionen sind Funktionen für funktionale Sicherheit integriert.

Die Kommunikation zwischen DR Automationsprogramm und DR Regelungsalgorithmus wird im DR Datenmodell modelliert. Dies beinhaltet alle Datenpunkte, die für die Berechnung (beobachtend) und Ausführung (steuernd) beider DR Maßnahmen benötigt werden. Der DR Regelungsalgorithmus, ein Model Predictive Control Algorithmus, minimiert die Energiekosten der wässrigen Bauteilreinigungsanlage basierend auf variierenden Energiepreisen.

Die DRAAD Methode wird auf die wässrige Bauteilreinigungsanlage MAFAC KEA in der ETA-Fabrik angewandt und validiert. Die DR Potentialanalyse resultiert in einem Potential zur Leistungsänderung von 87 % der Anschlussleistung und einem Potential zur Verschiebung des Energieverbrauchs von 99 %. Im Feldversuch kann eine Leistungsänderung von 49 % und eine Energieverschiebung von 82 % abgerufen werden.

Stichwörter: Cyber-physisches Produktionssystem, Digitaler Zwilling, Datenmodell, Energieflexibilität, Model Predictive Control

Contents

List of Figures

List of Tables

Abbreviations

AI	artificial intelligence.
AMQP	Advanced Message Queuing Protocol.
APCM	aqueous parts cleaning machine.
API	application programming interface.
CIRP	The International Academy for Production Engineering.
DDS	Data Distribution Service.
DR	demand response.
DRAAD	demand response automation architecture design.
EPEX	European Power Exchange.
ERP	enterprise resource planning.
ETA	Energy Technologies and Applications in Production.
EU	European Union.
FMU	Functional Mock-up Unit.
HMI	human machine interface.
HTTPS	Hypertext Transfer Protocol Secure.
HVAC	heating, ventilation, and air conditioning.
IEC	International Electrotechnical Commission.
IEEE	Institute of Electrical and Electronics Engineers.
IFAC	International Federation of Automatic Control.
IIC	Industry Internet of Things Consortium.
IoT	Internet of Things.
IP	Internet Protocol.
IPC	industrial personal computer.
IT	information technology.
JSON	JavaScript Object Notation.
MES	manufacturing execution system.
MILP	mixed integer linear programming.
MPC	model predictive control.
MQTT	Message Queuing Telemetry Transport.
OPC UA	Open Platform Communications Unified Architecture.
OPC UA FX	OPC UA Field eXchange.
OT	operating technology.

PID	proportional–integral–derivative.
PLC	programmable logic controller.
PTW	Production Management, Technology and Machine Tools.
RAMI4.0	Reference Architecture Model Industrie 4.0.
REST	Representational State Transfer.
ROS	Robot Operating System.
SCADA	supervisory control and data acquisition.
TCP/IP	Transmission Control Protocol/Internet Protocol.
UML	Unified Modeling Language.
US	United States of America.
VDI	Association of German Engineers e.V..
XML	Extensible Markup Language.
XMPP	Extensible Messaging and Presence Protocol.
YAML	YAML Ain’t Markup Language.

Symbols

Symbol	Description
$a_{n,k}$	binary variable, one during process step n at time step k
$\tilde{a}_{n,k}$	binary variable, one during and before process step n at time step k
$\mathbf{A}$	matrix containing all binary variables $a_{n,k}$
$\tilde{\mathbf{A}}$	matrix containing all binary variables $\tilde{a}_{n,k}$
$\mathcal{A}_g$	set of machine modules active in the process step g
b_n	binary help variable
$\mathcal{B}_n$	set of machine modules active in the process event n
C_k	electricity price at time step k
c_p	specific heat capacity of a material
$c_\mathrm{p,fluid}$	specific heat capacity of cleaning fluid
$c_\mathrm{p,parts}$	specific heat capacity of parts
d_f	duration machine module f is active during complete cleaning process
$d_{f,g}$	duration f-th machine module is switched on during process step g
$d_{f,j}$	duration a part traverses process chamber j of machine module f
d_g	duration of process step g
d_n	duration of cleaning process event n
$\mathbf{d}$	durations of all cleaning process events
d_clean	duration of cleaning
d_dry	duration of drying
d_event	fixed duration of an event
d_load	duration of loading
d_meas	measured activation duration
d_start	remaining duration of active event
E	total energy of a system
f	index of machine module
F	total number of machine modules
$\mathcal{F}$	set of machine modules rated green or yellow in the DR potential analysis
g	index of process step
G	total number of process steps
$\mathcal{G}$	set of process steps rated green or yellow in the DR potential analysis
h_k	binary setpoint of tank heater at time step k
$h_{l,k}$	setpoint of machine module l at time step k
$\mathbf{h}$	setpoints of tank heater at all time steps

Symbol	Description
$\mathbf{H}$	setpoints of all machine modules at all time steps
i_k	binary variable, is one if an interruption is set at time step k
j	index of cleaning chamber
J	total number of cleaning chambers
k	time step
K	optimization horizon
l	index of machine modules selected for *store energy inherently*
L	total number of machine modules selected for *store energy inherently*
m	mass of a material
m_{fluid}	mass of cleaning fluid
m_{parts}	mass of a part
n	index of cleaning process events
n_{start}	currently activated cleaning process event
N	total number of cleaning process events
$\mathcal{N}_{\text{parts}}$	total number of parts in one cleaning tray
Q	energy supplied to a system as heat
$\dot{Q}$	heat flow
$\dot{Q}_{\text{env}}$	heat flow from the cleaning liquid to the environment
$\dot{Q}_{\text{parts}}$	heat flow from the cleaning liquid to the parts
$\dot{Q}_{\text{spray}}$	heat flow from the cleaning liquid to the aqueous parts cleaning machine
P_f	rated power of machine module f
P_g	cumulated rated power of machine modules active in process step g
P_l	rated power of machine module l selected for *store energy inherently*
P_n	cumulated rated power of machine modules active in process element n
P_{clean}	cumulated rated power of machine modules active in process step cleaning
P_{dry}	cumulated rated power of machine modules active in process step drying
P_{flex}	absolute achievable energy-flexible power demand
P_{heat}	rated power of tank heater
P_{int}	cumulated rated power of machine modules active in process step interruption
P_{meas}	average measured power
P_{module}	cumulated rated power of machine modules selected for *store energy inherently*
P_{total}	rated power of the aqueous parts cleaning machine
R	thermal resistance
R_{env}	thermal resistance between liquid and production hall
R_{spray}	thermal resistance between liquid and aqueous parts cleaning machine

Symbol	Description
s_n	start of cleaning process element n
S	fixed end time step of cleaning process
t	time
Δt	time interval
T	temperature
T_k	tank temperature at time step k
T_{env}	temperature of the production hall (environment)
T_{f1}	temperature of the first fluid during heat exchange
T_{f2}	temperature of the second fluid during heat exchange
T_{lb}	lower limit for tank temperature
T_{ub}	upper limit for tank temperature
T_{parts}	temperature of the parts before loading
T_{start}	tank temperature for $k = 0$
ΔT	temperature difference
ΔT_k^+	tank temperature increased by tank heater at time step k
ΔT_k^-	tank temperature decrease at time step k
$\Delta T_{\text{clean},k}^-$	heat loss during spray cleaning at time step k
$\Delta T_{\text{env},k}^-$	heat loss to environment during spray cleaning at time step k
$\Delta T_{\text{parts},k}^-$	heat loss to parts during spray cleaning at time step k
$\Delta T_{\text{spray},k}^-$	heat loss to aqueous parts cleaning machine during spray cleaning at time step k
U	inner energy of a system
ΔU	change of inner energy of a system
$\Delta U_{\text{env},k}$	change of inner energy of a system caused by the temperature loss to the environment
$\Delta U_{\text{parts},k}$	change of inner energy of a system caused by the temperature loss to the environment
$v_{l,k}$	process value of system l at time step k
$\Delta v_{l,k}$	change of process value of system l at time step k
$v_{\text{lb},l}$	lower bound of process value of system l
$v_{\text{ub},l}$	upper bound of process value of system l
$v_{l,\text{start}}$	start process value of system l for $k = 0$
V_{tank}	tank volume
W	work done to a system by its surroundings
W_f	energy demand of machine module f
W_g	energy demand of process step g
W_{flex}	absolute energy DR potential
W_{meas}	measured energy consumption
W_{total}	total energy demand of aqueous parts cleaning machine for one cleaning process
x	setpoint of actuator or system

Symbol	Description
x_{lb}	lower bound of setpoint
x_{ub}	upper bound of setpoint
$z_{n,k}$	variable used for linearisation
α	heat transfer coefficient
β_{env}	regression factor for temperature loss to environment
β_{parts}	regression factor for temperature loss to parts
β_{spray}	regression factor for temperature loss to aqueous parts cleaning machine
δ	duration of a time step k
ρ_{fluid}	density of cleaning fluid
$\tau_{l,k}$	disturbance on system l at time step k
φ_f	share of energy demand of machine module f
φ_g	share of energy demand of process step g
Φ_{P}	power ratio, technical DR potential for *store energy inherently*
$\Phi_{\mathrm{P,meas}}$	power ratio of measured power change
Φ_{W}	energy ratio, technical DR potential for *interrupt process*
$\Phi_{\mathrm{W,meas}}$	energy ratio of measured energy shift

1 Introduction and motivation

This chapter describes the motivation for this thesis in Section 1.1, highlights the work's main contribution in Section 1.2 and explains the structure of this thesis based on the used research methodology in Section 1.3.

1.1 Motivation

In order to reduce CO_2 emissions many countries are integrating more renewable energies [Xu19]. The share of renewable energies in the electricity mix is growing worldwide [Ritc22]: In Germany it has increased from 3.64 % in 1985 to 40.32 % in 2021, in the United Kingdom from 1.33 % to 39.78 %, in the European Union (EU) from 15.66 % to 37.24 % and in the United States of America (US) from 12.18 % to 20.71 %. World wide the share of renewable energy grew from 20.83 % in 1985 to 28.28 % in 2021. The massive integration of renewable energies demands changes in the operation of the electricity grid towards a more flexible operation [Conn16; Aliz16]. In the past, electricity generation followed demand. When demand was high, more electricity was produced, when demand was low, less electricity was produced. This is not possible with renewable energies, as they are linked to environmental influences and produce energy when the sun is shining or the wind is blowing.

One of the biggest consumers of electricity is the industrial sector. In 2021 in the EU, industry accounted for 43 % of the overall electricity consumption [euro21], in the US for 26 % [US E21]. At the same time, several scientific studies in recent years have shown that it is possible to adapt the electricity consumption of industry. Through so-called demand response (DR), it is possible to adapt individual production machines such as machine tools [Schr18] or the supply technology of factories [Pant19] to fluctuating energy production. The energy-flexible control of cross-sectional technologies is also possible [Stro21].

Especially aqueous parts cleaning machines (APCMs) have a great potential for DR. APCMs have media tanks whose thermal inertia can be used as energy storage [Stro20]. In addition, the cleaning process can be interrupted [VDI5207-1]. Studies have shown that a short interruption of the cleaning process has no effect on the cleaning result [Auri09]. APCMs are the most commonly used kind of cleaning machines. A market survey showed that in 2020 86 % of the surveyed industrial companies used APCMs [Rögn21a].

The use of DR is made possible by the transformation of industrial automation. While the classic automation pyramid according to IEC 62264 [IEC62264-1] is hierarchical, new automation structures such as cyber-physical production systems [Mono16] and service-oriented architectures [Voge09] enable a more direct interaction of production machines and information technology (IT)-services that are needed for the execution of DR measures. New communication architectures such as Open Platform Communications Unified Architecture (OPC UA) are implemented in industry [Pfro16; Bruc19; Prof19; Liu20; Deve22] as well as in the electricity market [Claa11; Gil22; Wang22; Zhu22]. They may enable a seamless data exchange from electricity market to factories and individual production machines in the future.

Few approaches exist that implement automation architectures for DR. Related research only implements DR automation architectures on IT-level [Seit19], without describing how to integrate the production machines on operating technology (OT)-level, or focus on energy-efficiency and do not consider DR [Heut19; Vogt22].

In [Pant18] a first basic approach is presented to establish a DR automation architecture for industrial supply systems. This approach is extended in [Pant19] by a DR control algorithm which is validated in a simulation. In [Schr18] the author presents an approach to implement DR on machine tools including a DR potential analysis based on [Abel16], a DR control algorithm and a very brief description of how to adapt the automation system to enable the execution of DR. The method is validated in a field test on a real machine tool.

This work extends the ideas of [Abel16; Pant18; Schr18], transfers parts of their approaches from machine tools and industrial supply systems to APCMs, but focusses on a new method to design a DR automation architecture for the execution of DR on APCMs. The so called demand response automation architecture design (DRAAD) method shows in detail, how to analyse the DR potential of APCMs, how to design a DR automation program, how to implement a DR data model and how to develop a DR control algorithm that controls the execution of the DR measures *store energy inherently* and *interrupt process*.

1.2 Main contribution

The main contribution of this work is the development of the **DRAAD method** that enables the execution of DR on APCMs. The development of the DRAAD method consists of the following steps:

1. Development of a **DR potential analysis** for APCMs. The DR potential analysis takes an existing approach for machine tools [Abel16] and adapts and extends it to analyse APCMs. The DR potential analysis examines the technical DR potential of APCMs for the DR measures *store energy inherently* and *interrupt process*. The DR potential analysis results in a quantified, technical DR potential, compare [Fuhr23b].
2. Development of a DR automation architecture which enables the execution of DR. It consists of two parts:
   (a) Development of a **DR programming scheme** that specifies how an automation program must be designed to execute DR measures. After analysing the DR potential of APCMs the developed object-oriented DR automation program makes this potential available. It specifies objects, including their attributes and methods, to execute the DR measures *store energy inherently* and *interrupt process*. It also includes functionality to guarantee functional safety, compare [Fuhr21; Fuhr22a; Fuhr23b].
   (b) Development of a **DR data model** for the interaction of IT and OT. Combined, the data structures of the object-oriented automation program form a DR data model. This DR data model enables IT-OT-interaction such that OT-controll-systems, such as programmable logic controllers (PLCs), can interact with IT-frameworks to enable DR, compare [Gros22a; Fuhr23a; Fuhr23b].
3. Development of a **DR control algorithm** that controls the execution of DR measures. The DR control algorithm, implemented as a model predictive control (MPC) algorithm,

controls the execution of the DR measures *store energy inherently* and *interrupt process* based on a changing electricity price. The DR control algorithm is integrated in an IT-framework. The interpreter in this IT-framework uses an automation data dictionary to map the variables of the DR data model to the variables of the DR control algorithm, compare [Fuhr23a].

Furthermore this thesis shows the practical application of the DRAAD method:

4. Application of the DRAAD method to a real APCM. The DRAAD method is applied to a real APCM and is not just validated theoretically or in simulation. The application consists of the object-oriented DR automation program, implemented on the APCM's PLC, and the DR data model in the form of an OPC UA data model. It verifies the IT-OT-interaction of the APCM and the IT-framework *eta_utility*. A field test, in which the DR control algorithms successfully executes the DR measures *store energy inherently* and *interrupt process* simultaneously, validates the set-up, compare [Fuhr23a].

1.3 Research methodology and structure of this work

The research methodology of this work is based on the design research methodology presented in [Bles09]. The design research methodology consist of four stages [Bles09, pp 14–17]:

1. At the beginning of the research project stands the *Research Clarification*. The aim of this stage is to define the research goal and the research focus based on first assumptions. It is followed by the
2. *Descriptive Study 1* which is based on a literature review and additional field observations or interviews. The aim is to narrow down the research area and to identify open research questions.
3. Based on the increased understanding of the research area, the assumptions of the first stage are redefined in the *Prescriptive Study* stage and a concept is developed to reach the research goal.
4. In the *Descriptive Study 2*, the concept is evaluated and tested in experiments or empirical studies.

The four stages of the design research methodology do not necessarily take place chronologically one after the other, but are realised in iterations and sometimes take place simultaneously [Bles09, p 17]. This also applies to this work, which is based on several publications in which we developed, applied and tested parts of the concept for a DR automation architecture (Prescriptive Study and Descriptive Study II) based on literature reviews (Descriptive Study I).

This work presents the results of the Research Clarification in Chapter 2 and 3. It starts with the background of this work in Chapter 2, which includes descriptions of DR in general, of DR potential and of DR measures. Furthermore, current changes in industrial automation, IT-OT-communication, the service-oriented architecture OPC UA and control algorithms for DR are shown. The last part is an introduction to APCMs. Based on this background, in Chapter 3 follows the definition of the research objective. The chapter then presents research questions, derived from the research objective, as well as the formulation of assumptions, requirements and design criteria for the DRAAD method. These design criteria are later used to analyse existing research approaches.

The results of the Descriptive Study I, a literature review on existing DR automation architectures, is given in Chapter 4. The search for existing approaches, that implement a DR automation architecture, could not identify any approach that includes all necessary elements to implement DR on APCMs, except the research conducted and published as part of this thesis. Therefore, existing approaches for DR potential analyses, DR automation programming schemes, DR data models and DR control algorithms are analysed separately. These four parts are the main parts of the DRAAD method. None of the existing research can fulfil the defined design criteria.

The development of the DRAAD method is the result of the Prescriptive Study. These results are presented in Chapter 5, 6 and 7. The first part is the DR potential analysis of APCMs, that analyses the DR potential of the DR measures *store energy inherently* and *interrupt process* in Chapter 5. Then, the DR automation architecture is developed in Chapter 6. This includes an object-oriented DR automation program and a DR data model for IT-OT-interaction. Chapter 7 shows the structure of a DR control algorithm that includes a MPC algorithm and that executes the DR measures *store energy inherently* and *interrupt process*.

To conduct the Descriptive Study II the DRAAD method is applied to an APCM and the results are shown in Chapter 8. The applied DRAAD methods identifies the DR potential of the APCM and implements an object-oriented automation program that creates the possibility to execute DR measures as well as a DR data model implemented as an OPC UA data model. The DR control algorithm is adapted to the used APCM and its functionality is verified in a field test using the *eta_utility* IT-framework. The thesis ends with the conclusion in Chapter 9.

2 Demand response, industrial automation and cleaning machines: the fundamentals

This chapter presents the basic knowledge and definitions needed in this research work. Since the DRAAD method is used to implement DR, the chapter starts with the definition of the term DR and related terms in Section 2.1, followed by a explanation of DR potential in Section 2.2. The current transformation in industrial automation, described in Section 2.3, is the basis for the development of the DR automation architecture, which is the core feature of the DRAAD method. Section 2.4 explains the terms IT and OT as well as IT-OT-interaction, which is used in the DR automation architecture. For a better understanding of the DR control algorithm, shows Section 2.6 a brief insight into control algorithms and optimisation models. Section 2.7 describes APCMs in detail. Section 2.8 presents the summary of the chapter.

2.1 Demand response, demand response measures and energy-flexibility

The DRAAD method enables the execution of DR measures. Similarly to [Alba07; Graß14a; Vard15; Walt22], for the term DR this work uses the definition stated by the US Department of Energy.

Definition 2.1 (Demand response) Demand response (DR) is defined as "changes in electric usage by end-use customers from their normal consumption patterns in response to changes in the price of electricity over time, or to incentive payments designed to induce lower electricity use at times of high wholesale market prices or when system reliability is jeopardised." [US D06]

DR is part of demand-side management. Demand-side management summarises measures that aim to adjust the electrical energy demand of consumers in the power grid. Demand-side management measures relevant to industry include, in addition to DR, increasing energy efficiency, which means increasing the output in relation to the energy input [Pale11; Lamp13].

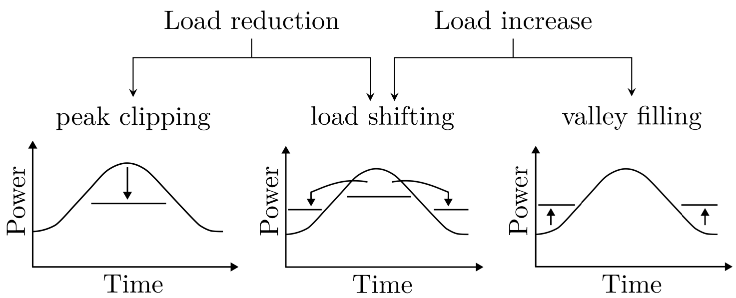


Figure 2.1: Different DR schemes exist. DR aims to reduce the electrical load (*peak clipping*), shift the energy demand in time (*load shifting*) or to increase the load (*valley filling*). Figure based on [Walt22].

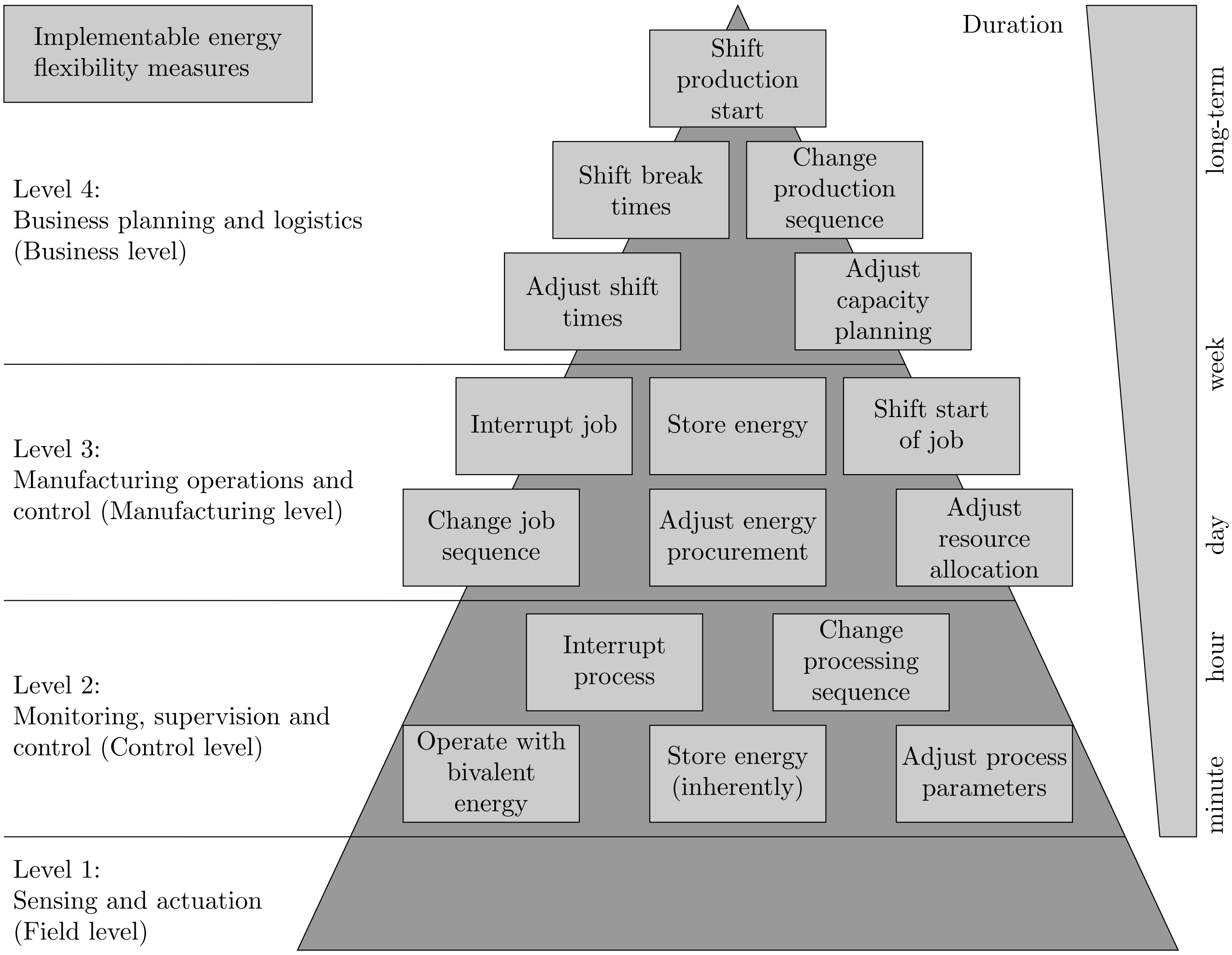


Figure 2.2: DR measures according to the VDI guideline 5207 divided based on the automation levels as defined in IEC 62264. Figure based on [VDI5207-1; IEC62264-1].

DR measures have the aim of reducing electrical load peaks (*peak clipping*, also called *peak shaving*), shifting the energy demand in time (*load shifting*) or increasing the load (*valley filling*) [CIGR11; VDI5207-1], see Figure 2.1. They are either price-based, when variations in the energy price lead to an adaption of the energy consumption, or incentive-based, when the consumer is paid a fixed amount for the provision of a specific rate of load that can be used for DR [US D06; Alba07; Pale11; Vard15; Deng15].

The Association of German Engineers e.V. (VDI) guideline 5207 [VDI5207-1] defines 16 DR measures grouped into three categories based on the automation hierarchy, see Figure 2.2. The DR measures differ not only in their allocation to different hierarchical levels, but also in the time horizon in which they are implemented. Please note that the VDI guideline uses hierarchy level names that differ from the International Electrotechnical Commission (IEC) 62264-1 standard defined in [IEC62264-1], which is presented in Section 2.3. For consistency, this work uses the nomenclature according to [IEC62264-1] and classifies the DR measures accordingly.

In the following, the measures defined by VDI and visualised in Figure 2.2 are explained in more detail. The DR measures on business level are implemented long-term, with a time horizon of more than a week:

- *shift break times*: changing and rescheduling of the employees' break times in the factory
- *adjust shift times*: adjusting the production times
- *change production sequence*: changing the production order of entire batches
- *adjust capacity planning*: changing production resources in a long-term time horizon based on energy requirements, for example purchasing new production machines
- *shift production start*: long-term delay or early start of production

The DR measures which are executed in factory operations are located on the manufacturing level, the DR measures have a horizon between a few hours and a week:

- *shift start of job*: adapting the start of production jobs for individual production machines based on production scheduling
- *interrupt job*: short-term interruption of the job execution of individual production machines such that these can be shut down for a short time interval
- *adjust resource allocation*: consideration of the energy demand when selecting the processing production machine if several production machines can carry out the same processing operation
- *change job sequence*: adaption of factory production scheduling, for instance the job order in which the product is processed on different production machines, taking into account the energy consumption of the whole factory
- *store energy*: use of installed energy storage, such as chemical or thermal storages
- *change energy carrier*: Changing the energy carrier, for example electric heating instead of gas-powered heating, to influence the energy demand

The DR measures on control level, applied to single machines, have a duration of a few hours or less:

- *interrupt process*: short-term interruption of the production process on individual production machines or of machine modules during the execution of the process
- *adjust process parameters*: short-term adjustment of process parameters, for example modifying the hydraulic pressure of a cooling system
- *change processing sequence*: adapting the sequence of machining steps, for example changing the execution of milling and countersinking on a machine tool
- *store energy inherently*: utilisation of internal tolerance bands and storage systems, for example heating a hot water tank while considering the energy demand instead of only operating in a fixed temperature hysteresis
- *operate with bivalent energy*: using different energy forms in one production machine, for example variation between decentralised electrical heating and central thermal heating supplied by gas-boilers

This work focuses on the last group since these measures are executed at production machine level and the aim of this work is to implement a DR automation architecture on single production machine basis. The DRAAD method implements the DR measures *store energy inherently* and *interrupt process* as they are determined to be suitable for the application to APCMs, see Section 5.1.

Closely related to DR is the term energy-flexibility.

Definition 2.2 (Energy-flexibility) Energy-flexibility is defined as the ability of a production system to adapt quickly and in a process-efficient way to changes in the energy market [VDI5207-1].

This means that energy-flexibility is the ability of production systems in general and machines in particular to implement DR measures [Walt22]. There are three options for the economic use of energy-flexibility in industry [VDI5207-2; Stro21, p 34]:

1. provision of the energy-flexibility to be used directly by the grid operator, for example for improvement of power quality,
2. utilisation of fluctuating electricity prices and
3. in-house load management, for example to avoid peak loads or to increase the utilisation of self-generated (renewable) electricity.

In the first case, the energy-flexible production system is used for incentive-based DR, in the other cases it is used for price-based DR.

2.2 Demand response potential

Before methods for the analysis of the DR potential in industry can be discussed, the term DR potential must be clarified.

Definition 2.3 (DR potential) DR potential is divided into the following sets [Duft17; VDI5207-1]:

- The *theoretical potential* is the maximum theoretical DR potential based on the rated power of an electricity consumer.
- To determine the *technical potential*, a subset of the theoretical potential, safety, plant and process-relevant restrictions are taken into account.
- The *economic potential* is the subset of the technical potential that is economically viable.
- The *practical potential* is the subset of the technical potential, which considers business, regulatory and administrative constraints.
- The *realisable potential* is the intersection of the economic and the practical potential and is thus the potential which can be used in an economically sensible way while considering the company's restrictions.

Figure 2.3 visualizes the five DR potential sets. This work implements a DR potential analysis that quantifies the technical DR potential of APCMs to determine how the APCM's automation program should be implemented to use this DR potential, see Chapter 5. To analyse the technical DR potential the rated power value of the APCM is used.

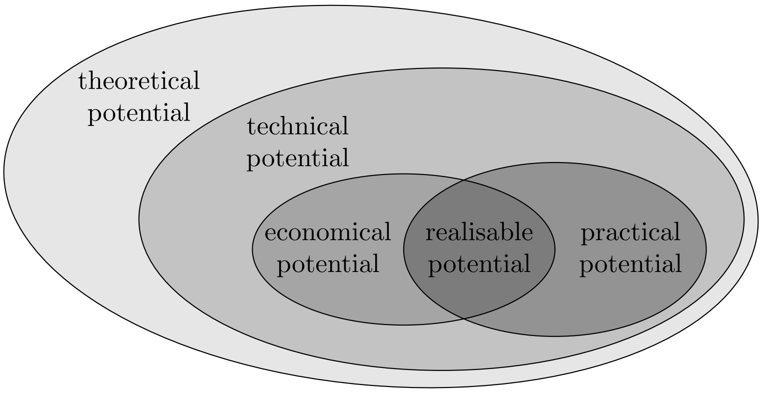


Figure 2.3: The different sets of DR potential. The economical and the practical potential are subsets of the technical potential which is a subset of the theoretical potential. The realisable potential is the intersection of the economical and the practical potential. Figure based on [Duft17; VDI5207-1].

Definition 2.4 (rated power) The rated power is the power demand of a machine under rated operation. It is indicated on its nameplate or in its technical documentation [VDI4661].

Usually the rated power is given as an active power value but can also be given as an apparent power value. If it is given as an apparent power value, for the DR potential analysis, it is used as if it was given as an active power value.

2.3 Transformation of industrial automation: cyber-physical production systems and service-oriented automation

The structure of this section follows [Fuhr23b], where we describe the transformation of industrial automation architectures from hierarchically structured to service-oriented. In the past, industrial automation has been organised in accordance with the automation pyramid defined in IEC 62264 [IEC62264-1], which is shown in Figure 2.4. The production process at the bottom of the pyramid (level 0) is influenced by actuators and sensors which are located on the field level (level 1). The actuators and sensors are connected to control systems such as PLCs to monitor and control the productions process. The control systems are connected to supervisory control and data acquisition (SCADA) systems. SCADA systems are the human machine interfaces (HMIs) used for process monitoring and supervision. Control systems and SCADA systems are both located on the control level (level 1). Since production machines include actuators, sensors and control systems, they are located on the field and control levels (levels 1 and 2). Production machines are linked to high level systems such as the manufacturing execution system (MES) and the enterprise resource planning (ERP) on the manufacturing level (level 3) and business level (level 4) via SCADA systems.

Classical automation structured by the automation pyramid is not intended to be used for DR as the automation's system boundary is limited to the company and an interaction with entities outside the company such as a DR market is not considered [Körn19]. Approaches exist to

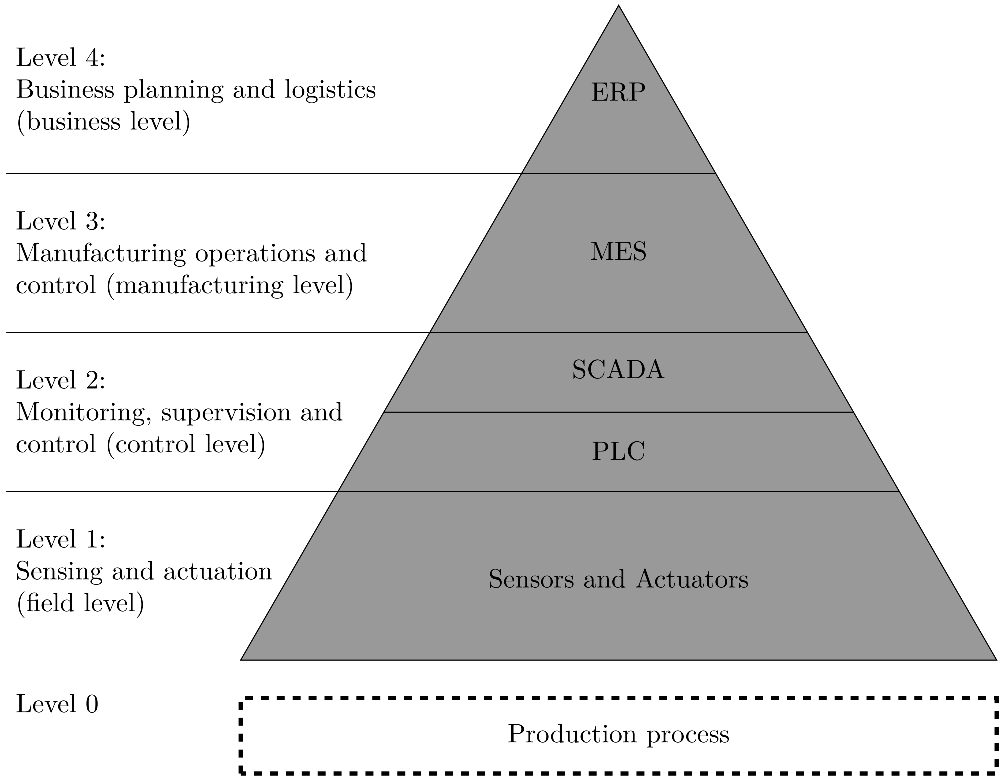


Figure 2.4: Classic industrial automation pyramid based on IEC 62264 [IEC62264-1]. Figure adapted from [Fuhr23b].

extend the automation pyramid with additional external layers [Körn19], but a simple extension of the hierarchical automation architecture is inconvenient as all layers of the automation pyramid have to be traversed for the execution of DR measures.

For use cases such as the execution of DR measures, new types of automation architectures exist. One option is modelling the automation architecture as a cyber-physical production system, such as that defined by The International Academy for Production Engineering (CIRP) in [Mono16; Mono18].

Definition 2.5 (Cyber-physical production system) Cyber-physical systems are computer systems which interact with each other and with the physical, mechanical world [Mono18]. In their application to production they are called cyber-physical production systems [Mono16].

Cyber-physical production systems consist of two functional layers, one high level layer where more complex functions are implemented, and one low level layer with functions responsible for real-time data processing and control of the physical machine. They are connected by the third layer, the communication layer, which enables the interaction between the digital (cyber) and physical part of the cyber-physical production system [Voge09; Mono16]. Figure 2.5 shows this design. High level applications such as ERP, SCADA, MES, energy monitoring or a DR control algorithm are located at the management and organisation layer. The PLC, sensors and actuators, which are all the automated parts of the production machine, are located on the field and control layer. Both layers are connected by the communication layer.

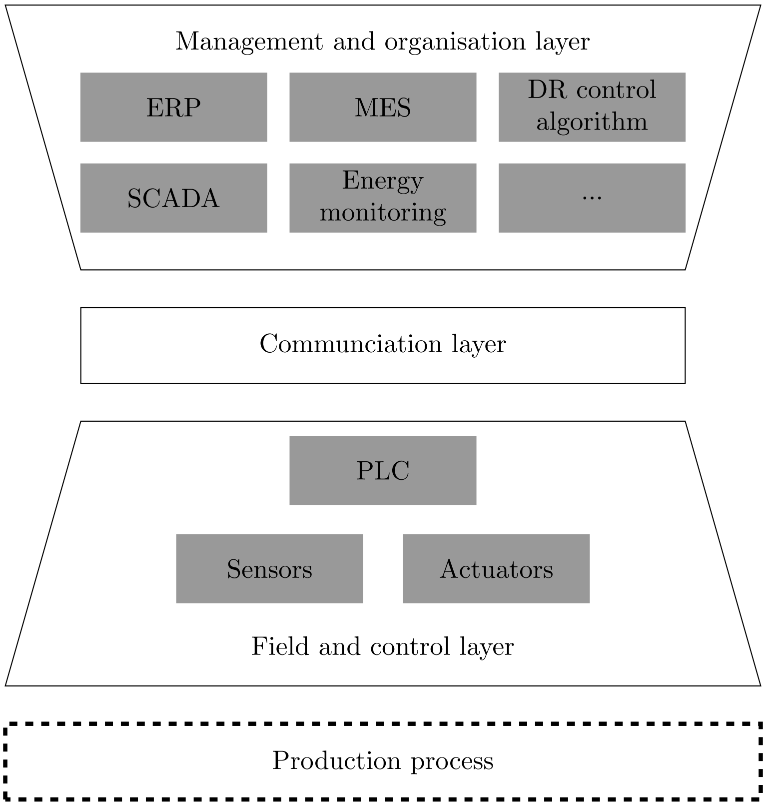


Figure 2.5: Automation architecture based on the idea of cyber-physical production systems. High level services such as ERP or DR control algorithms are located at the top at management and organisation layer which is connected to the field and control layer by the communication layer. The field and control layer includes sensors and actuators controlled by industrial control systems such as a PLC. Figure adapted from [Fuhr23b].

Closely related to cyber-physical production system is the term digital twin [Star18]. At the moment there is no commonly accepted definition for digital twin [Star18]. This work uses the definition proposed by CIRP.

Definition 2.6 (Digital twin) "A digital twin is a digital representation of an active unique product (real device, object, machine, service, or intangible asset) or unique product-service system (a system consisting of a product and a related service) that comprises its selected characteristics, properties, conditions, and behaviours by means of models, information, and data within a single or even across multiple life cycle phases." [Star18]

Following [Star18], a digital twin consists of

- a digital master that incorporates generalised description models of the product,
- a digital shadow which includes current live data of the operation and use of the product and
- a linkage between instances of the digital master and their digital shadows using algorithms, simulations or correlations for example.

Based on [Kohn21], this work calls a linkage a *digital service*, which is consistent with the terms digital master and digital shadow.

In [Voge09] the authors describe their architecture as a service-oriented architecture. Basic concepts of an interaction-based architecture as part of the Reference Architecture Model Industrie 4.0 (RAMI4.0) is presented in [DIN16593-1]. There, service-oriented architecture is described as a style that defines and restricts "roles, characteristics and allowed relationships of services and service consumers". In both cases, service-oriented architecture primarily describes services used for the communication between different systems such as OPC UA services [DIN16593-1].

This work presents a method for the design of a DR automation architecture that enables the execution of DR measures on APCMs. The focus of the design is not the development of a service-oriented architecture for communication. Instead, it relies on existing communication services and integrates them into the DR automation architecture. The Internet of Things (IoT) vocabulary defines an architecture as a "set of fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution" [ISOIEC20924]. Based on this definition, DR automation architecture is defined as follows.

Definition 2.7 (DR automation architecture) An automation architecture describes how industrial software and communication systems interact to enable the automated execution of specific control measures on production machines. If the automation architecture is used for the execution of DR measures it is called DR automation architecture.

In [DIN16593-1] a service is defined as a "mechanism to enable access to one or more capabilities where the access is provided by a prescribed interface and is exercised consistent [*sic*] with constraints and policies as specified by the service description". Following [ISOIEC20924], a service is a "distinct functionality that is provided by an entity through interfaces". The DR automation architecture in this work is used for the execution of a DR control algorithm which can be seen as a service.

Definition 2.8 (DR control algorithm) A DR control algorithm is a software program that implements an algorithm for the execution of DR measures. The algorithm decides when to execute which DR measure based on rules that include information from external entities, for example energy prices or the power consumption of the factory. The DR control algorithm interacts with the production machine based on the interfaces provided in the DR automation architecture.

The capability of the DR control algorithm is the execution of DR measures.

2.4 Interaction of IT and OT in production

This work discusses the interaction of IT- and OT-systems as part of the DR automation architecture. Different definitions of IT exist. In [Merr23] IT is defined as "the technology involving the development, maintenance, and use of computer systems, software, and networks for the processing and distribution of data". Following [Gart23a] IT is "the entire spectrum of technologies for information processing, including software, hardware, communications technologies and related services. In general, IT does not include embedded technologies that do not generate data for enterprise use." According to [Shil22], IT "is the hardware, software, and other information processing systems found in the enterprise or business information layer". The last definition includes aspects of industrial automation. This work combines these definitions and uses the following definition.

Definition 2.9 (Information technology) Information technology (IT) is the hardware, software, communications technologies and other systems used for the processing and distribution of data [Merr23; Gart23a]. IT-systems are located on the manufacturing and business level [Shil22] and management and organisation layer in the automation hierarchy respectively.

Although IT is used in OT-systems, the two areas differ in their purpose and their requirements [Karm18, p 14]. In [Gart23b] OT is defined as "hardware and software that detects or causes a change, through the direct monitoring and/or control of industrial equipment, assets, processes and events." This definition is used in scientific articles [Hahn16; Lara19], standardisation [ISOIEC30164] and by organisations such as the Industry Internet of Things Consortium (IIC) [Karm18]. [Hahn16] discusses similarities and differences of IT and OT and specifies that OT demands real-time operation. In [Shil22], OT is assigned to the control and field level of the automation pyramid. None of these definitions includes all aspects of OT. Again, the individual definitions are combined into one.

Definition 2.10 (Operating technology) Operating technology (OT) is the hardware, software and other information processing systems used for the monitoring and control of industrial equipment, assets, processes and events [Gart23b]. OT is located on the field and control level [Shil22] and in the field and control layer of the automation hierarchy respectively. To control physical industrial processes, OT-systems guarantee operation in hard real-time [Hahn16], which means that the OT-systems guarantee execution of their operations in a fixed cycle time.

Important OT-components are programmable logic controllers (PLCs). PLCs are electronic systems, that are able to control different kinds of machines and industrial processes using implemented digital control algorithms [IEC61131-1]. They have interfaces for the connection of sensors and actuators and are able to communicate with other digital entities such as other PLCs or computers.

In [IEC61131-3] five languages are described for programming a PLC. They can either be programmed graphically using ladder diagrams, function block diagrams and sequential function charts or they can be programmed text-based using structured text and instruction lists. Since 2013, [IEC61131-3] also includes features for object-oriented programming of PLCs. Object-oriented programming facilitates the reusability of program components and favours the rapid realisation of even large software frameworks [Broy19, p 398]. It enables scalability and transferability of automation programs. Object-orientation is already used in automation programs [Fran11; Li14].

For systems that enable an interaction between IT and OT this work uses the term IT-OT-data exchange service. Different communication systems exist that can be used for the communication between IT and OT. [Shil22] use OPC UA, Message Queuing Telemetry Transport (MQTT) and Representational State Transfer (REST)-based communication as IT-OT-data exchange service. [Viel18] name MQTT, Advanced Message Queuing Protocol (AMQP), Extensible Messaging and Presence Protocol (XMPP), Data Distribution Service (DDS), REST services and OPC UA as solutions for machine to machine communication. [Prof19] compare OPC UA, DDS, Robot Operating System (ROS) and MQTT and describe them as some of the most commonly used protocols in industrial automation. [Deve22] compare the communication standards OPC UA, MQTT, AMQP and REST-based communication and name OPC UA and MQTT as the most frequently used standards. Also, combinations of communication standards, such as OPC UA and DDS [Pfro16] or OPC UA and MQTT [Liu20] are proposed.

2.5 OPC UA

In the application of the DRAAD method, this work uses OPC UA as IT-OT-data exchange service. OPC UA is standardized in the IEC 62541 series [IEC62541-1]. It is a service-oriented architecture for machine to machine communication [Prof19]. The concept of an interaction-based service architecture using OPC UA services is illustrated in [DIN16593-1, pp 46–47].

OPC UA uses a server client communication system for the exchange of data [IEC62541-1, pp 20–25]. One system can include multiple OPC UA servers and clients simultaneously and an interaction between multiple servers and clients at the same time is possible. The client server interaction is a point-to-point interaction: The client requests specific data from the server, which delivers the requested data as a response [IEC62541-1, p 20]. In 2019 an additional publish-subscribe specification for OPC UA was introduced [IEC62541-14]. In this specification clients can subscribe to data published by servers [Prof19] such that real-time capabilities are increased by sending point-to-multipoint messages [Deve22].

While the other parts of the IEC 62541 series specify OPC UA such that it is independent of the technologies used for its implementation, IEC 62541-6 specifies the physical network protocols used to implement the OPC UA specifications [IEC62541-6, p 22]. The aim of OPC UA is to be as flexible as possible regarding future development of technology [IEC62541-6, p 22]. If other communication protocols emerge in the future they could be integrated in the OPC UA architecture and then used for the communication of OPC UA data.

OPC UA is based on existing communication middleware such as Transmission Control Protocol/Internet Protocol (TCP/IP) and WebSockets [IEC62541-6, p 78]. IEC 62541-6 names UA TCP, Hypertext Transfer Protocol Secure (HTTPS) and WebSocket as transport protocols [IEC62541-6, pp 78–91] and multiple data encodings such as OPC binary, OPC Extensible Markup Language (XML) and OPC JavaScript Object Notation (JSON) [IEC62541-6, pp 23–61]. By using different communication protocols OPC UA enables cross-platform communication [Prof19].

OPC UA integrates security by supporting an authentication via user name and password or by using digital certificates [Prof19]. By implementing OPC UA Field eXchange (OPC UA FX) for the application of OPC UA in real-time field level communication [OPC10000-80], OPC UA spans all layers of the automation hierarchy [Shil22].

One key aspect of OPC UA is its semantic information model [Prof19; Deve22]. The focus of OPC UA is to enable interoperability of different devices which is needed, for example, in a typical production line consisting of multiple machines and systems of different kinds [Prof19]. OPC UA data models include semantic annotations, such that clients consuming data automatically can derive the meaning of specific data values even without previous knowledge. In other communication protocols such as MQTT, ROS or DDS, clients need to know the topic of the data points in order to know which data points to subscribe to [Prof19].

For OPC UA, standardised data models, so-called OPC UA Companion Specifications, are developed by working groups. Employees of different manufacturers form a working group to enable standardised data exchange between systems of different manufacturers. The OPC UA Companion Specification *OPC UA for Machinery* defines common information for machines in general [OPC40001-1]. The OPC UA objects can be used to identify the machine and its components as well as to monitor the machine's behaviour [OPC40001-1, pp 21–22]. This can be used to construct a digital nameplate of the machine. Also OPC UA Companion Specifications for production machines exist for example the OPC UA Companion Specification *OPC UA for Machine Tools* [OPC40501-1] or the OPC UA Companion Specification *OPC UA for Robotics* [OPC40010-1]. Currently, no OPC UA Companion Specification for cleaning machines exists.

The OPC UA Companion Specification *OPC UA for Energy Management* defines important OPC UA data objects for energy management that must be provided by OPC UA servers that implement this OPC UA Companion Specification [OPC30141]. The OPC UA data objects are designed such that they can be used by OPC UA clients for different use cases, such as energy data visualisation and analysis or calculation of an emission footprint [OPC30141, pp 24–25]. The use cases also include DR measures such as general load management and avoidance of load peaks. The OPC UA Companion Specification only implements OPC UA data objects for energy measurement and standby management and states that these could be useful for the implementation of DR measures, but lacks a concrete implementation of DR measures such as *store energy inherently* and *interrupt process*. Section 6.3 presents the data model for these DR measures.

For the implementation of DR, an interaction between power grid operators and factory operators is needed. That means, that a data exchange between entities of the power grid and industrial entities such as production machines must be possible. The application of OPC UA in power grids is discussed in multiple research papers, for example [Claa11; Gil22; Wang22; Zhu22]. In the future, OPC UA could provide a seamless data exchange from entities of the power grid to production machines and thereby simplify the implementation of DR in industry.

In summary, OPC UA offers many features beyond the interaction of IT and OT. As a service-oriented architecture, it is not a simple communication protocol, but uses different communication protocols as part of its architecture. The main feature is the possibility to create semantic information models. In the future, OPC UA could simplify the integration of factories into the electricity market to implement DR.

2.6 Demand response control algorithms and optimisation models

In industrial automation, different control algorithms are used for the control of machines and processes. Commercial software libraries exist that provide typical controllers such as proportional–integral–derivative (PID) controllers or bang-bang controllers [Beck23c]. Even the integration of AI models directly into PLC automation programs is possible using specific libraries [Beck23b]. These libraries are vendor-specific as they are not standardised yet.

Algorithms developed in research, that implement DR measures, often use complex mathematical optimisation, such as mixed integer linear programming (MILP), heuristic algorithms or artificial intelligence (AI) approaches such as reinforcement learning [Abik17; Gros19; Webe18; Kohn20]. These are usually not implemented in automation programs at OT-level, but in high-level programming languages at IT-level. For the classification of different control algorithms based on different optimisation models, this work uses the terms black-, white-, and grey-box model.

Definition 2.11 (Black-, white-, and grey-box models) Black-box machine learning models are purely statistical models. White-box models, refer to analytical and physical descriptions whose modelling is usually very complex. Grey-box models combine both approaches in order to unite the respective advantages [Döbe18].

The DR control algorithm in this work uses MPC as a control algorithm, based on a grey-box approach, see Chapter 7.

MPC is a control method that is widely used such that fundamental descriptions can easily be found in literature [Rako19, pp 12–15; Cama07, pp 2–5; Ditt09, pp 38–44]. MPC predicts the future behaviour of a system based on the current system state and a time-discrete dynamic system model. The system is modelled using mathematical equations which are part of the formulation of an optimisation problem, usually a minimisation. Using this formulation, the optimal input of the system can be predicted by solving the optimisation problem. The optimisation problem includes time-discrete system states up to a certain point in time, the so-called prediction horizon. Only the predicted input value for the first time-step is used and the optimisation is repeated in intervals so that it is also possible to react to disturbances influencing the system. Limitations of the system are included in the system-model formulation as constraints of the optimisation problem.

2.7 Aqueous parts cleaning machines

In [DIN8580, p 12] cleaning as a manufacturing process is assigned to the main group *separation*. The standard [DIN8592, p 3] defines cleaning as the removal of unwanted substances (soil) from the surface of parts to a required, agreed or possible degree. [DIN8592, p 7] lists six subcategories of cleaning, including fluidic cleaning, chemical cleaning and solvent cleaning. Fluidic cleaning is defined as cleaning in which loosely adhering contaminants are removed by flowing gases or liquids [DIN8592, p 4]. It includes washing, blowing off, suction and ultrasonic cleaning. In industrial applications, these different cleaning methods are often combined with each other [Ange19, p 24]. This also applies to APCMs, hence, they cannot be assigned to just one of the defined cleaning methods [Jung19, p 8].

For a market survey, in [Bilz13, pp 20–21] the authors define five different groups of industrial cleaning methods in practical use: blast cleaning, mechanical cleaning, thermal cleaning, aqueous cleaning, special cleaning and other cleaning methods. Aqueous cleaning here mainly includes fluidic cleaning and combines these with chemical cleaning and solvent cleaning by using different cleaning liquids [Ange19, p 23]. As can be seen, no precise definition of APCMs exists, such that this work combines the definitions of [DIN8592; Bilz13; Ange19] as follows.

Definition 2.12 (Aqueous parts cleaning machine) An aqueous parts cleaning machine (APCM) is a production machine that combines fluidic cleaning with chemical cleaning and solvent cleaning to clean industrial parts by using aqueous cleaning liquids [DIN8592; Bilz13; Ange19].

Market surveys have shown that aqueous cleaning is the most used cleaning method in industry and the share is increasing. While in 2007 aqueous cleaning was used in 59 % of the companies surveyed, this share rose from 84 % in 2012 to 86 % in 2020 [Bilz13, p 21; Rögn21a, p 45]. The share of aqueous cleaning in the total turnover of cleaning machine manufacturers also increased from 58 % in 2007 to 68 % in 2012 and to 84 % in 2020 [Bilz13, p 21; Rögn21a, p 45]. Hence, when focussing on APCMs in this work, the main part of the industrial cleaning machines in use is considered.

Different kinds of APCMs are defined in [Ange19, pp 26–27]. Single-chamber APCMs have one cleaning chamber and usually are operated in a batch process. The machines can be equipped with only a single tank that stores the cleaning liquid or can have an additional tank storing a medium for rinsing. Multi-chamber APCMs have multiple cleaning chambers with multiple media tanks and are also used in a batch process but can clean multiple batches simultaneously. Aqueous throughput parts cleaning machines also exist. Here, the part passes through different cleaning zones by transportation on a conveyor.

The cleaning process in APCMs consists of three steps [Durk06, pp 2–3]:

1. Cleaning: separation of soil from the part by using an aqueous cleaning agent.
2. Rinsing: dilution or displacement of the emulsion consisting of the cleaning agent and the soil until all soil is removed.
3. Drying: separation of the part and the cleaning agent until the part is dry.

All these steps are necessary to obtain a high quality cleaning result [Durk06, p 3]. In simple cleaning systems without an extra rinsing system, as in the application example of this work, the cleaning and rinsing steps are combined into one step. The first phase of this process step comprises cleaning, that is the loosening of the soil, the second step then comprises rinsing, that is the further removal of the soil.

The empirical studies in [Sinn60] show that in aqueous cleaning of textile the four influencing factors temperature, chemistry, mechanics and time have a decisive impact on the cleaning result [Sinn60, pp 9–10]. If one factor is reduced, this must be compensated by an increase in at least one of the other three factors. [Ange19; Rögn21b] extend the four factors by the soil and the part to be cleaned and thereby adapt the approach to industrial cleaning. The cleaning liquid must be adapted depending on type, condition and quantity of the soil as well as the part's shape and material to reach a satisfying cleaning result. When applying DR measures to an APCM, it must be ensured that the cleaning result is not affected negatively.

2.8 Summary

This chapter introduced the most important terms used in this thesis and showed the current state of the art in the research area. The goal of this work is to develop a DR automation architecture that enables the execution of DR measures on APCMs. Therefore, first DR in general was described and then the current developments in industrial automation. The chapter explained the interaction of IT and OT in general and specified OPC UA as one technology used for this interaction. Black-, white- and grey-box models were introduced as a rating for optimisation models and the functionality of MPC which are used for the DR control algorithm in this work were described. At the end of the chapter an introduction to APCMs was given. Based on this, the research concept is defined in the following.

3 Research concept

The previous chapter describes the fundamentals of this work to show its research area and focus, which is the first part of the *Research Clarification*. In this chapter, in Section 3.1, the research objective is defined and research questions derived from this objective. Based on the research objective and questions Section 3.2 defines assumptions and requirements of this this work. The research objective, questions, assumptions and requirements form the second part of the *Research Clarification*.

3.1 Research objective and questions

In [Schr18] the author shows that it is possible to use production machines such as machine tools for DR by adapting their automation architecture and in [Stro20] we describe that cleaning machines have the potential to be used as inherent energy storages. Motivated by this research, this work's research objective is as follows.

Objective The research objective of this work is to enable the execution of DR measures on APCMs by developing a method for implementing a DR automation architecture.

To fulfil this objective it must be investigated, if APCMs have a DR potential and how this can be identified. This leads to the first research question.

Question 1 (DR potential analysis) How can the DR potential of APCMs be identified and quantified?

One part of the literature review in Section 4.2 is to investigate whether approaches already exist for analysing the DR potential of APCMs or whether approaches exist that can be adapted to APCMs. Since the literature review shows that no DR potential analysis exists for APCMs, a DR potential analysis for APCMs is developed in Chapter 5, extending and adapting existing methods for other production machines to APCMs. The DR potential analysis is the first step of the DRAAD method.

After identifying the DR potential of APCMs this potential must be made usable by adapting the APCM's automation architecture.

Question 2 (DR automation architecture) How must an automation architecture be designed to enable the execution of DR?

An automation architecture includes the actual automation program that controls the process and the modules of the APCM as well as the interaction of the APCM's automation program and external entities. Therefore, the second research question is divided into two sub-questions.

Question 2.1 (DR automation program) How must an automation program on machine level be designed to enable the execution of DR?

At field and control level the DR automation program must be able to control the APCM such that DR measures can be executed. The communication between the DR automation program, implemented in the machine automation system, and the DR control algorithm, implemented in an IT-framework, needs a DR data model located at the information layer.

Question 2.2 (DR data model) How must a DR data model be designed to enable the interaction of machine automation on OT-level and DR control algorithm on IT-level?

The literature review in Sections 4.1, 4.3 and 4.4 tries to find answers to these three questions. No comprehensive method for the implementation of a DR automation architecture exists that fulfils all requirements. Chapter 6 presents the design of the DR automation architecture as part of the DRAAD method.

Once the DR potential is identified and it is clear how the DR automation architecture needs to be built to exploit the potential, a service is needed that executes DR measures and finally exploits the potential. To do so, the last question must be answered.

Question 3 (DR control algorithm) How must an algorithm be designed to control the execution of DR measures?

The DR control algorithm is the last part of the DRAAD method. Existing DR control algorithms are analysed in Section 4.5 and the DR control algorithm is developed in Chapter 7.

3.2 Assumptions and requirements

The development of the DRAAD method is based on several assumptions. The first assumptions describe a typical development process of a production machine's automation system.

- A1 All machine modules and the cleaning process are controlled by the APCM's automation program.
- A2 The automation program of the APCM is usually developed by the machine manufacturer during the design and construction of the APCM.
- A3 The machine manufacturer has knowledge about the APCM, the cleaning process and how to develop an automation program, but only little knowledge of DR.

The DRAAD method must be designed such that the restricted knowledge of the machine manufacturer is considered. Even if the automation programme is developed in the design and construction phase, it can also be updated by the machine manufacturer to implement DR on existing APCMs, if the machine control fulfils the hardware requirements. The last assumptions consider DR.

- A4 The DR control algorithm is developed by an external developer who has knowledge of DR and mathematical modelling, but who has no knowledge of the design of automation programs.

A5 An external or internal demand for DR via a market or for internal load management exists.

A6 It is possible to communicate with the entities requesting the DR measures so that they can provide information to the DR control algorithm and that the DR control algorithm can access an electricity price.

The restricted knowledge of the developer must also be considered for the development of the DRAAD method and communication with external entities must be included.

In [Litt70] the author postulates requirements for the design of models that should assist managers in their decision-making process. Including these requirements into the design of the model should increase the likelihood that it will be used by managers and that they benefit from using it. The machine manufacturer and the developer are not managers, but since the requirements are designed such that everyone can handle the model, some of the requirements can be applied to other procedures. In [Pant19] the author develops a DR automation architecture for industrial supply systems and adapts the requirements of [Litt70] for the development of such a DR automation architecture. Based on [Litt70; Pant19, pp 80-81], the DRAAD method must respect the following requirements:

R1 The design of the DR automation architecture should be simple and easy to implement.

R2 Simultaneously, the design should be complete on important issues. This requirement contrasts with the first and a balance must be found between the two.

R3 The design should be adaptable to different kinds and numbers of APCMs.

R4 It must be robust, such that it can cope with errors during the execution of DR measures and system safety is guaranteed.

R5 The implemented DR automation architecture must be comprehensible and its behaviour easy to monitor.

In [Pant19, pp 81-82] the author assumes that the machines that his DR control algorithm interacts with are already able to execute DR measures. Therefore, he makes three assumptions regarding the machines ability to be observable, controllable and able to communicate. Since the aim of this work is to enable machines to execute DR measures, these three assumptions become requirements for the development of the DRAAD method. The requirements are as follows [Pant19, pp 81-82]:

R6 The APCM's system states that are important for the execution of DR measures must be observable.

R7 The setpoint values of the APCM that are used to execute DR measures must be controllable.

R8 It is possible to communicate information such as the aforementioned system states and setpoint values between the APCM and IT-systems.

Using the requirements and the assumptions in the following design criteria for the different parts of the DRAAD are presented, which specify the requirements in more detail. The design criteria were developed in iterations during the development of this thesis. After defining design criteria, they were integrated in the DRAAD method, the resulting design was evaluated and depending on the evaluations' outcome design criteria were adapted and additional design criteria added if needed. This results in the following 19 design criteria. For each design criterion, the requirement to which it relates is indicated in parenthesis.

The DRAAD method for the development of a DR automation architecture starts with the identification of the technical DR potential of APCMs. The design criteria for the DR potential analysis were already presented in our work [Fuhr23b]. They are as follows:

C1.1 The DR potential analysis is simple (R1) such that it can be applied by the machine manufacturer during design and construction of the APCM. Therefore, the analysis can be carried out at machine level only since during the design and construction phase no knowledge exists of the exact production environment the APCM will be operated in.

C1.2 The DR potential analysis is as simple and quick as possible and should need little additional knowledge to be applied by machine manufacturers (R1). It is executed manually, provides a quick preselection of machine modules that can be enabled for DR measures and does not need complex calculations, simulations or additional energy measurements.

C1.3 The DR potential analysis is complete (R2) such that the full technical DR potential is identified. It results in quantifiable values to allow a comparison of the DR potential of different machine modules.

C1.4 The DR potential analysis is adaptive (R3) which means it is scalable to APCMs of different sizes and transferable to different kinds of APCMs.

C1.5 The DR potential analysis considers the influence of the DR measure on the cleaning process and estimates its criticality to prevent negative influences on productivity and quality of the process results and to guarantee robustness and system safety (R4).

The main part of the DRAAD method is the development of a DR automation architecture including a DR automation programming scheme and a DR data model. These are integrated in the *automation data specification* described in Section 6.3. The design criteria for the DR automation architecture, DR automation program and DR data model extend the design criteria presented in our work [Fuhr23b]. The DR automation architecture must fulfil the following design criteria:

C2.1 The DR automation architecture implements the DR measures that are identified to be executable on APCM: *store energy inherently* and *interrupt process* (R2).

C2.2 The DR automation architecture is designed such that it is adaptable to different IT-architectures and can be deployed on different systems (R3). The IT-system can be integrated into the APCM itself if an industrial personal computer (IPC) is used that combines the PLC as an OT-system with a non-real time IT-operating system such as Windows or Linux. The IT-system may also be installed on a different computer connected to the OT-system. To cover all cases, the DR automation architecture is designed such that it is independent of the deployment architecture.

C2.3 The DR automation architecture is designed such that the states of the APCM that are needed for the execution of the DR measures are observable (R6).

C2.4 The DR automation architecture is designed such that the setpoints are controllable (R7) to enable the interaction of the APCM and the DR control algorithms.

C2.5 The DR automation architecture is designed such that a communication between different IT-systems is possible (R8). Since the DR control algorithm needs to interact with other IT systems, communication between different IT-systems is required.

For the DR automation program the design criteria are:

C2.6 The DR automation program is object-oriented to enable easy (R1) scalability and transferability (R3). The DR potential of a single APCM is constrained. Therefore, the DR automation program is object-oriented, such that it can be scaled for systems of various dimensions and can be transferred easily to other APCMs and a greater DR potential can be accessed.

C2.7 The DR automation program includes elements for functional safety (R4). Humans interacting with the APCM must not be harmed when executing DR measures.

For the DR data model the additional design criteria are:

C2.8 The DR data model is designed such that it enables an IT-OT-communication in the DR automation architecture (R8). It is reasonable to implement DR control algorithms on IT-systems since they usually include more complex optimisation algorithms that are hard to be integrated in standard automation programs. However, the APCM is controlled by an OT-system such that the DR data model must enable IT-OT-interaction to execute DR measures.

C2.9 The DR data model must include a strict naming convention for variables. Using this naming convention, developers can easily identify the variables in the DR data model (R1), which are needed for the execution of the DR control algorithm, and map them to the variables of the DR control algorithm.

The last part of the DRAAD method is the development of a DR control algorithm that interacts with the APCM to execute DR measures and that is implemented in the DR automation architecture. The development of the DR control algorithm considers the following design criteria:

C3.1 The DR control algorithm implements both DR measures, *store energy inherently* and *interrupt process*, to use as much of the determined DR potential as possible (R2).

C3.2 The DR control algorithm is scalable and transferable to different kinds and numbers of APCMs (R3) to access a greater DR potential.

C3.3 The DR control algorithm avoids a negative impact on process stability (R4). Therefore, it models the machine behaviour of the systems that are influenced by the execution of DR measures and enables the definition of critical states such that they can be avoided.

C3.4 The operation of the DR control algorithm is easy to monitor, which means the model uses a white-box or grey-box approach such that the behaviour of the integrated optimisation model is comprehensible (R5).

C3.5 The DR control algorithm incorporates the interaction with the APCM and a DR market (R8) by including the APCM's machine states and changing electricity prices.

This results in 19 concrete design criteria for the development of the DRAAD method and for its evaluation. Before presenting the development of the DRAAD method, the next section analyses existing research based on the defined design criteria. The review shows that existing research does not meet these design criteria and thereby identifies the existing research gap.

4 Existing industrial automation architectures for demand response: a literature review

This chapter analyses existing research based on a literature review and compares it to the approaches which were developed in the context of this thesis. The first step is a systematic literature review to identify existing industrial automation architectures for DR in Section 4.1. The DRAAD method, proposed in this work, includes a DR potential analysis, a DR automation programming scheme, DR data model and a DR control algorithm. Related research in these four areas is analysed in Section 4.2, 4.3, 4.4 and 4.5. Section 4.6 summarises the review and shows existing research gaps. The literature review also analyses to which extend the reviewed literature allready satisfies the DRAAD method's design criteria.

4.1 Systematic literature review

In the following, research related to this work is analysed in a systematic literature review. Based on [Pant19; Stro21], the three dimensions of the review are *objective*, *system boundary* and *approach*. The objective of this work is to enable the execution of DR measures on industrial APCMs by developing an automation architecture. Hence, the dimension *objective* is demand response, the *system boundary* is cleaning machine and the *approach* is automation architecture. Also, analogously used and closely related terms are considered, resulting in the following terms for the *objective*:

- demand response
- demand-side management
- energy-flexibility

To identify related work, the extended *system boundary* includes:

- cleaning machine
- production machine
- machine tool

An APCM is a production machine and machine tools have been a focus of research in production engineering in the past. The *approach* also includes related terms:

- automation architecture
- automation system
- automation program
- cyber-physical production system

For the systematic literature review the research data bases *ScienceDirect*, *Springer Link* and *Web of Science* are used which include scientific journals of major engineering communities such as the Institute of Electrical and Electronics Engineers (IEEE) for electrical engineering, the

Table 4.1: Results of the systematic literature review.

Combination	Number of articles
System boundary & approach	2324
Objective & approach	967
Objective & system boundary	311
Objective & system boundary & approach	29
Objective & *cleaning machine* & approach	7

International Federation of Automatic Control (IFAC) for automation and CIRP for production engineering.

Please note, that possibly individual articles are listed in several databases. Thus, by combining the terms and analysing the total number of research articles of all three data bases only a tendency can be determined with this analysis.

An overview of the analysis carried out last on 14.02.23 is depicted in Table 4.1. The highest number of results is archieved by a combination of the terms for *system boundary* and *approach*, which results in 2324 articles. This shows that a lot of approaches exist that consider automation of production machines in general. Less automation architecture approaches exist for DR. The combination of *objective* and *approach* results in 967 articles. DR on production machines is still reasonably analysed in science as the combination of *objective* and *system boundary* results in 311 articles.

A combination of all three dimensions results in only 29 articles. When choosing *cleaning machine* as system boundary only seven articles remain, of which six are developed by our research group Energy Technologies and Applications in Production (ETA) and four of them were created in the context of this thesis. The literature review shows that the research area of this work has not been frequently studied for production machines in general. Especially in the field of cleaning machines, the research potential is high.

The sheer number of articles provides an indication of the general relevance of the topic, meaning the research activities in this area. However, the numbers do not reflect the relevance in terms of content. Therefore, the titles and abstracts of the publications were analysed and checked for relevance. In addition, the four articles developed in the course of this work were sorted out. After this pre-selection, six papers remained for a detailed analysis. [Schu18] implement a DR potential analysis, which is compared to other methods in Section 4.2. [Abik17] present a DR control algorithm which is discussed in detail in Section 4.5. Automation architectures with a relation to energy are developed in [Seit19; Heut19; Vogt22; Pant18], which are discussed in the following. Since the systematic literature review resulted in only six articles, findings of a manual literature search are added.

[Seit19] describe an IT-based DR control architecture. The described software framework closes the gap between an external energy trading platform and individual factory components such as ERP, MES or SCADA systems. The authors assume that the machines are already enabled to execute DR measures such as *store energy inherently* or *interrupt process* and the article does not contain any information on the structure of an automation programme at machine level.

Table 4.2: Comparison of existing automation architectures for DR. Six of the nine analysed architectures include components needed for the implementation of DR measures on APCMs: a DR potential analysis, a DR automation programming scheme, a DR data model and a DR control algorithm.

	[Seit19]	[Heut19]	[Vogt22]	[Pant18]	[Pant19]	[Schr18]	[Gros22a]	[Fuhr23b]	[Fuhr23a]
DR potential analysis						x		x	
DR automation programming scheme				x	x	x		x	x
DR data model				x	x		x	x	x
DR control algorithm					x	x	x	x	x

In [Heut19] two concepts that integrate energy-efficient control into MES are presented. The article's literature review also analyses two papers that discuss DR, but the two concepts proposed in [Heut19] only include energy-efficiency and do not include DR. Also, the automation program of production machines is not part of the concepts.

[Vogt22] implement a cyber-physical production system for the evaluation of four different control schemes that control a heating, ventilation, and air conditioning (HVAC) system used in battery production with the aim of increasing the energy-efficiency of the production. The article focusses on the description of the control schemes and describes the automation framework only roughly. The cyber-physical production system includes the connection of sensors and actors using a PLC and an IT-OT-connection that can be implemented using different communication systems. How the communication can be implemented or how the automation program should be structured is not described. The article also does not consider DR.

[Pant18] develop an energy-efficient and energy-flexible automation framework for industrial supply systems. The approach includes an object-based programming scheme for building automation programs, a DR data structure for IT-OT-communication via OPC UA and functions for functional safety. The architecture provides the possibility for the integration of a DR control algorithm that could execute the DR measure *store energy inherently*, but a DR control algorithm is not presented in the article. [Pant19] extends this automation framework and implements a DR control algorithm.

[Schr18] presents an automation framework to implement DR on the auxiliary systems of machine tools. Modules are selected for DR based on the determination of their technical DR potential. The author adapts the automation system of the machine tool to apply a DR control algorithm but does not present a DR data model. The DRAAD method developed in this thesis was inspired by and extends the ideas of [Pant18; Schr18], which both were developed in the ETA research group.

In [Gros22a] we present a cyber-physical production system to execute DR measures on APCMs. We introduce a DR data model for *store energy inherently* and *interrupt process* and validated the set-up using a DR control algorithm.

In [Fuhr23b] we develop an automation program for DR based on an analysis of the technical DR potential of APCMs. The automation system includes a DR data model implemented in

OPC UA which is used for the execution of a simple DR control algorithm. We integrate a DR control algorithm based on MPC that implements *store energy inherently* and *interrupt process* and uses the previously deveoped automation architecture in [Fuhr23a].

The first three mentioned articles do not include DR potential analyses, DR automation programming schemes, DR data models nor DR control algorithms. These parts are only included in the research approaches which where developed in our research group, see Table 4.2. [Schr18; Pant18; Pant19] do not specify in detail how a DR automation program must be designed and do not show how it is implemented. Section 4.3 discusses these approaches. In the following, further literature is analysed in more detail and existing concepts on DR potential analysis, automation programming schemes, DR data models and DR control algorithms are examined, which were established as standalone ideas and not as part of a DR automation architecture. The analysis also evaluates if the existing research fulfils the design criteria defined in Section 3.2.

4.2 Methods for the analysis of industrial demand response potential

This thesis aims to implement a DR automation architecture that enables the utilisation of an APCM's DR potential. Therefore, the DR potential must be identified first. Different methods that determine the DR potential in industry exist. This section presents a literature review to identify existing approaches that analyse the DR potential following our previous literature review in [Fuhr23b].

Some of the existing methods analyse the DR potential only of entire factories. [Roth18; Roth19; Schu18] develop simulation models of the factory and estimate the economical DR potential of the entire factory using these simulation models. In [Löbe19] a preselection method is presented that uses key factors such as the rated power and time of use of machines to identify machines or auxiliary systems in the factory that could have a high technical DR potential and should be analysed in more detail.

Further approaches exist, that analyse the DR potential of factories and of single production machines. In [Graß14a] a mathematical model is used to identify the DR potential of production systems, which can be single production machines or the factory as a whole. A comprehensive method that can analyse the realisable DR potential of the whole factory, auxiliary systems and single machines is presented in [Tris20]. It analyses production units considering energy consumption measurements, rated power of machines, process specifications, the criticality of DR measures to the process and economical invests amongst others. [VDI5207-2] implements a practical method which includes forms that enable practitioners to determine the technical DR potential of their factory and individual machines based on preselection, comprehensive energy consumption measurements, process criticality and machine specific key factors.

All the aforementioned approaches can analyse factories as a whole and are therefore formulated in such a general way, that they do not consider the requirements of special production machines such as APCMs. This work focuses on analysing the DR potential of single machines, which allows a more specific DR potential analysis. [Graß14b] develop a DR potential analysis for production machines based on its machines states. Thereby, only the DR potential of process-related DR measures such as *interrupt job* is considered. [Graß15] develops a comprehensive

Table 4.3: Comparison of different methods for the DR potential analysis. The methods can be applied at factory or production machine level and determine the technical or economical DR potential. The methods are based on either simulations, mathematical modelling or use manual approaches and can be based on measurements. Some approaches assess how critical the execution of DR measures is to the production process.

	[Roth18]	[Roth19]	[Schu18]	[Graß14a]	[Löbe19]	[Tris20]	[VDI5207-2]	[Graß14b]	[Graß15]	[Rein14]	[Popp14]	[Abel16]	[Schr18]	[Stro20]	[Fuhr23b]
Factory level	x	x	x	x	x	x	x								
Machine level*				x		x	x	x	x	x	x	x	x	x	x
Technical potential*					x	x	x	x	x	x	x	x	x	x	x
Economical potential	x	x	x	x					x						
Simulation	x	x	x												
Mathematical model				x				x	x					x	
Manual execution*					x	x	x		x	x	x	x	x		x
No measurements*	x	x	x	x				x				x	x		x
Criticality*						x	x				x	x	x		x

*Design criteria for the DR potential analysis, see Section 3.2.

method to analyse the DR potential of production machines. The method combines a manual analysis of the machines with mathematical modelling and energy consumption measurements. In [Rein14], a multi-stage manual method for the identification of DR potential is presented. First, production machines are selected based on their rated power and share of total energy demand for further analysis. The technical potential of the selected machines is then identified based on measurements of electrical energy consumption. A preselection method is presented in [Lieb15]. Machines are analysed based on a questionnaire to determine their suitability for DR such that only promising machines are analysed in detail.

[Popp14] analyses the technical DR potential of machine modules for *store energy inherently*, taking into account the criticality of the DR measure to the process. The method analyses the energy storage capacity of the modules and how the modules are controlled, for example using bang-bang-controller in a hysteresis or continuously. Using these two criteria and a measurement of the activation time and mean electric power demand of the modules the DR potential of the machine is determined. The method in [Abel16] extends [Popp14] using parts of [Rein14] and [Lieb15] but does not need energy measurements to determine the technical DR potential of a machine. Instead, the activation time of machine modules is estimated by analysing the automation program code and the rated power is used instead of the mean electric power demand. In [Schr18, pp 59–71] the concept is slightly modified and the relevance of auxiliary modules for the main production process of machine tools is added as third dimension besides energy storage capacity and control mode.

Table 4.4: Only few automation programming schemes for DR exist.

	[Schr18]	[Pant18]	[Fuhr21]	[Fuhr22a]	[Fuhr23b]
Object-based		x	x		
Object-oriented*				x	x
Functional safety*		x	x	x	x
Observable*	x	x	x	x	x
Controllable*	x	x	x	x	x
Store energy inherently*	x	x	x	x	x
Interrupt process*					x

*Design criteria for the DR automation programming scheme, see Section 3.2.

In [Stro20] we present an approach to identify the technical DR potential of inherent energy storage systems in the factory based on energy measurements and mathematical modelling. We apply the method to an APCM amongst others.

The DR potential analysis used as part of the DRAAD method in this thesis has been published in [Fuhr23b]. The DR potential analysis extends [Abel16] to be applicable to APCMs and adds a potential analysis for *interrupt process*. It determines the technical DR potential at machine level, is executed manually and does not need energy measurements. Chapter 5 describes the method in detail. Table 4.3 shows a comparison of the different approaches to analyse the DR potential. Several of the features for comparing the different approaches are based on the design criteria for the DR potential analysis, which are defined in Section 3.2.

4.3 Automation programming schemes for demand response

Only few research papers describe how automation programs for DR should be structured, see Table 4.4. The author in [Schr18, p 102] only describes that read and write access to PLC variables must be established via OPC UA and if necessary integrates additional variables to implement *store energy inherently* on the machine. [Pant18] propose the usage of an object-based automation program to ensure scalability and transferability of the program. The authors also propose to implement functions for functional safety. The two approaches do not consider *interrupt process*.

Following the design suggestions of [Pant18], we develop an object-based automation scheme for building automation that includes functions for the secure execution of *store energy inherently* in [Fuhr21]. In [Fuhr22a] we extend our previous work and introduce an object-oriented building automation program including a strict naming convention for an easy use of the associated DR data model. We transfer the automation program from building automation to an APCM in [Fuhr23b] and add the functionality to execute *interrupt process*. Section 6.3 explains the approach as part of the DRAAD method in detail.

4.4 Demand response data models

Some approaches exist to describe DR in data models. Most of the models describe DR from the viewpoint of the electricity market. [Parv13] introduce four groups of DR measures: load curtailment, load shifting, on-site power generation, and utilizing electricity storage systems. The authors model the constraints of each group and implement a mathematical decision-making model for DR aggregator in the electricity market.

[Pete13] use the taxonomy *bucket*, *battery* and *bakery* to model energy-flexibility in the electricity market. Buckets are electricity consumers and producers, battery describes a process that can store energy and needs to be charged and bakery is used for a batch process with constant power consumption that must be finished in a given time period. The model includes a mathematical description of power states, energy levels and a capacity as well as a minimum runtime for every item.

[Bart18] use a general approach to model DR that includes multiple existing DR data models. The authors specify a list of characteristics to describe all potential varieties of DR. The data model is used as part of an optimisation.

Data models for an industrial approach also exist. [Scho19] implement a generic data model expanding the model of [Bart18] for industrial applications. The model defines three DR elements. Machines and process that can be used for DR are modelled as *flexible loads*. Dependencies between flexible loads are described in *dependencies* elements. Thermal, chemical and inherent energy storages as well as stocks for products in batch processes that can be used for DR are included as *storages*. The three elements have different characteristic values such as IDs, reaction and holding duration, regeneration duration, costs and usable capacity amongst others. The generic data model is used to offer the modelled, aggregated realisable DR potential of whole factories or industrial companies to an electricity market in a system consisting of two interacting IT-platforms [Sche18; Rösc19].

[Stro21, pp 94–103] implements a software framework to apply the DR measure *store energy inherently* in factories. The author identifies the DR potential of inherent energy storages and introduces characteristic values to describe these storages based on [Scho19]. The data model is then used in an optimisation. These data models have an industrial use case but were also developed from the perspective of the electricity market.

[Pant18] develop a DR data structure for IT-OT-communication via OPC UA without focussing on the commercialisation on the electricity market but on using the machines for DR in general. Individual HVAC facilities are mapped in the data model and aggregated into a data model of the entire factory's HVAC systems. The data model can be used for monitoring and control and implements the DR measure *store energy inherently*.

We extend this data model and present a DR data model for HVAC systems in [Fuhr22a]. While [Pant18] mainly explain the idea of the data model, we present the concrete programming scheme, including a strict naming convention and concrete DR safety and control modules and data structures. We transfer the data model to an APCM and integrate the DR measure *interrupt process* in [Fuhr23b]. We show the IT-OT-interaction of the data model in [Gros22a] where we describe the interaction of the data model with an DR control algorithm on IT-level.

This approach is described in detail in Section 6.3 and Section 6.4. The comparison of the different data models in Table 4.5 shows that only the data models of the research approaches

Table 4.5: A lot of DR data models focus on the energy market. This work concentrates on DR data models which can be used for industrial application.

	[Parv13]	[Pete13]	[Bart18]	[Scho19]	[Stro21]	[Pant18]	[Fuhr22a]	[Fuhr23b]	[Gros22a]
Focus electricity market	x	x	x	x					
Industrial application				x	x	x	x	x	x
Factory level				x		x	x		
Machine level					x	x	x	x	x
Observable*				x	x	x	x	x	x
Controlable*					x	x	x	x	x
IT-OT-communication*						x	x	x	x
Naming convention*							x	x	x
Store energy inherently*						x	x	x	x
Interrupt process*								x	x

*Design criteria for the DR data model, see Section 3.2.

developed in the frame of this thesis fulfil all design criteria of a DR data model as part of a DR automation architecture that enables the execution of DR measures on APCM.

4.5 Demand response control algorithms and optimisation models for demand response in industry

A lot of research articles exist that discuss and develop optimisation models for demand response in industry. [Biel16] analysed 89 papers in detail and identified 46 papers which developed models for production planning based on price-driven DR programs. The literature review in [Bäns21] resulted in 192 articles that include energy-aware optimisation models for production. Approximately 41 % of the articles included energy costs and energy consumption in the objective functions. [Brag23] analysed 53 articles in detail that developed algorithms implementing DR in industrial production planning.

Most of these approaches lack a defined method for applying the suggested energy optimisation on real machines in real production settings. They do not describe or use DR automation architectures with which the optimisation can be carried out on production plants and often they only validate the optimisation models in a simulation.

One example is the DR control algorithm identified in the systematic literature review in Section 4.1. In [Abik17] the authors present an integer programming model for production scheduling which includes electricity costs for the scheduling of multiple machine tools. The optimisation intends to maximise the usage of a photovoltaic power plant by interrupting the production process between different cutting operations. The optimisation is validated in a simulation including ten machine tools.

The objective of this thesis is to develop a DR automation architecture that enables the execution of different DR control algorithms. Therefore, the focus of the literature review in

Table 4.6: Most of the existing research on DR control algorithms does not describe how the included models should be applied to real machines, for example [Abik17]. This literature review focuses on papers that show the application to real production systems.

	[Abik17]	[Schr18]	[Pant19]	[Fuhr23b]	[Gros22a]	[Fuhr23a]
Factory			x			
Production machine*	x	x		x	x	x
Communication to external entities*	x		x	x	x	x
Functional safety*		x	x		x	x
Scalable*	x	x	x	x	x	x
Transferable*	x	x	x	x	x	x
Store energy inherently*		x	x	x	x	x
Interrupt process*	x					x
White box*	x	x		x	x	
Grey box*						x
Black box			x			
Simulation	x		x			
Real application		x		x	x	x

*Design criteria for the DR control algorithm, see Section 3.2.

this section lies on articles that do not only present DR control algorithms, but also include a description how these DR control algorithms can be applied to real production systems.

[Schr18, pp 85–93] develops a DR control algorithm based on a MILP model that implements *store energy inherently* on machine tools. The DR control algorithm controls the axillary modules of machine tools such that load peaks are reduced. The service only uses information generated by the machine tool such as sensor values and the machine state and does not interact with external entities. The DR control algorithm considers the inherent storage capacity of the modules and thereby guarantees a stable execution of the production process.

[Pant19] develops a DR control algorithm for industrial supply systems based on deep reinforcement learning. The DR control algorithm includes the DR measure *store energy inherently* and uses external energy prices. The author develops a comprehensive simulation library to enable the training of the black box AI models and then validates the trained AI using three different industrial supply systems in a simulation. The possible interaction of the DR control algorithm with real industrial supply systems via OPC UA is outlined but not validated [Pant19, pp 88–89, 175–178, 187–188]. The reward-function used to train the AI models includes aspects of functional safety since exceeding system limits such as minimum and maximum temperatures is punished [Pant19, pp 114–117].

In [Fuhr23b] we develop a DR control algorithm that executes *store energy inherently* on an APCM by controlling its cleaning liquid tank heater based on the electricity price. We use a rule-based control that turns the tank heater on if it is above a specific price limit and off if it is below a second price limit. We use the algorithm to apply *store energy inherently* and *interrupt process* in two separate field tests. In [Gros22a] we use a similar algorithm as DR control algorithm to apply *store energy inherently* and include functional safety by using a simulation model. We validate the algorithm's decision in a simulation of the tank system and check if the tank temperature stays inside its limits before applying the setpoint value on the real APCM.

The DR control algorithm for APCM which we implement in [Fuhr23a] combines *store energy inherently* and *interrupt process* using MPC based on a MILP optimisation model. We integrate critical system parameters such as the temperature limits of the cleaning liquid tank as constraints of the MILP model to ensure functional safety. It is a grey box approach since we use experimental data to parametrise the thermal model of the tank system. The DR control algorithm is presented in detail in Chapter 7 and its application on an APCM in Section 8.4. Table 4.6 compares the presented DR control algorithms. As can be seen in the last row only [Schr18] and the research approaches developed in the frame of this thesis validate DR control algorithm in real application. While in [Schr18] machine tools are analysed we investigate the application to APCMs.

4.6 Summary and research gap

The systematic literature review in this chapter showed that industrial automation architectures are already examined intensively in science, but there are only a few approaches for DR on production machines. When limiting the system boundary to cleaning machines almost no approaches exist despite the papers developed in the context of this thesis. In the chapter, existing research was analysed in detail for the four main elements of this work's concept: the DR potential analysis, the DR automation programming scheme, the DR data model and the DR control algorithm.

Several methods to determine the industrial DR potential exist, see Table 4.3, but only the DR potential analysis for machine tools in [Abel16] fulfils all of the design criteria. Chapter 5 extends and transfers this method to APCMs.

Almost no research exists on programming schemes for automation programs that implement DR measures, see Table 4.4. Although there are some DR data models for industrial application, no approach meets all design criteria, see Table 4.5. The approaches either only focus on the electricity market, are not usable for IT-OT-communication or do not include the DR measure *interrupt process*. The approach for a DR automation programming scheme and data model of [Pant18] is extended in Chapter 6.

Many DR optimisation models exist in research but most of them are validated in simulations and only a few are used as DR control algorithms integrated in DR automation architectures and applied to real production. In the review, the approaches implemented in DR automation architectures were analysed but none met all design criteria, see Table 4.6. Especially no DR control algorithm exists that combines *store energy inherently* and *interrupt process*, besides the one developed in relation to this work. The DR control algorithm of this thesis is presented in Chapter 7.

5 Demand response potential analysis of aqueous parts cleaning machines

Section 4.2 has shown that different methods to analyse the DR potential in industry already exist. The following chapter covers the developed approach for analysing DR potential of APCMs as the first step of the DRAAD method. Section 5.1 presents a summary of the design criteria for a DR potential analysis for APCMs defined in Section 3.2, followed by investigating the suitability of existing methods and preselecting DR measures for the DR potential analysis. Then, the DR potential analysis for the two selected DR measures *store energy inherently* and *interrupt process* is developed in Section 5.2 and Section 5.3, before summarising the chapter in Section 5.4. Figure 5.1 depicts a graphical overview of the method is depicted. The approach of this chapter has already been published in our work [Fuhr23b].

5.1 Design criteria for the demand response potential analysis and evaluation of existing approaches

We use the design criteria for the DR potential analysis defined in Section 3.2 to evaluate existing methods regarding whether and how they can be adjusted and applied to APCMs. The following list presents the summarised design criteria:

C1.1 The DR potential analysis is carried out at machine level only.

C1.2 The DR potential analysis is executed manually and does not need complex calculations, simulations or additional energy measurements.

C1.3 The DR potential analysis identifies and quantifies the full technical DR potential.

C1.4 The DR potential analysis is scalable and transferable.

C1.5 The DR potential analysis estimates the criticality of DR measures regarding their influence on the cleaning process result.

How the DR potential analysis fulfils these design criteria is summarised in Section 5.4.

We consider the VDI list of common DR measures presented in the VDI 5207 guideline as a basis for our DR potential analysis. The list is described in Section 2.1. The guideline assigns five DR measures to the control level. Since APCMs are located at the control level of the automation pyramid, see Section 2.3, we examine these five DR measures. The DR measures are [VDI5207-1]:

- interrupt process
- change processing sequence
- adjust process parameters
- operate with bivalent energy
- store energy inherently

Not all of these are applicable to an APCM. The possibility to change the processing sequence is limited since drying processes must be executed after cleaning processes. Hence, only the order of cleaning steps or drying steps could be changed among themselves. The customer using the APCM can modify the order of the APCM's cleaning process individually. To change the order of cleaning process steps, knowledge of the customer's individual cleaning process is needed which cannot be anticipated by the machine manufacturer in the design and construction phase of the APCM. Therefore, we do not consider the DR measure *change processing sequence* further.

An adjustment of the process parameters would be possible, but could affect the cleanliness or dryness of the part. To avoid such effects, we would need information regarding the customer's process quality requirements and execute a constant quality control to analyse if these quality requirements can be fulfilled. Hence, detailed knowledge of the customer's cleaning process

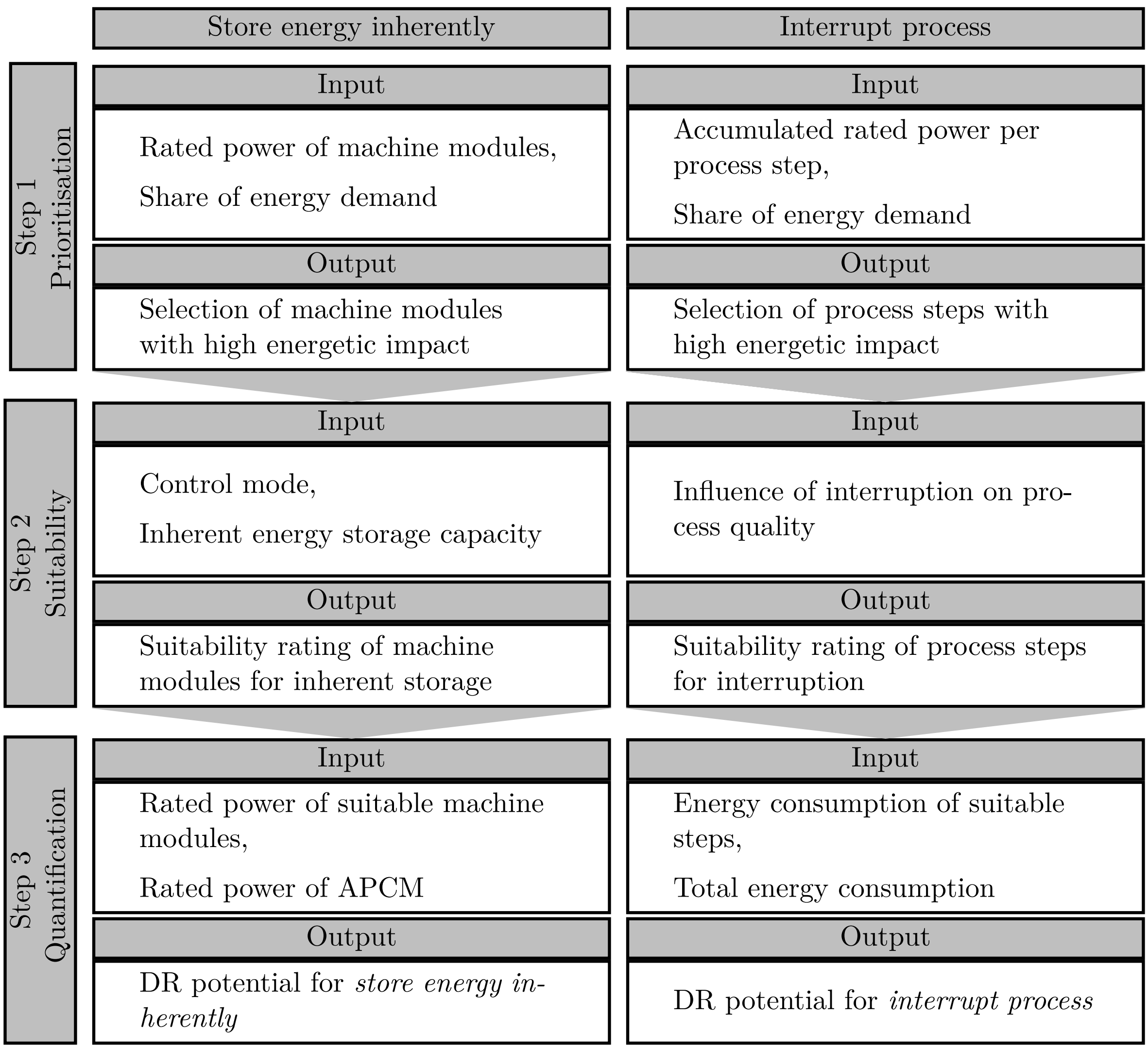


Figure 5.1: Graphical overview of the DR potential analysis for APCMs. The method is divided into the DR potential analysis of machine modules for *store energy inherently* (see Section 5.2) and the cleaning process for *interrupt process* (see Section 5.3). Figure adapted from [Fuhr23b], extending [Abel16].

is necessary. Since the machine manufacturer does not have this detailed knowledge, we also exclude *adjust process parameters*.

No data could be found indicating how many APCMs exist, that use multiple energy sources and can execute *operate with bivalent energy*. Hence, we focus on electrically powered APCMs. Consequently, the two DR measures *interrupt process* and *store energy inherently* remain for the DR potential analysis. Since a process interruption before the start of the cleaning process is possible, we include *shift start of job* when analysing *interrupt process*.

Most of the methods described in Section 4.2 are not suitable to be used for a preselection on machine level. These methods either only focus on factories and need complex simulation models [Roth18; Roth19; Schu18] or knowledge of the individual cleaning process to define specific DR measures prior to analysing a DR potential [Tris20]. Most of the methods are based on comprehensive energy consumption measurements since they are made for the DR potential analysis of existing machines already working in factories [Löbe19; Tris20; VDI5207-2; Graß15; Rein14; Popp14; Stro20].

We extend the approach of [Abel16] for the development of a DR potential analysis for APCMs. This approach determines the technical DR potential of machine tools for *store energy inherently* manually and does not need energy measurements. The approach uses parts of the work presented in [Popp14] as well as [Graß14b] and is partly similar to the VDI guideline [VDI5207-2] if applied only to production machines. We adapt [Abel16] to APCMs and add a DR potential analysis for *interrupt process*.

We divide our DR potential analysis into two parts. We first identify the APCM's technical DR potential for *store energy inherently* by analysing machine modules individually in Section 5.2. Then, we take a look at the cleaning process and determine the APCM's technical DR potential for *interrupt process* in Section 5.3.

5.2 Demand response potential analysis for store energy inherently

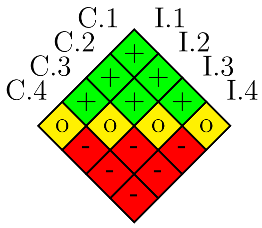
The DR potential analysis for *store energy inherently* adapts [Abel16] from machine tools to APCMs and consists of three steps. In the first step, we identify machine modules with a high energetic impact by identifying the rated power $P_f \in \mathbb{R}_{\geq 0}$ of every machine module $f \in \mathbb{N}$ individually and the total rated power of the APCM. Note that in the technical documentation, the rated power is usually given as an active power value, rarely as an apparent power value. In this work and in the DR potential analysis, we only consider the active power and require that the rated power is specified as an active power value or has been converted to an active power value.

In the case of an aqueous batch cleaning machine, we calculate the energy demand $W_f \in \mathbb{R}_{\geq 0}$ for every machine module $f = 1, \dots, F$ by

$$W_f = \sum_{g=1}^{G} P_f d_{f,g}, \tag{5.1}$$

where $F \in \mathbb{N}$ is the total number of machine modules, where $d_{f,g} \in \mathbb{R}_{\geq 0}$ is the duration the f-th machine module is switched on during the process step $g \in \mathbb{N}$ and $G \in \mathbb{N}$ is the total number of process steps.

Control mode
C.1: Nearly uncontrolled
C.2: Process-independent control in time
C.3: Process-dependent control in time
C.4: Entirely controlled in time and quantity



Inherent energy storage Capacity
I.1: Capacity covers more than one cycle
I.2: Capacity covers less than one cycle but uncritical process
I.3: Capacity covers less than one cycle and critical process
I.4: Direct energy conversion in real-time

Energy-process-independence

+	High degree of energy-process-independence	→	highly suitable for energy-flexibility
o	Medium degree of energy-process-independence	→	case-specific suitability for energy-flexibility
-	Low degree of energy-process-independence	→	not suitable for energy-flexibility

Figure 5.2: Assessment of the energy-process-independence of machine modules considered for DR measures. Figure taken from [Fuhr23b] based on [Abel16; Popp14].

In the case of an aqueous throughput parts cleaning machine, the machine modules are running all the time while the parts traverse the different process chambers. Here, we use the duration $d_{f,j} \in \mathbb{R}_{\geq 0}$ a part needs to traverse the process chamber of the corresponding machine module, instead of the duration a machine module is switched on during a process step. The duration a part traverses a process chamber represents the energetic importance of the machine module and therefore is used to calculate the energy demand. We calculate the energy demand by

$$W_f = \sum_{j=1}^{J} P_f d_{f,j}, \tag{5.2}$$

where $j \in \mathbb{N}$ is the index of the cleaning chamber and $J \in \mathbb{N}$ is the total number of cleaning chambers. We calculate the proportional share of energy demand $\varphi_f \in [0, 1]$ for every machine module f by

$$\varphi_f = \frac{W_f}{\sum_{f=1}^{F} W_f}. \tag{5.3}$$

Using the machine module's rated power P_f and its proportional share of energy demand φ_f we obtain two dimensions for a preselection of machine modules with a high DR potential. The user must set a threshold for each feature such that all machine modules with a percentage of the total rated power P_f and a minimum share of energy demand φ_f are selected. The thresholds should be selected so low that all machine modules are included that could have a high DR potential, but high enough to exclude small electrical consumers. As a good initial guess, the thresholds can be set between 5 % to 10 % of the APCM's rated power, or share of energy demand respectively.

In the second step, we determine the suitability of the preselected machine modules to be used for *store energy inherently*. We analyse whether and how much the use of the machine module for DR affects the cleaning process. Following [Popp14], we define the control mode and the inherent energy storage capacity as criteria to determine the energy-process-independence of each machine module. We summarize the energy-process-independence in Figure 5.2.

Using the *control mode* we describe how much the machine module's control depends on the state of the cleaning process. The machine modules can be nearly uncontrolled (C.1), such that they are controlled without any time or process dependence, for example infrequently switched on or off. They can also be controlled based on time intervals that do not correlate with the cleaning process (C.2), for example a temperature-controlled tank-heating system. The machine modules can be controlled on time intervals that depend on the active process step (C.3), such as a pulsing fan during the drying step. They can also be fully controlled in time and quantity (C.4), such as a frequency-controlled motor with varying speed which is PID-controlled.

With the *inherent energy storage capacity* we describe if and to what extent the system in which a machine module is operating in can store energy. For example, we take a tank heating system, that consists of the tank heater, which is the machine module, and a tank containing the cleaning liquid. We rate the system (I.1), if the inherent energy storage capacity is high enough such that the machine module does not have be turned on during one cleaning cycle, if the initial storage level is sufficient. In the case of an aqueous throughput parts cleaning machine, a cleaning cycle corresponds to the duration a part needs to traverse the process chamber that is to be rated. If the storage capacity is smaller, but the system's process value is uncritical for the process quality, we rate it (I.2), otherwise we rate it (I.3) if it is critical. If electric energy is converted in real-time such that the capacity is zero, the system is rated (I.4). An example for the last rating is a fan during a drying process.

The third step is the quantification of the DR potential. We define

$$\mathcal{F} := \{f \mid f\text{-th machine module is green or yellow}\} \tag{5.4}$$

as the index set of all machine modules to be considered for DR measures, which are those with a degree of energy-process-independence in green or yellow according to Figure 5.2. We determine the absolute achievable energy-flexible power demand $P_{\text{flex}} \in \mathbb{R}_{\geq 0}$ based on [Abel16] by calculating the sum over the rated powers P_f of these machine modules

$$P_{\text{flex}} = \sum_{f \in \mathcal{F}} P_f. \tag{5.5}$$

Using the total rated power of the APCM $P_{\text{total}} \in \mathbb{R}_{\geq 0}$, we calculate

$$\Phi_{\text{P}} = \frac{P_{\text{flex}}}{P_{\text{total}}}, \tag{5.6}$$

where $\Phi_{\text{P}} \in \mathbb{R}_{\geq 0}$ is the ratio of power that can be used for DR and which is consequently the technical DR potential [Abel16].

5.3 Demand response potential analysis for interrupt process

To analyse the APCM's DR potential for *interrupt process* we need a reference cleaning process. We assume that manufacturers of APCMs use a reference cleaning process for designing and setting up their APCMs. A reference cleaning process consists of different cleaning process steps such as *spray cleaning* or *convection drying*. In the following we use this process to analyse the DR potential. In the first step of the DR potential analysis for *interrupt process*, we determine the accumulated rated power and the share of the total energy demand for every process step, similarly to the first step of the DR potential analysis for *store energy inherently*. We define

$$\mathcal{A}_g := \{f \mid f\text{-th machine module is active in the process step } g\} \tag{5.7}$$

as the index set of all machine modules active in the process step g. We calculate the accumulated rated power $P_g \in \mathbb{R}_{\geq 0}$ for the step g by

$$P_g = \sum_{f \in \mathcal{A}_g} P_f \tag{5.8}$$

using the rated power P_f of all machine modules active in the process step. We estimate the energy consumption $W_g \in \mathbb{R}_{\geq 0}$ per process step g by

$$W_g = \sum_{f \in \mathcal{A}_g} P_f d_{f,g}, \tag{5.9}$$

considering the rated power P_f of the machine modules active in the process step and their programmed activation duration $d_{f,g} \in \mathbb{R}_{\geq 0}$. Analogously to the calculation for each machine module in (5.3), we calculate the proportional share of energy demand $\varphi_g \in [0, 1]$ for every step g by

$$\varphi_g = \frac{W_g}{\sum_{g=1}^{G} W_g}, \tag{5.10}$$

in order to compare the process steps with each other. Similarly to the DR potential analysis for *store energy inherently*, we select process steps with a high energy impact for the second step of the DR potential analysis based on the accumulated rated power P_g and the share of the energy demand φ_g. Therefore, the user must select suitable thresholds for both criteria to rank process steps regarding their energy impact. The thresholds can be chosen analogously to the procedure during the DR potential analysis for *store energy inherently*. Then, we execute the second and third step of the DR potential analysis only for these selected machine modules.

In the second step of the DR potential analysis, we asses whether and where a cleaning process can be interrupted. We can interrupt the process at different stages and the measure *interrupt process* defined in [VDI5207-1] can be expressed as

- shift start of job (interruption before the first process step)
- interruption between process steps
- interrupt a running process step

We summarise the first two as interruption before the beginning of a process step.

To fulfil design criterion C1.5, the DR potential analysis must consider influences of the DR measures on the process result, meaning the cleanliness and dryness of the work piece. Since there is not much scientific research on whether and where cleaning processes can be interrupted, case-by-case assessments by the machine manufacturer are necessary. Following [Auri09], delaying the start of the cleaning process has no negative impact on the work piece's cleanliness which makes an interruption before the start of the cleaning process usually feasible. As far as we know, if or how an interruption during the cleaning or drying process affects the cleanliness or dryness of the work piece was not part of scientific research yet.

A process interruption is more difficult in the case of an aqueous throughput parts cleaning machine than in the case of an aqueous batch parts cleaning machine. If the APCM does not

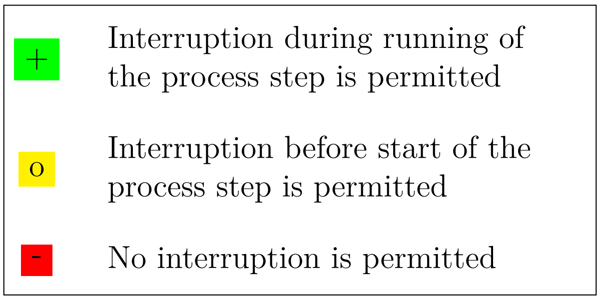


Figure 5.3: The rating of a process step's DR suitability is based on three levels that indicate whether and when the cleaning step may be interrupted. Figure taken from [Fuhr23b].

have buffer storages between process steps, an interruption of one step signifies an interruption of the whole APCM or could delay adjoining processes. Our DR potential analysis could be used to determine if and where buffer storages could be integrated in the APCM. However, this would mean adapting the machine design, which is beyond the scope of this work and is therefore not considered further.

We use the three categories shown in Figure 5.3, to rate the influence of a process step interruption on the process quality. If a process step can be interrupted during its execution, the step is rated green. If it is only possible to interrupt before the start of the process step, we rate it yellow and if no interruption is permitted red.

We quantify the DR potential for *interrupt process* in the third step. Similar to the DR potential analysis of *store energy inherently* we consider all process steps with green and yellow rating. We define

$$\mathcal{G} := \{g \mid g\text{-th step is green or yellow}\} \tag{5.11}$$

as the index set of all steps with green or yellow rating. Similar to the achievable energy-flexible power demand P_{flex} we calculate the absolute energy DR potential $W_{\text{flex}} \in \mathbb{R}_{\geq 0}$ for *interrupt process* by

$$W_{\text{flex}} = \sum_{g \in \mathcal{G}} W_g. \tag{5.12}$$

Using the total energy consumption $W_{\text{total}} \in \mathbb{R}_{\geq 0}$, which is defined by

$$W_{\text{total}} = \sum_{g=1}^{G} W_g, \tag{5.13}$$

we can calculate the ratio $\Phi_{\text{W}} \in [0, 1]$ of the energy that can be used for DR

$$\Phi_{\text{W}} = \frac{W_{\text{flex}}}{W_{\text{total}}}. \tag{5.14}$$

For the measures *store energy inherently* and *interrupt process* our DR potential analysis yields a total and relative quantifiable DR power and energy value. As a result, it is possible to compare the potential of different APCMs and express the technical DR potential.

The absolute technical DR potential for *store energy inherently* P_{flex} can only be achieved if all chosen machine modules are simultaneously used. Whether or not this is possible depends on the particular cleaning process. In the case of batch cleaning for instance, the absolute technical DR potential W_{flex} for *interrupt process* only represents the DR potential prior to the execution of the first energy-flexible process step. Only the potential of the remaining energy-flexible steps is available after the first energy-flexible process step has been completed. To precisely determine the practical DR potential, meaning the power and energy that can be used for DR, it is also necessary to measure the APCM's specific electrical consumption, as can be seen in the field test in Section 8.5.

5.4 Summary

This chapter introduced the DR potential analysis for APCMs as first step of the DRAAD method. The DR potential analysis is suitable for the preselection of machine modules that might be used for *store energy inherently* as well as process steps that have a high DR potential for *interrupt process*.

The DR potential analysis satisfies the design criteria outlined in Section 5.1. It only considers the DR measures *store energy inherently* and *interrupt process* that are located at machine level as it adopts the view of a machine manufacturer (C1.1). The DR potential analysis is a method that is easy to use for machine manufacturers (C1.2), since it only needs information about the APCM that is easily accessible by analysing the documentation and the automation program of the APCM as well as a structured classification of the machine modules and process steps. The method analyses electrical consumers and results in quantifiable values for power and energy that can be used for DR measures (C1.3). The DR potential analysis can be applied to different kinds of APCMs such as aqueous batch part cleaning machines and aqueous throughput part cleaning machines (C1.4). It includes the influence of the DR measures on the cleaning process by integrating the control mode for *store energy inherently* and the influence of a process interruption on the process quality (C1.5).

Details regarding the APCM's temporal availability are needed in order to calculate and implement DR measures. These details must be supplied by external energy control systems or production planning systems that have knowledge of the characteristics of the APCM's application setting at the manufacturing facility. For the DR measures to be carried out by the external control systems, the preselected machine modules and process must be made accessible for external control. In the following, the DR automation architecture is presented that enables the utilisation of the determined DR potential for the execution of DR measures.

6 Demand response automation architecture

After analysing the APCM's DR potential, this chapter shows the second part of the DRAAD method, the DR automation architecture. To execute DR control algorithms it is not only necessary to develop an automation program that implements DR measures, also a DR automation architecture must be established that enables the interaction of the DR control algorithm as an IT-system with the machine automation, the PLC, as an OT-system. The main parts of the DR automation architecture are the DR automation program and the DR data model, which enable the control of the APCM by the DR control algorithm.

In this chapter, Section 6.1 defines the requirements for the DR automation architecture. Then, a cyber-physical production system framework is introduced as the structure for the DR automation architecture in Section 6.2. The cyber-physical production system includes a digital twin representing the components of the APCM relevant for DR. This digital twin includes the automation data specification that specifies how to construct the DR automation program. Section 6.3 describes the object-oriented structure of the DR automation program and the resulting DR data model. The second element of the digital twin, described in Section 6.4, is the automation data dictionary which is used for IT-OT-communication between the APCM and the DR control algorithm. The chapter terminates with a summary in Section 6.5.

The contents of this chapter were already published in [Fuhr21; Fuhr22a; Fuhr23b; Gros22a], which is highlighted in the individual sections.

6.1 Design criteria for the demand response automation architecture

The DR automation architecture is based on the automation diabolo, a cyber-physical production system introduced in [Voge09], as described in Section 2.3. It consists of the field and control layer as well as the management and organisation layer which are connected by a communication layer, see Figure 2.5. The DR automation architecture must fulfil the design criteria defined in Section 3.2 to execute DR measures on APCMs. The following list presents the summarised design criteria:

C2.1 The DR automation architecture implements the DR measures *store energy inherently* and *interrupt process*.

C2.2 The DR automation architecture is adaptable to different IT-architectures and can be deployed on different IT-systems.

C2.3 The DR automation architecture is designed such that the machine states are observable.

C2.4 The DR automation architecture is designed such that the setpoints are controllable.

C2.5 The DR automation architecture is designed such that a communication between different IT-systems is possible.

C2.6 The DR automation program is object-oriented to enable scalability and transferability.

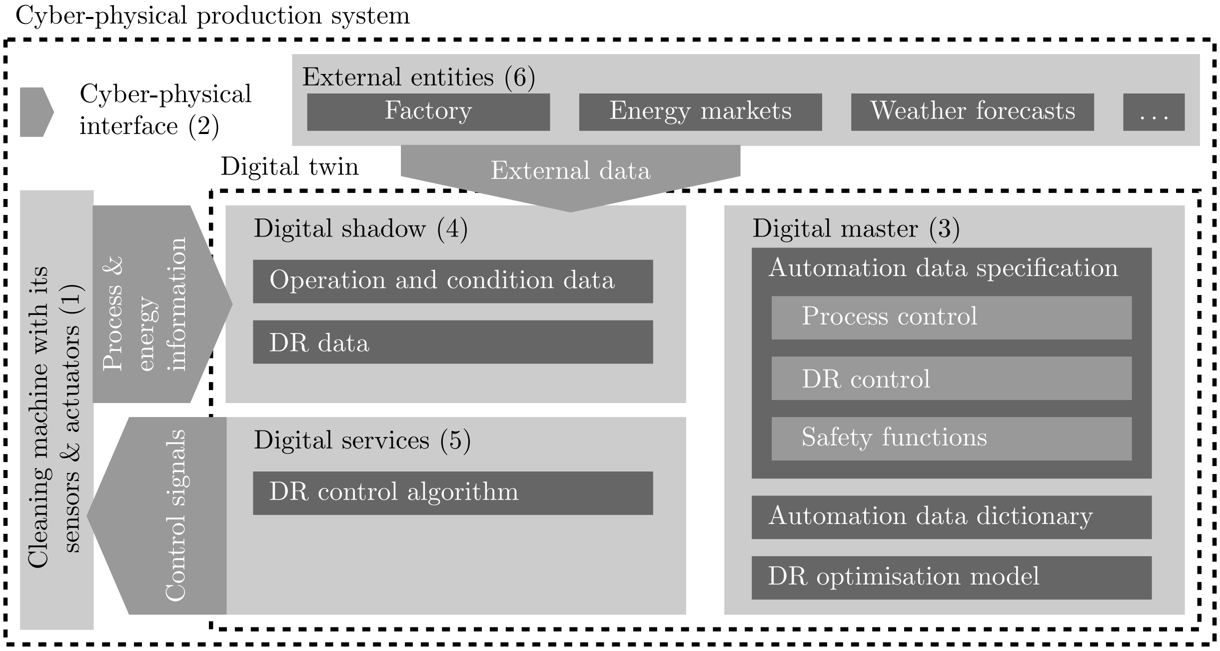


Figure 6.1: Structure of the cyber-physical production system for DR in the operational life cycle phase. The cyber-physical production system consists of a digital twin which interacts with the physical APCM and external entities. Figure adapted from [Gros22a].

C2.7 The DR automation program includes elements for functional safety.

C2.8 The DR data model enables IT-OT-communication.

C2.9 The DR data model includes a naming convention for variables.

The DR automation architecture is designed considering these design criteria. Section 6.5 summarises how the design fulfils the design criteria.

6.2 Cyber-physical production system aqueous parts cleaning machine

The APCM, its automation system and its digital representation can be described as a cyber-physical production system. The cyber-physical production system includes a digital twin for the APCM's digital representation which includes all elements necessary to execute DR measures. The cyber-physical production system follows our publication [Gros22a] and is visualised in Figure 6.1. It consists of four main elements:

- The physical APCM including its actuators and sensors (1),
- a cyber-physical interface (2) that represents the interaction between the physical APCM and the digital twin as well as the interaction of the digital twin and external entities,
- the digital twin which consists of a digital master (3), a digital shadow (4) and digital services (5) and
- external entities (6) needed to execute DR measures.

The cyber-physical production system focuses on the design and operational life cycle phases and uses the CIRP definitions for cyber-physical production system [Mono18] and digital twin [Star18]. It extends the digital twin concept presented in [Kohn21]. For a detailed explanation of cyber-physical production systems see Section 2.3.

The APCM (1) with its actuators and sensors is the physical part of the cyber-physical production system. It is connected to the digital twin by the cyber-physical interface (2). The cyber-physical interface represents the data exchange in the cyber-physical production system. This includes the IT-OT-communication between the physical APCM and the digital twin as well as the data exchange between the digital twin and external entities.

The first part of the digital twin is the digital master (3) which is developed in the APCM's design phase and is the same for all instances of the APCM. The digital master includes the automation data specification, the automation data dictionary and the DR optimisation model.

The automation data specification specifies three programming elements that must be included in the machine automation program to enable the execution of DR measures.

- The *process control* includes the standard machine control procedures that are needed to run the cleaning processes.
- The *demand response control* consists of the control procedures and data models for *store energy inherently* and *interrupt process*.
- The *safety functions* ensure safe operation of the APCM during the execution of DR measures.

These functionalities must be included in the automation program and interact with each other to enable the execution of DR measures. Section 6.3.1 describes the implementation of the automation data specification in the APCM's automation program.

The automation data dictionary maps the variables of the automation data specification from OT-format to IT-format such that it can be used directly by the DR control algorithm. This is described in Section 6.4.

The DR optimisation model models the energy costs of the APCM based on its process related energy consumption. It is part of the DR control algorithm, which uses the instantiated DR optimisation model to predict how the execution of DR measures changes the APCM's energy consumption and thereby can reduce the energy costs.

The digital shadow (4) includes operation and condition data generated in process control and DR data. As this is live data, the digital shadow is different for each instance of the digital twin.

The DR control algorithm as a digital service (5) uses the data in the digital shadow to parametrise the DR optimisation model to the instance of an APCM and to calculate and execute DR measures. Chapter 7 describes the DR control algorithm and the DR optimisation model it uses in detail.

The cyber-physical production system also includes external entities (6). These are entities needed to execute DR measures on the APCM, for example other production machines inside the factory, if the APCM is used for internal peak load shifting. If the DR measures are used to adapt the machine operation to local renewable energy production, external entities would be the energy market and the weather forecast.

6.3 Automation data specification

The automation data specification specifies how the automation program should be structured and which information must be included in the communication layer such that DR measures can be executed. The object-oriented DR automation program, extends our previous work [Fuhr22a] which in turn extends and implements the concept of [Pant18]. In [Fuhr22a], we presented an object-oriented design for an automation program to execute the DR measure *store energy inherently* which we implemented in a building automation system using the example of an air-cooled chiller. In the following, this design is transferred to APCMs and extended to be able to execute *interrupt process* as well.

The design described in Section 6.3.1 consists of data objects for actuators and sensors as well as systems that combine multiple data objects. If these data objects are used for DR, they include the safety functions presented in [Fuhr21] as well as energy-flexibility functions and data structures for *store energy inherently* and *interrupt process*, taken from [Fuhr22a; Fuhr23b]. The safety functions are explained in detail in Section 6.3.2, Section 6.3.3 specifies the functions and data structures for *store energy inherently* and Section 6.3.4 describes the functions and data structures for *interrupt process*.

6.3.1 Object-oriented demand response automation program

The automation program is object-oriented to enable scalability and transferability of the program [Broy19, p 398]. Additionally, the structure of the resulting program code is easier to understand for developers that may want to understand, reuse or extend the automation program. Figure 6.2 shows the base classes of the data model as well as the methods and data objects used for DR. Most of the classes are used to implement *store energy inherently*. System objects that implement the interface `InterruptProcess` are used for *interrupt process*.

Four classes represent actuators and sensors which are the basic field devices in automation [Fuhr22a]:

- The `Actuator` class represents actuators that are not used for DR.
- The `Sensor` class is the base class for sensors such as pressure or temperature sensors.
- The `Actuator2Point` class represent an actuators which are used for *store energy inherently* and can take two discrete states for example on and off. These actuators are usually connected to a digital output of the APCM's PLC for example control valves or a heating rods.
- The `ActuatorContinuous` class represents actuators for *store energy inherently* which are operated in continuously modulated control. These are usually connected to the PLC by an analogue output or by a fieldbus. Examples for these actuators are speed-controlled pumps or motors.

All these classes should be used to implement the process control by adding individual methods and objects adapted to the specific APCM. The `ActuatorContinuous` class extends the class `Actuator2Point`, inherits the basic DR safety function `SelectControlMode()` and control structures and adds methods to set a continuous setpoint (see Section 6.3.2). The `Actuator` and `Sensor` base classes do not specify any DR attributes or methods since they are not used directly for DR but can be part of a system used for DR.

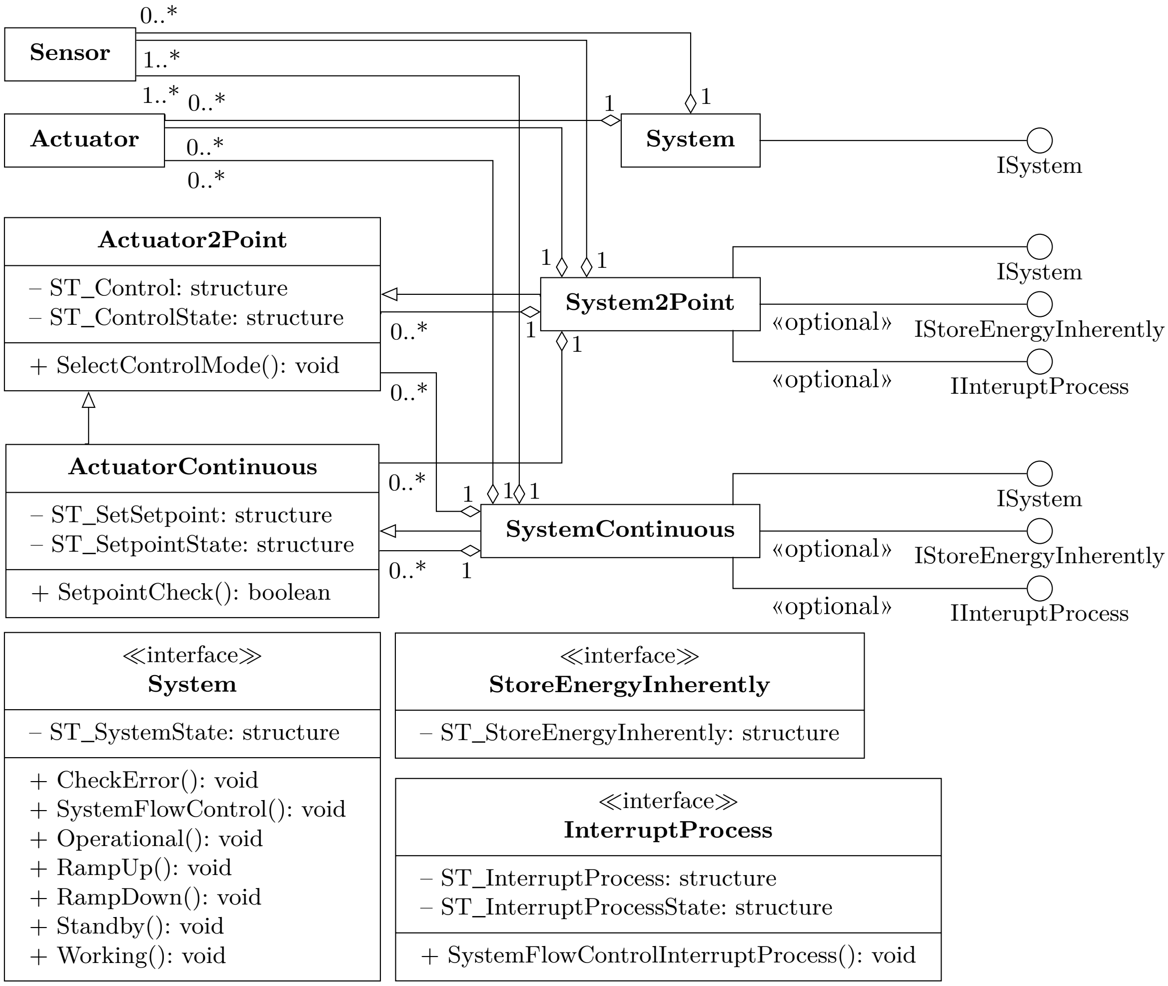


Figure 6.2: Class diagram which presents the base classes of the object-oriented data model. Actuators are modelled as `Actuator2Point`, if they are controlled in two states, or `ActuatorContinuous`, if they are controlled continuously. Multiple actuators and sensors are grouped in systems. These systems all implement the basic `System` interface. If the systems are used for *store energy inherently* or *interrupt process* they must implement the corresponding interfaces as well. Figure based on [Fuhr22a].

Actuators and sensors are grouped to systems. It is reasonable to group logical components of the APCM to systems such as cleaning chambers or heating systems. An APCM can consist of multiple systems which, grouped together as a system of systems, represent the APCM as a whole. There are three kinds of system classes [Fuhr22a]:

- The basic `System` represents a system that cannot be used for DR. It can also be a system containing only `Sensor` and no `Actuator` objects.
- Systems that are used for DR and that can be set to two states, such as a heating system which is controlled in a hysteresis, are modelled as `System2Point`. The system inherits from `Actuator2Point` such that it can be controlled and used for *store energy inherently*.
- Systems that are used for DR and are set to specific setpoints in continual control are modelled as `SystemContinuous`. An example would be a heating system that can be

Table 6.1: Variables included in the ST_SystemState structure. For each variable its static or dynamic type and access mode are described.

Data structure / variable	Type*	Access*
ST_SystemState		
System operating state	d	r
Error state	d	r

* s - static, d - dynamic, r - read, w - write

set to a specific temperature. This system inherits from ActuatorContinuous and *store energy inherently* can be applied similarly.

All systems must implement the interface System for process control. The interface includes the SystemFlowControl method whereby an interaction of different actuators depending on the systems state can be added. The current system state is displayed in the ST_SystemState structure. It also includes a variable that indicates if a system is in an error state and cannot be used for DR, see Table 6.1. Based on the machine states defined in [VDMA34179], systems can take on five states [Fuhr22a]:

- In *stand-by*, all actuators of the system are turned off.
- During *ramp-up* actuators are started to prepare for the process. For a tank heating system as an example, ramp-up signifies the initial heating of the cleaning liquid.
- This state is terminated when the *operational* state is reached. Here, the system is kept ready to be set to the next state.
- The next state is *working*, where the actuators of the system are controlled according to their process utility, for example the APCM's liquid pump is activated during spray cleaning.
- In *ramp-down*, the controlled, chronological shut-down of actuators in the system is implemented. An example would be to stop the heater and the fan before closing the shut-off valves in a drying system.

For every state we include a method in the System interface, in which the process control of the APCM must be implemented. If a system does not implement a state, for example *ramp-down*, the method can be implemented as an empty method. We also include the CheckError() method to include routines that monitor the system and set an error state if an error occurs. As an example, the method could set an error state if the temperature of a system is out of bounds for a fixed time period.

To enable *store energy inherently*, the systems System2Point and SystemContinuous include the ST_StoreEnergyInherentlyState structure, which Section 6.3.3 describes in detail.

Both systems can also be used for *interrupt process* if they implement the optional interface InterruptProcess. For this, their SystemFlowControl() method has to be adapted to include the possibility to interrupt the process and the SystemFlowControlInterruptProcess() method is used instead of SystemFlowControl(). Also, the structures ST_InterruptProcess and ST_InterruptProcessState are added, which is specified in Section 6.3.4.

Table 6.2: Prefixes used to include the data-type in the name of all DR data model variables, based on [Beck23a].

Prefix	Data-type
b	boolean
n	integer
f	float
s	string
a	array

All class attributes of data type structure, which are shown in Figure 6.2 with the prefix `ST_`, are used for IT-OT-communication and published in the communication layer using an IT-OT-data exchange service. These structures are needed for the interaction of the automation program with the DR control algorithm, see Section 6.4. The abbreviated data-type of variables must be included in the variable name of variables that are part of the DR data model [Fuhr22a]. Following [Beck23a], for the implementation we use the prefixes in Table 6.2. As example, `fTankTemperature` would be the variable name of a variable representing a tank temperature as float value.

6.3.2 Demand response safety functions

To guarantee a safe operation of the APCM while executing DR measures, th DR automation program includes the `SelectControlMode()` and `SetpointCheck()` methods presented in [Fuhr21; Fuhr22a] in the actuator and system classes. The safety functions enhance the safety functions introduced by [Pant18]. The `SelectControlMode()` method, visualised in Figure 6.3, enables the safe control of actuators and systems by the DR control algorithm. The method selects between the *automatic mode*, the *algorithm mode* and the *manual mode*:

- In *automatic mode* the normal control routine of the actuator or system is executed.
- In *manual mode* they can be controlled by an external signal set manually for example on a HMI. The *manual mode* has the highest priority and can be activated directly. It can be used for maintenance such as a manual functionality check of individual actuators and systems, for example.
- In *algorithm mode* the control by an external algorithm such as the DR control algorithm is granted. To activate the *algorithm mode*, the *algorithm permission* has to be true. This variable must only be set true, if the actuator or system is in a safe state. As an example, when controlling a heating system, the permission could only be given while the system temperature is inside defined bounds. The check for *algorithm permission* can be implemented in a separate method or set by an external sensor signal.

Depending on the activated mode, the respective setpoint (automatic, manual or algorithm) is set in the corresponding actuator or system. The variables to set *manual enabled*, *algorithm enabled* and *algorithm enabled* and the setpoints for *manual mode* and *algorithm mode*, which are set externally, must be included in the `ST_Control` structure. `ST_ControlState` should then show the control state and the setpoints that are currently activated in machine- and human-readable form as integer and string values respectively.

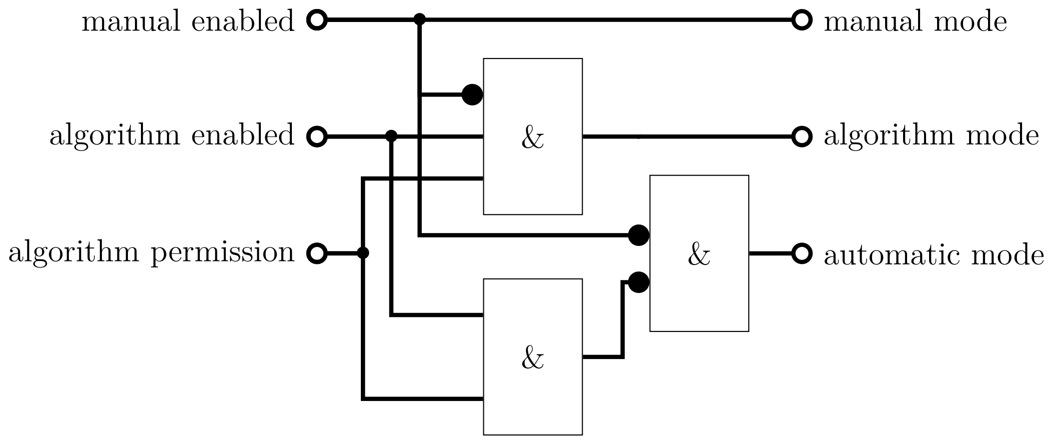


Figure 6.3: Method `SelectControlMode()` as part of the DR safety functions. The *manual mode* can be selected at all times, *algorithm mode* can only be accessed if the *algorithm permission* is set. The *automatic mode* is set by default if *manual mode* and *algorithm mode* are inactive. Figure adapted from [Fuhr21].

Table 6.3: Information that is included as variables in the `ST_Control`, `ST_ControlState`, `ST_SetSetpoint` and `ST_SetpointState` structures. For each variable its static or dynamic type and access mode are described.

Data structure / variable	Type*	Access*
`ST_Control`		
Activate manual mode	d	w
Switch on/off in manual mode	d	w
Activate algorithm mode	d	w
Switch on/off in algorithm mode	d	w
`ST_ControlState`		
Current actuator state (on/off)	d	r
Currently activated access mode	d	r
`ST_SetSetpoint`		
Setpoint in manual mode	d	w
Setpoint in algorithm mode	d	w
`ST_SetpointState`		
Current operating point	d	r
Current setpoint	d	r
Setpoint unit	d	r
Operating point unit	d	r
Setpoint limits	d	r

* s - static, d - dynamic, r - read, w - write

The second safety function is `SetpointCheck()`. We define $x \in \mathbb{R}$ as setpoint of the actuator or system and $x_{\mathrm{lb}} \in \mathbb{R}$ as lower and $x_{\mathrm{ub}} \in \mathbb{R}$ as upper bound of the setpoint. By implementing

$$x = \begin{cases} x, & \text{if } x_{\mathrm{lb}} \leq x \leq x_{\mathrm{ub}} \\ x_{\mathrm{lb}}, & \text{if } x < x_{\mathrm{lb}} \\ x_{\mathrm{ub}}, & \text{if } x > x_{\mathrm{ub}} \end{cases}, \tag{6.1}$$

the method guarantees that the setpoint x stays within its bounds. Thereby, it prevents damage to the APCM if a wrong setpoint is set manually or by the DR control algorithm. The setpoint is set depending on the activated control mode (*automatic mode*, *manual mode* or *algorithm mode*) such that the setpoints for each mode must be included as a variable in `ST_SetSetpoint` and the currently activated setpoint in `ST_SetpointState`. Table 6.3 gives an overview of the variables included in the `ST_Control`, `ST_ControlState`, `ST_SetSetpoint` and `ST_SetpointState` structures.

6.3.3 Demand response data model for store energy inherently

For all machine modules that have been selected for *store energy inherently* by applying the DR potential analysis in Section 5.2 the information needed for the execution of a DR control algorithm must be include in the communication layer of the DR automation architecture. Therefore, the automation data specification includes the information as variables of the APCM's automation program. These variables are published by the PLC via Ethernet. For *store energy inherently*, the following data object structures are needed:

- The structure `ST_Control`, for `Actuator2Point` and `System2Point` objects, and additionally `ST_SetSetpoint`, for `ActuatorContinuous` and `SystemContinuous` objects, to enable control of the actuators and systems by the DR control algorithm.
- The structure `ST_ControlState`, for `Actuator2Point` and `System2Point` objects, and additionally, `ST_SetSetpotintState`, for `ActuatorContinuous` and `SystemContinuous` objects, to get a feedback in the DR control loop.
- The `ST_StoreEnergyInherently` structure includes additional information for the DR control algorithm.

The last section already described the methods and data structures needed for *store energy inherently*, which are part of the safety functions and can be directly used for the DR control of actuators and systems. The `ST_StoreEnergyInherently` structure includes [Fuhr23b]:

- **Rated power**: The rated power P_f of the machine module f that is used for *store energy inherently* needs to be included. The value can be used to determine the DR potential. The data is static and only changes when a machine module is replaced.
- **System state correlation**: Here, all system states are listed in which the machine module can be used for *store energy inherently*. As described in Section 6.3.1, we use the states *stand-by*, *ramp-up*, *operational*, *working* and *ramp-down* based on [VDMA34179]. Most of the machine modules are only active and can be used for *store energy inherently* when their systems is in the state *working*. However, a tank heater for example can also be used during *ramp-up* while heating up the cleaning liquid. The correlation is defined in the automation program and it is hence static.

Table 6.4: Information that is included as variables in `ST_StoreEnergyInherently` as part of the DR data model for *store energy inherently*. For each variable its use for the DR control algorithm, its static or dynamic type and access mode are described. Table adapted from [Fuhr23b].

Data structure / variable	Use for DR control algorithm	Type*	Access*
`ST_StoreEnergyInherently`			
Rated power	DR measure's power or energy value	s	r
System state correlation	Execute DR measure only when active	s	r
Process step correlation	Execute DR measure only when active	s	r
Power consumption	Feedback of DR measure execution	d	r
Operating point	Load reduction or load increase possible	d	r
Process value	Prediction of machine behaviour	d	r
Flexibility limits	DR measure constraints	d	r

* s - static, d - dynamic, r - read, w - write

- **Process step correlation**: If the *working* state of the system is separated into different process steps, we add the process step in which the machine module can be used for *store energy inherently*. An air heater can only be used if drying is active, for example. Similar to the system state correlation, this information is defined in the automation program and therefore static.
- **Power consumption**: The current power consumption of the machine module used for DR should be included as feedback if a DR is executed successfully. If the power consumption is not measured per machine module, the APCM's total power consumption can be used. The power consumption can also be used to forecast the APCM's energy consumption.
- **Operating point**: This represents the current state of the machine module. Depending on whether the machine module is modelled as `Actuator2Point` or `ActuatorContinous`, this is either a boolean value (on/off) or the current operating point in percentage. The mandatory value is used to determine whether the machine module can currently be used for load increase (for example if the machine module is off) or reduction (if it is on) and as feedback for successful execution of the DR measure.
- **Process value**: This represents the value of the system that the machine module is controlled to manipulate. For example, a tank heater is controlled such that a specific tank temperature is reached. In this case, the tank temperature is the process value. The current process value is needed to predict the machine behaviour when planning DR measures. In our example, the DR control algorithm can only switch off the tank heater as long as the tank temperature is high enough and a stable process can be guaranteed. Therefore, it needs to model the cooling of the tank based on the current tank temperature.
- **Flexibility limits**: We also include the limits of the process value that have to be satisfied for a stable process. In the example of the tank temperature, this would be a minimal and maximal temperature. The limits can vary, for example depending on the

system state. If the tank heating system is in *ramp-up*, the limits may be more flexible than in *working*.

Table 6.4 presents an overview of all variables needed for *store energy inherently*.

6.3.4 Demand response data model for interrupt process

For *interrupt process*, the method `SystemFlowControlInterruptProcess()` adapts the flow control of a `System2Point` or `System2Continuous` object such that the system's process can be interrupted. In addition, the following data structures are needed:

- `ST_Control` and `ST_ControlState` to activate the *algorithm mode* and to check whether it is activated respectively.
- `ST_InterruptProcess` to enable the setting of a process interruption by the DR control algorithm.
- `ST_InterruptProcessState` which includes the information needed by the DR control algorithm to calculate a possible interruption.

The `ST_InterruptProcess` must include variables for manual interruption and interruption by the DR control algorithm. The data structure `ST_InterruptProcessState` models the following information [Fuhr23b]:

- **Step energy consumption**: For all steps, that can be interrupted, we include the total energy consumption W_g per step g. It can be used to determine the DR potential. The step energy consumption is calculated using the duration of the process steps and therefore varies depending on the active cleaning program. It is also possible to include the power consumption P_g and the duration of a process step in `ST_InterruptProcessState` such that the step energy consumption can be calculated externally by the DR control algorithm.
- **System power consumption**: Equal to `ST_StoreEnergyInherently` we include the system's power consumption as feedback for the DR control algorithm, if available.
- **System state**: We include the current system state. Normally, interruptions are only possible during *working*.
- **Interruption countdown**: The value represents the duration until the next possible interruption point in the process. Using this value, the DR control algorithm can determine the next feasible time for a process interruption.

Table 6.5 shows an overview of the variables included in the DR data model for *interrupt process*.

The `SystemFlowControlInterruptProcess()` method modifies the `SystemFlowControl()` method such that interruptions become possible. Based on the DR potential analysis for *interrupt process* in Section 5.3, the possibility for interruptions is included in the selected process steps. Usually, interruptions are carried out at machine level, such that the system that represents the APCM as a whole must be modified. Figure 6.4 shows an example for a modified process flow. Here, the *working* state is modified and the possibility is included to interrupt before the start of the *cleaning* and the *drying step 2*. If the variable *interrupt* is TRUE, before *cleaning* or *drying step 2*, the process is interrupted until the variable is set to FALSE.

Table 6.5: Information that must be included as variables in the `ST_InterruptProcess` and `ST_InterruptProcessState` data structures for *interrupt process*. For each variable its use for the DR control algorithm, its static or dynamic type and access mode are described. Table adapted from [Fuhr23b].

Data structure / variable	Use for DR control algorithm	Type*	Access*
`ST_InterruptProcess`			
Interruption command	Enable process interruption	d	w
`ST_InterruptProcessState`			
Step energy consumption	DR measure's energy value	d	r
System power consumption	Feedback of DR measure execution	d	r
System state	Availability for interruption	d	r
Interruption countdown	Next moment for DR measure	d	r

* s - static, d - dynamic, r - read, w - write

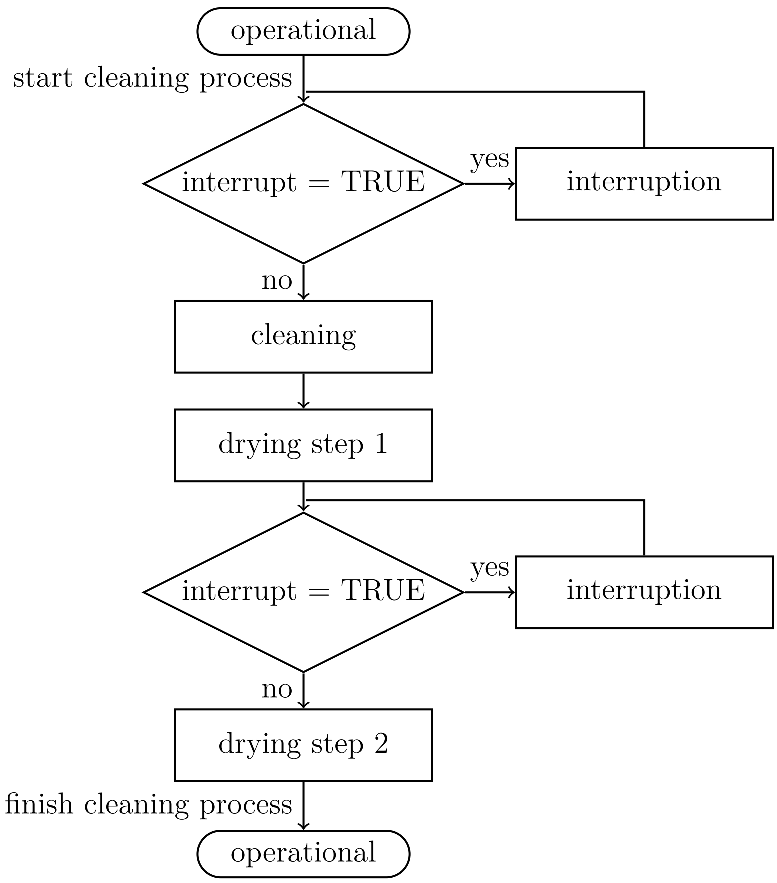


Figure 6.4: Modified program flow of an exemplary cleaning process that includes the possibility to interrupt the process before the start of *cleaning* and before the start of *drying step 2*. The flow chart only displays the *operational* state and the cleaning process steps of the *working* state of the APCM. Figure adapted from [Fuhr23b].

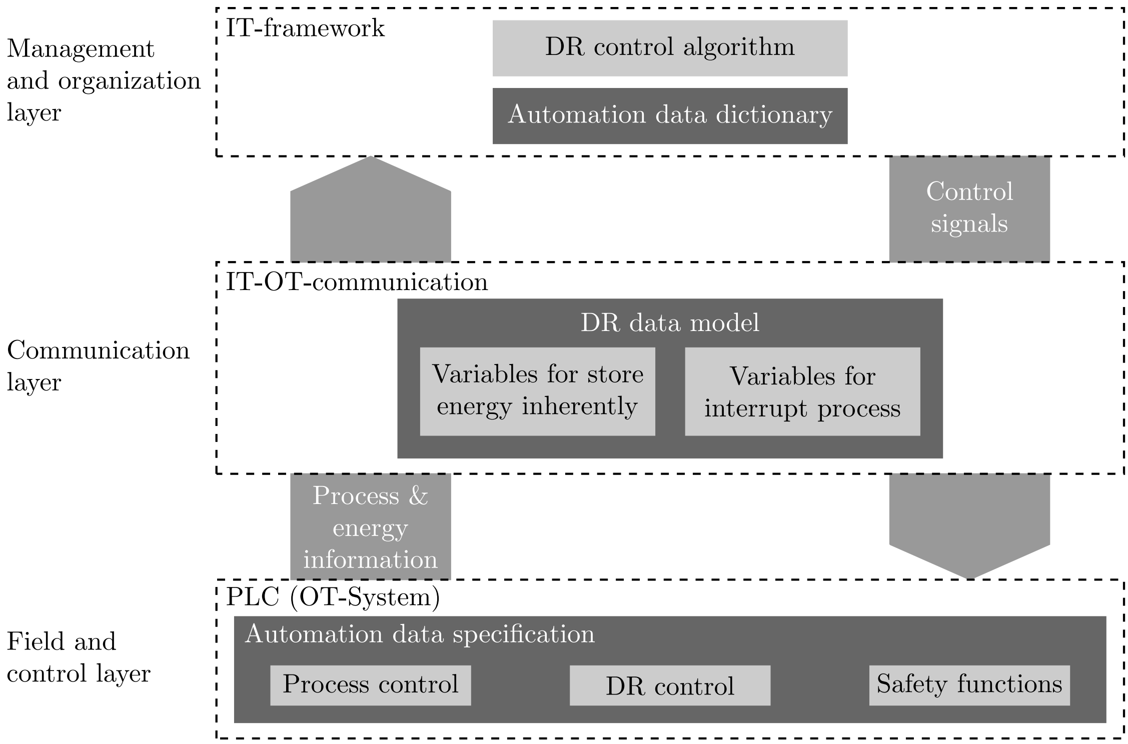


Figure 6.5: In the DR automation architecture the demand response (DR) data model is used for the IT-OT-communication between the field and control layer and the management layer. A mapper service as part of the IT framework maps the variables of the DR data model to the variables of the DR control algorithm using the automation data dictionary. Figure adapted from [Fuhr23b].

6.4 Automation data dictionary

The cyber-physical production system APCM, introduced in Section 6.2, includes two different kinds of communication. The first kind is the communication between the digital twin and external entities. It can be assumed that the digital twin and external entities are each part of IT-systems and the communication between them is implemented as a standard feature of these IT-systems. The second kind is the communication between the physical APCM and the digital twin. To execute DR measures, an IT-OT-communication between the APCM's automation system and the DR control algorithm must be implemented. In the following, the focus is the IT-OT-communication which we already presented in [Gros22a].

For the communication between the APCM's automation system and the DR control algorithm an IT-OT-data exchange service is needed, the automation data dictionary and an interpreter which is part of an IT-framework [Gros22a]. The IT-framework is also where the DR control algorithm is deployed, see Figure 6.5. The PLC is the OT-system where the DR automation program specified in the automation data specification is implemented. All the variables needed to execute the DR measures *store energy inherently* and *interrupt process* are included in the DR data model as part of the IT-OT-communication. This includes process and energy information as well as control signals. The interpreter uses the automation data dictionary to map the variables of the DR data model to the variables of the DR control algorithm.

The interpreter and the DR control algorithm are both implemented using a high-level programming language such as Java, C++ or Python. The mapper maps the variables of the DR data model to the variables of the DR control algorithm using the automation data dictionary. This includes:

- the Internet Protocol (IP)-address and identifier of the IT-OT-data exchange service,
- authentication data such as user name and password to connect to the IT-OT-data exchange service if needed,
- the information to identify variables in the IT-OT-data exchange service for all variables that are used by the DR control algorithm and
- a mapping of name and data-type between the associated variables of the IT-OT-data exchange service and their high-level language variables.

The DR automation program includes the data-type in the variable's name to facilitate a mapping of the data-type between IT-OT-data exchange service and high-level language, see Section 6.3.1.

In the application of the DRAAD method, this work uses OPC UA as IT-OT-data exchange service. When using OPC UA and implementing the object-oriented automation program as proposed in Section 6.3.1, this creates a hierarchical structured OPC UA server [Fuhr21] and the automation data dictionary includes [Gros22a]:

- the IP-address and name of the OPC UA server,
- the OPC UA client user name and password to connect the DR control algorithm as OPC UA client,
- the OPC UA node IDs and data types of all OPC UA nodes used for interaction with the DR control algorithm and
- mapping of name and data-type between the associated OPC UA nodes and their high-level language variables.

The automation data dictionary should be implemented in a lightweight, structured and text-based file format such as JSON or YAML Ain't Markup Language (YAML), because these file formats are open, standardized and made for data-exchange between applications which are written in different programming languages.

6.5 Summary

In this chapter a DR automation architecture that enables the execution of DR measures on APCMs and fulfils all design criteria specified in Section 6.1 was presented. This is the second step of the DRAAD method.

The DR automation architecture implements the DR measures *store energy inherently* and *interrupt process* using a DR automation program with data objects modified for *store energy inherently* and *interrupt process* (C2.1). By integrating the DR control algorithm in an IT-system that is separated from the DR automation program at OT-level and implementing an IT-OT-communication between them, a DR automation architecture is realised that is independent of the deployment architecture (C2.2). Different deployment architectures can be chosen, as long as a functioning IT-OT-communication is ensured.

The DR data model includes all variables needed to observe the state of the APCM (C2.3) and to control it for DR (C2.4). The DR control algorithm is implemented in an IT-framework that is capable of communicating with other IT-systems (C2.5). By using an object-oriented automation program specified in the automation data specification, a DR automation architecture is created that is scalable to APCMs of different sizes and transferable to other machine models (C2.6). In [Fuhr22a] we show that the automation program is not limited to APCMs but can also be implemented in building automation systems.

The DR automation program also includes safety functions to guarantee process safety while executing DR measures (C2.7). An IT-OT-communication can be realised by using an IT-OT-data exchange service such as OPC UA (C2.8). For the IT-OT-communication a DR data model is used, which is mapped to the DR control algorithm using the automation data dictionary. A naming convention for the variables of the DR data model enables an easy mapping (C2.9).

In the following, the DR control algorithm is described that interacts with the automation program.

7 Demand response control algorithm

The third step of the DRAAD method is the development of the DR control algorithm. The DR control algorithm interacts with the machine automation and uses the data points of the automation data model to calculate and execute the DR measures *store energy inherently* and *interrupt process*. The DR control algorithm applied to an APCM, which is presented in Section 8.4, has already been published in our work [Fuhr23a]. This chapter shows the generalised version of the DR optimisation model integrated in the DR control algorithm.

Section 7.1 recapitulates the design criteria for the DR control algorithm. Based on these design criteria in Section 7.2 the DR optimisation model is developed. Section 7.3 summarises the chapter.

7.1 Design criteria for the demand response control algorithm

For the design of the DR control algorithm, Section 3.2 defines design criteria. The following list presents the summarised design criteria:

C3.1 The DR control algorithm schedules the DR measures *store energy inherently* and *interrupt process*.

C3.2 The DR control algorithm is scalable and transferable to different kinds and numbers of APCMs.

C3.3 The DR control algorithm avoids a negative impact on process stability.

C3.4 The DR control algorithm uses a white-box or grey-box approach.

C3.5 The DR control algorithm includes the APCM's machine states and changing electricity prices.

The DR control algorithm is developed based on these design criteria.

7.2 Mathematical modelling

Following our work in [Fuhr23a], we choose to use a MILP scheduling model to represent the behaviour of the APCM. The model is used to predict the APCM's behaviour as part of an economical MPC algorithm. We split the model in an event-based sub-model of the cleaning process for *interrupt process* and a time-based sub-model of the machine modules that are used for *store energy inherently*. Both sub-models are explained in detail in Section 7.2.1 and Section 7.2.2.

The objective of the overall model is to minimise the energy costs of the APCM. The costs are the product of the energy price $C_k \in \mathbb{R}$ at time step $k = 0, \ldots, K$, where $K \in \mathbb{N}$ is the optimisation horizon, and the APCM's power consumption. The APCM's power consumption is separated into two parts. First, we include the power consumption $P_n \in \mathbb{R}_{\geq 0}$ of the process event $n = 1, \dots, N$, where $N \in \mathbb{N}$ is the total number of cleaning process events. A cleaning process event is defined by its start time $s_n \in \mathbb{N}_0$ and duration $d_n \in \mathbb{N}_0$, for example in seconds, and represents either one single process step, a fraction of a process step, a group of coherent process steps or a process interruption. Second, we include the product of the electrical power consumption $P_l \in \mathbb{R}_{\geq 0}$ of the $L \in \mathbb{N}$ machine modules selected for *store energy inherently* with their setpoint $h_{l,k} \in [0, 1]$ at time step k. We obtain

$$f(\mathbf{d}, \mathbf{H}) = \sum_{n=1}^{N} \sum_{k=s_n}^{s_n+d_n-1} P_n C_k + \sum_{l=1}^{L} \left(P_l \sum_{k=0}^{K} h_{l,k} C_k \right) \tag{7.1}$$

as the objective function. The energy costs are calculated over the process event durations $\mathbf{d} = (d_1, \dots, d_N) \in \mathbb{N}_0^N$ and the machine module's setpoint states

$$\mathbf{H} = \begin{bmatrix} h_{1,0} & \cdots & h_{1,K} \\ \vdots & \ddots & \vdots \\ h_{L,0} & \cdots & h_{L,K} \end{bmatrix} \in [0, 1]^{L \times (K+1)}. \tag{7.2}$$

If the duration d_n of an event n is zero, the energy costs

$$\sum_{k=s_n}^{s_n+d_n-1} P_n C_k = 0 \tag{7.3}$$

become zero, since the special case of an empty sum, defined by

$$\sum_{x=y}^{z} = 0, \text{ if } z < y, \tag{7.4}$$

applies [Heus09, p 90]. This also applies for the energy costs of energy-flexible machine modules

$$\sum_{l=1}^{L} \left(P_l \sum_{k=0}^{K} h_{l,k} C_k \right) = 0 \tag{7.5}$$

if no machine module was selected for *store energy inherently*, where $L = 0$.

7.2.1 Event-based sub-model of the cleaning process for interrupt process

As stated above, an event n is characterized by its starting time s_n, its duration d_n and its power consumption P_n. The event durations $\mathbf{d}$ are considered as optimisation variables, which means that the algorithm changes $\mathbf{d}$ and thereby modifies starts and durations of the events to minimise the energy costs. Different cleaning process steps should be combined to one event if they cannot be interrupted. A process step can also be separated into multiple events, if the process step can be interrupted during its execution.

An example is shown in Figure 7.1 where we separate the process step *Cleaning* into five events. If we look at the first step *Cleaning 1*, the events $n = 1, 3, 5$ are segments of the actual cleaning procedure and the events $n = 2, 4$ are interruptions. The step *Cleaning 2* which consists of the events $n = 7, \dots, 11$ has a longer duration since the interruptions are longer, but the duration

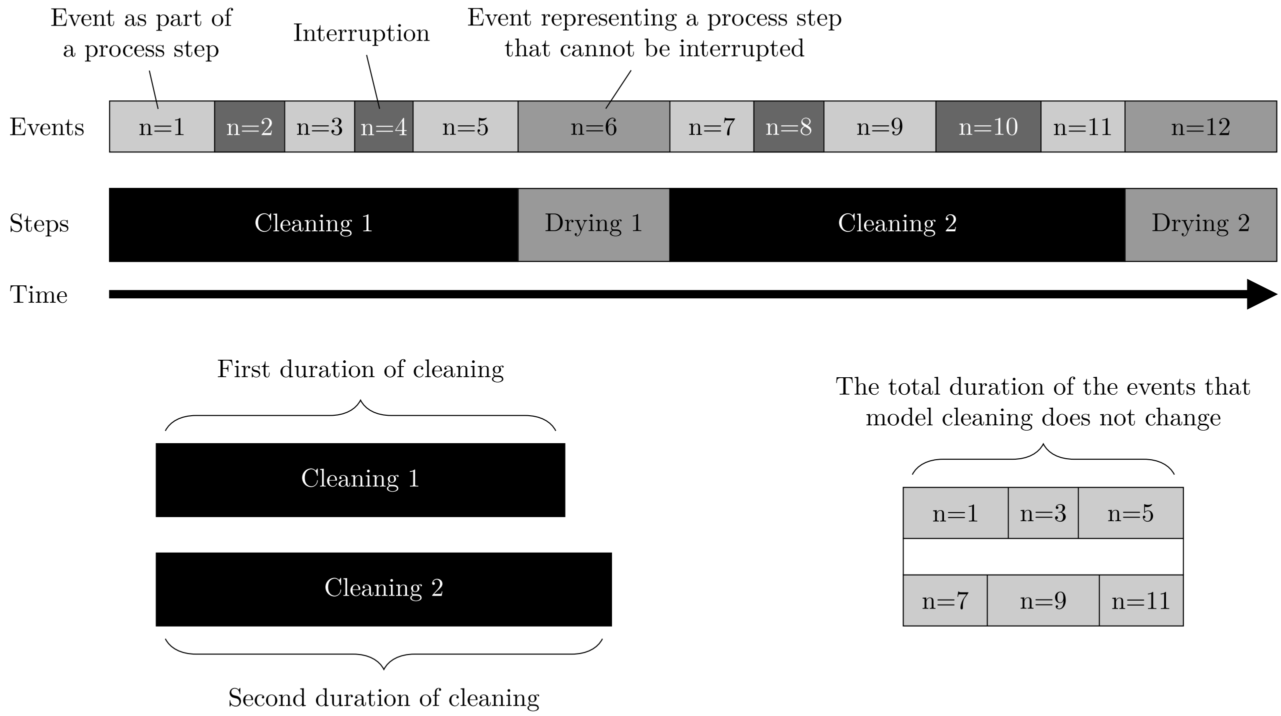


Figure 7.1: Model of two consecutive cleaning processes as an event-based model. The process step *cleaning* is separated into three cleaning events ($n = 1, 3, 5$ and $n = 7, 9, 11$) as well as two interruption events ($n = 2, 4$ and $n = 8, 10$), without fixed durations. The durations of the *drying* process events are fixed and the corresponding events ($n = 6, 12$) cannot be interrupted.

of the cleaning procedure does not change. The process step *Drying* cannot be interrupted and therefore is modelled as one event. It is also possible to include a sequence of cleaning processes in our model, as shown in the example.

The event's power consumption P_n is the sum over all rated power values P_f of all machine modules active during the process event n, see Section 5.2. Analogously to (5.7) and (5.8), we define

$$\mathcal{B}_n := \{f \mid f\text{-th machine module is active in the process event } n\} \tag{7.6}$$

as the index set of all machine modules active during the process event n and calculate the event's power consumption P_n by

$$P_n = \sum_{f \in \mathcal{B}_n} P_f. \tag{7.7}$$

If the event n represents more than one process step, the time-based weighted average of the step's cumulated rated power should be used.

Since the model is part of a MPC loop, it is optimized multiple times during the execution of the cleaning process. Therefore, we set $n_{\text{start}} \in \{1, \ldots, N\}$ as a variable that represents the event that is currently activated on the APCM. In every iteration of the MPC, n_{start} is updated using the APCM's live data. We set the starts and durations of all past events to zero, that is

$$s_n = 0, \, \forall n \leq n_{\text{start}} \tag{7.8}$$

and

$$d_n = 0, \forall n \leq n_{\text{start}}. \quad (7.9)$$

The start of the events following the start event is

$$s_{n+1} = s_n + d_n, \forall n = n_{\text{start}}, \ldots, N-1. \quad (7.10)$$

Hence, they are not subject to the optimisation any more since they have passed. If a process event n represents a process step or a group of process steps that cannot be interrupted, its duration d_n is fixed to the duration $d_{\text{event}} \in \mathbb{N}$ defined by the total duration of the process step or the sum of the durations of the process step group. If the starting event n_{start} cannot be interrupted we set its duration to the remaining event duration $d_{\text{start}} \in \mathbb{N}$ which is also updated in every MPC iteration.

The cleaning process must be terminated before or at the time step $S \in \mathbb{N}$ such that start and duration of the last event N is set to

$$s_N + d_N \leq \min(K, S). \quad (7.11)$$

This ensures that the production's takt time is met.

7.2.2 Time-based sub-model of the machine modules selected for store energy inherently

The second sub-model represents the machine modules selected for *store energy inherently*. A machine module l is characterized by its electrical power consumption P_l and its setpoint state $h_{l,k}$ at time step k. To model the power consumption P_l, we use the machine module's respective rated power. The set point states $\mathbf{H}$ of the machine modules selected for *store energy inherently* are considered as optimisation variable.

As described in Section 5.2, a machine module is always part of the system it operates in. For example, a tank heater is part of a tank heating system which includes the tank heater and a tank storing the cleaning liquid. The operation of the machine module changes the system's process value $v_{l,k} \in \mathbb{R}$, in our example the tank temperature. We model the change of the system's process value $v_{l,k}$ with the time-discrete function

$$v_{l,k+1} = v_{l,k} + \Delta v_{l,k}, \forall l = 0, \ldots, L, \forall k = 0, \ldots, K, \quad (7.12)$$

where $\Delta v_{l,k}$ is the change of the process value during the time step k. The process value change $\Delta v_{l,k}$ depends on the module's setpoint $h_{l,k}$ and disturbance $\tau_{l,k} \in \mathbb{R}$ and can be described as the function

$$\Delta v_{l,k} = g\left(h_{l,k}, \tau_{l,k}\right). \quad (7.13)$$

For the tank heating system, the temperature loss to the environment or to the work pieces would be modelled as a disturbance. We define

$$v_{l,0} = v_{l,\text{start}}, \forall l = 0, \ldots, L, \quad (7.14)$$

where $v_{l,\text{start}} \in \mathbb{R}$ is the current process value of the machine system where the machine module l is located and which this machine module influences. The process value $v_{l,k}$ is constrained by

$$v_{\text{lb},l} \leq v_{l,k} \leq v_{\text{ub},l}, \forall l = 0, \ldots, L, \forall k = 0, \ldots, K, \quad (7.15)$$

with $v_{\mathrm{lb},l} \in \mathbb{R}$ as the lower bound and $v_{\mathrm{ub},l} \in \mathbb{R}$ as the upper bound of the process value $v_{l,k}$. In our example, this would be the minimal and maximal temperature the cleaning liquid in the tank can have, that guarantees a stable cleaning process.

Finally, considering the objective function (7.1) and the constraints stated in the last two sections, we formulate the optimisation problem by

$$\begin{aligned} &\min_{\mathbf{d},\mathbf{H}} && f(\mathbf{d},\mathbf{H}) \\ &\text{such that} && (7.8),\ (7.9),\ (7.10),\ (7.11)\text{ and }(7.15)\text{ hold.} \end{aligned} \tag{7.16}$$

7.3 Summary

The third step of the DRAAD method, presented in this chapter, creates the DR control algorithm based on a MILP model. The DR control algorithm implements *store energy inherently* and *interrupt process* in an time-based and event-based sub-model which are combined in one common objective function (C3.1). The DR control algorithm integrates a MILP model that can be used to model APCMs of different kinds and sizes (C3.2).

By modelling the process value of systems which are influenced by *store energy inherently* and introducing lower and upper bounds for the process values, it is ensured that the DR control algorithm does not have a negative impact on the process stability of the APCM (C3.3). The used MILP model is a white-box model (C3.4), that models the interaction with the APCM by including the APCM's machine states in the constraints of the MILP. The interaction with the electricity market is incorporated by including the electricity price in the objective function (C3.5).

In the next chapter, the DRAAD method is applied to a real APCM, which includes the DR potential analysis, implementation of the DR automation architecture as well as the implementation of the DR control algorithm. The set-up is validated by executing the DR measures *store energy inherently* and *interrupt process* in a field test.

8 Application to an aqueous parts cleaning machine

The DRAAD method is applied to the APCM MAFAC KEA, shown in Figure 8.1, in the ETA Research Factory at the Technical University of Darmstadt. Section 8.1 starts with the description of the APCM and the reference cleaning process, followed by the analysis of the APCM's DR potential. We already presented both topics in [Fuhr23b]. Section 8.3 shows the implementation of the DR automation architecture extending our work in [Fuhr22a; Fuhr23a]. Section 8.4 presents the implementation of the DR control algorithm. In Section 8.5 the complete set-up, as the result of the applied DRAAD method, is validated in a field test. The applied DR control algorithm and the field test were already presented in our work [Fuhr23a].

As part of [Fuhr23a] we published the complete source code of the applied DR automation architecture, see `https://github.com/PTW-TUDa/cpsl2023-dr-for-cleaning-machines`. The data includes the DR automation program as an OpenPLC project, the DR control algorithm implemented in Python and the results of the field test.

Figure 8.1: APCM MAFAC KEA in the ETA Research Factory.

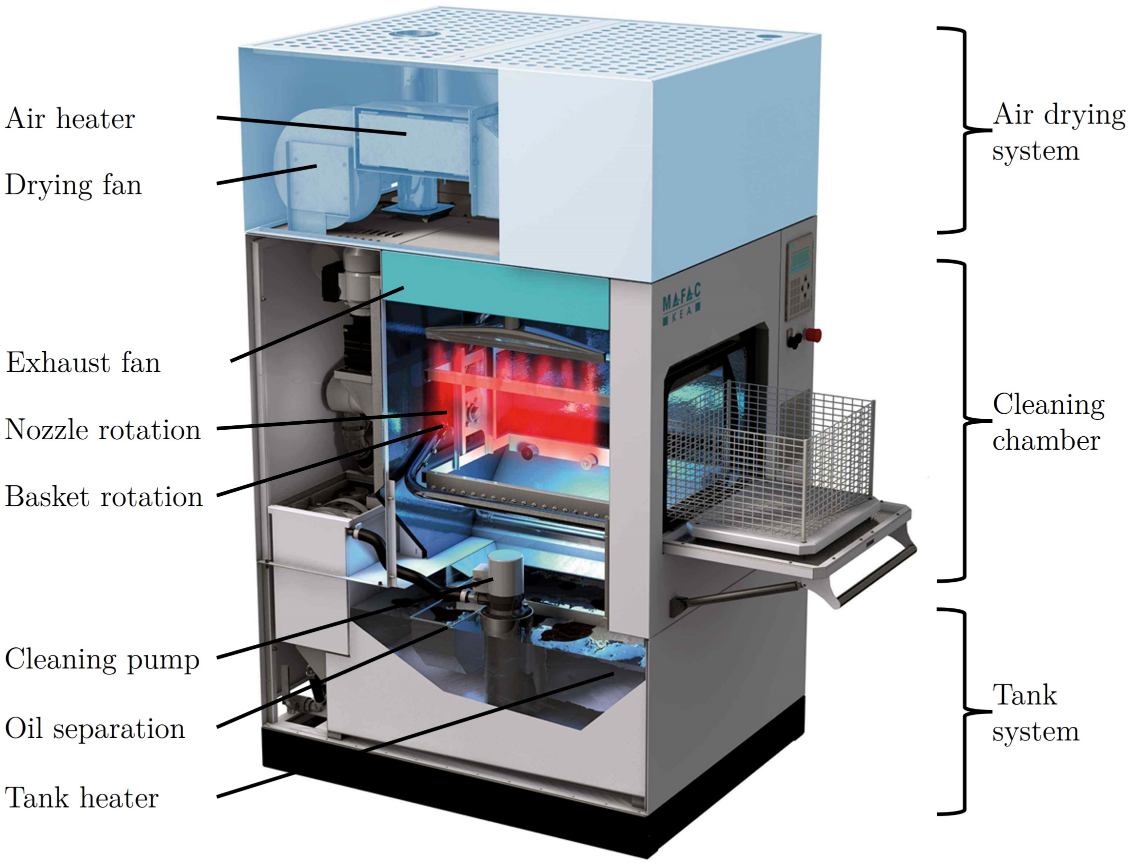


Figure 8.2: Cross section of the APCM MAFAC KEA including the three main systems and the electrical consumers. Figure adapted from [MAFA12]. Figure: ©MAFAC 2023.

8.1 Description of the examined aqueous parts cleaning machine and its process

The MAFAC KEA is an APCM operating in batch process with a closed cleaning chamber. Its three main systems are the cleaning chamber, the tank system and the air drying system, see Figure 8.2. The cleaning chamber contains a rotating basket carrying the parts, a rotating nozzle system, the central cleaning pump, compressed air and an exhaust fan for removal of moist air. The 320-litre tank filled with aqueous cleaning detergent, an electric tank heater, a temperature sensor and an oil separation system form the tank heating system. The air drying system consists of an electric air heater, a drying fan and a temperature sensor. The parts are loaded and unloaded manually. The APCM's total rated power P_{total} is 20.7 kV A and its total power consumption is measured by a Janitza UMG 96 power analyser.

The reference cleaning process has a duration of approximately 12 min. Table 8.1 lists the process steps and the machine modules with their activation duration. The durations of the process steps and the activation durations of the machine modules are fixed. The air heater, tank heater and the oil-separation system are hysteresis-controlled or demand-driven and their activation duration varies. We track the activation duration for these machine modules throughout multiple cleaning cycles and calculate the average activation duration.

Table 8.1: Process steps of the reference process of the MAFAC KEA APCM with a total duration of approximately 12 min. The process consists of the five process steps *spray cleaning*, followed by *dripping*, *suction*, *blowing* and *convection drying*. The step duration d_g, the machine modules f that are active in the respective process steps and their activation duration $d_{f,g}$ per process step g are shown. Table based on [Fuhr23b].

Process step	d_g in s	Machine modules activated	$d_{f,g}$ in s
Spray cleaning	600	Cleaning pump	600
		Basket rotation	600
		Nozzle rotation	600
Dripping	10	Basket rotation	10
Suction	10	Exhaust fan	10
		Basket rotation	10
Blowing	10	Exhaust fan	10
		Basket rotation	10
Convection drying	90	Exhaust fan	90
		Drying fan	90
		Basket rotation	90
		Air heater	57*
Homing	5*	Basket rotation	5*
		Nozzle rotation	5*

*temperature-controlled or position-dependent, average value over multiple process cycles

The main cleaning process step *spray cleaning* is the first step and has a duration of 600 s. In this step, the cleaning pump pumps the detergent into the cleaning chamber where it is sprayed at the parts. The nozzle frame and basket both counterrotate. All three machine modules are active during the whole step.

Spray cleaning is followed by the three short drying process steps *dripping*, *suction* and *blowing*, which each has a duration of 10 s. During *dripping* only the basket rotation is active, in *suction* and *blowing* the exhaust fan is activated additionally to remove moist air from the cleaning chamber. The nozzles are operated with compressed air while *blowing*. We do not consider the nozzles since they are not operated electrically but by compressed air generated outside the APCM. The final drying step is *convection drying*, where the parts are dried by hot air for 90 s. The exhaust fan, the drying fan and the basket rotation motor operate constantly during the entire process step while the air heater is controlled in hysteresis to heat the air. The heater is switched on and off in intervals averaging at five to six activations lasting 57 s each.

At the end of the cleaning process, the basket and nozzle frames are rotated to their respective starting positions in the *homing* step. The activation duration of the basket and nozzle rotation depends on the position of the frames at the end of *convection drying*, such that it is hard to measure the duration of the movement individually. We use the average duration of 5 s for homing both rotation systems.

The tank heater is controlled in a hysteresis with a range between 55 °C and 65 °C. The activation duration of the tank heater varies throughout each process cycle and occasionally is zero if the tank temperature was sufficiently high prior to the execution. We measured the activation duration throughout several cleaning cycles resulting in 152 s average activation duration. The oil separation also operates independently from the process steps. It is operated depending on the determined oil level in the detergent and has an average activation duration of 100 s.

8.2 Demand response potential analysis

In this chapter, the DR potential analysis presented in Chapter 5 is used to analyse the DR potential of the APCM MAFAC KEA for *store energy inherently* and *interrupt process* . This is the first step of the DRAAD method. We already presented the DR potential analysis applied to the APCM MAFAC KEA in [Fuhr23b].

8.2.1 Demand response potential of store energy inherently

We start with the analysis of the machine modules and determine the corresponding rated power P_f of the electrical consumers from the APCM's technical documentation. We calculate the energy demand W_f and share of energy demand φ_f for each electrical consumer during the entire cleaning procedure based on their activation durations using (5.1) and (5.3). The five columns at the left of Table 8.2 summarise the results. In the following, we only examine controllable consumers that can be used for DR and therefore exclude electrical consumers such as the APCM's PLC or the control cabinet cooling.

Figure 8.3 shows the rated power P_f and share of energy demand φ_f for every machine module. We set the rated power $P_f = 1\,\mathrm{kW}$ and the share of energy demand $\varphi_f = 0.05$ as thresholds. We only analyse machine modules further with characteristic values above at least one of these thresholds. Therefore, we analyse the energy-process-independence in the next step only for the tank heater, the air heater and the cleaning pump.

In the second step, we evaluate the control mode and the inherent energy storage capacity of the three selected machine modules. The categories are visualised in Figure 5.2. The temperature-controlled tank heater operates independently of the cleaning process. The tank heater is neither controlled temporally nor in relation to the process. Consequently, we select C.1 as its control mode. As already mentioned, if the tank is warm enough at the start of a cleaning cycle, the tank heater can stay switched off during a full cycle, leading to the rating I.1.

The air heater is process-dependent since it is only active during *convection drying*. It can only be switched on and off with a fixed operating point and is hence rated with control mode C.3, as it is not controlled in quantity. The air heater has a small buffer capacity. The air heater is turned on, if the temperature of the air flow falls under 105 °C and off, if it is above 110 °C, resulting in five to six activations during *convection drying*. Hence, its inherent energy storage capacity does not cover the function. We must distinguish between two scenarios in order to decide if the process is critical. A minor undershoot of the normal temperature during drying is not critical if the APCM is utilised for the parts' final cleaning and corrosion can be excluded, because the parts can then air-dry. Drying is critical if the APCM is in the middle of a process chain and prepares the parts for delicate subsequent operations, such as hardening or coating. We rate the inherent energy storage capacity I.2 since we have no knowledge of the process

Table 8.2: The table shows rated power P_f, total activation duration per cleaning cycle d_f, absolute energy demand W_f and share of energy demand φ_f for every machine module. It also shows the results of the energy-process-independence analysis of the selected machine modules: control (ctrl.) mode, the inherent energy storage (IES) capacity and the resulting overall rating, based on [Fuhr23b].

Machine module	P_f in kW	d_f in s	W_f in Wh	φ_f	Ctrl. mode	IES capacity	Rating
Tank heater	10	152*	422.22	0.364	C.1	I.1	green
Air heater	8	57*	126.67	0.109	C.3	I.2	yellow
Cleaning pump	3	600	500.00	0.431	C.3	I.4	red
Exhaust fan	0.55	110	16.81	0.014	-	-	-
Drying fan	0.55	90	13.75	0.012	-	-	-
Basket rotation	0.25	725	50.34	0.043	-	-	-
Nozzle rotation	0.18	605	30.25	0.026	-	-	-
Oil separation	0.045	100*	1.25	0.001	-	-	-

*temperature- or level-controlled, average value over multiple process cycles

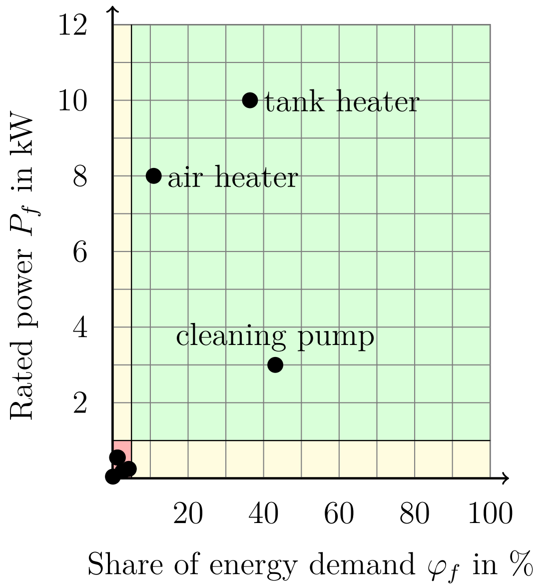


Figure 8.3: Rated power P_f and share of energy demand φ_f of all machine modules. The thresholds are $P_f = 1\,\mathrm{kW}$ and $\varphi_f = 0.05$. Only three machine modules with values above both thresholds, shown in the green sections, are examined further. The data values are shown in Table 8.2. Figure based on [Fuhr23b].

chain set-up the APCM will be operated in. The machine operator can set the DR limits to zero if drying is crucial and thereby omit the use of the air heater system for DR.

The cleaning pump is operated only during *spray cleaning* by an on-off controller leading to the control mode rating C.3. The pump directly converts electricity into hydraulic energy, a real-time energy conversion without any storage capacity, such that we rate it I.4. We summarise the evaluation of energy-process-independence in the three columns at the right side of Table 8.2.

We estimate the APCM's DR potential for *store energy inherently* to $P_{\mathrm{flex}} = 18\,\mathrm{kW}$ using (5.5) and calculate the relative DR potential with (5.6) to $\Phi_{\mathrm{P}} = 0.87$. We can only achieve the total DR potential if the tank heater and the air heater are used for *store energy inherently* at the same time. The air heater is only active during *convection drying* for a total average duration of 57 s. In average, it is switched on five to six times for approximately 10 s, such that the air heater and the full DR potential can only be used for short-term measures such as internal peak load shifting.

8.2.2 Demand response potential of interrupt process

To determine the DR potential of *interrupt process* we calculate the accumulated rated power P_g and the share of energy demand φ_g for every process step g using (5.8) and (5.9). The results are shown in Table 8.3 and Figure 8.4. Again, we set the thresholds $P_g = 1\,\mathrm{kW}$ and $\varphi_g = 0.05$. *Spray cleaning* has the highest share of energy demand and *convection drying* the highest accumulated rated power. The remaining four steps are energetically insignificant hence we do not consider them further for DR. We do not include the tank heater in this part of the potential analysis since it is process independent. The cleaning process has a total energy consumption of $W_{\mathrm{total}} = 739.75\,\mathrm{Wh}$.

In the second step, we determine if an interruption is technically possible without impairing the process quality. Statements are either supported by research or evaluated on a case-by-case basis. The three levels for the rating of the DR suitability are shown in Figure 5.3. As already mentioned, the cleaning result is not affected by a delayed start of the cleaning [Auri09]. Therefore, an interruption before the initial *spray cleaning* process is possible.

Convection drying is the second step taken into account. To prevent corrosion and prepare the parts for delicate subsequent operations such as hardening or coating, the drying process removes any residual cleaning liquid from the parts' surface. The majority of aqueous cleaning operations are followed by this procedure, which is typically accomplished through evaporation [Durk06, pp 2–3]. As far as we know, no studies have been done on how prolonged drying affects the cleanliness of the parts. We assume that corrosion is prevented if the drying is delayed only for short time intervals by a maximum of 10 minutes duration, such that it is acceptable to interrupt the process before *convection drying*.

We are unable to assess the impact on the parts' cleanliness if we interrupt a running cleaning process step. As a result, we avoid this until it is established how an interruption may affect the cleaning quality. The evaluation yields a yellow DR suitability rating for *convection drying* and *spray cleaning*, as indicated in the column at the right side of Table 8.3.

We consider all process steps with a green or yellow rating for the quantification of the DR potential. We calculate the absolute DR potential $W_{\mathrm{flex}} = 732.08\,\mathrm{Wh}$ and the DR ratio $\Phi_{\mathrm{W}} = 0.99$ by (5.12) and (5.14). The DR potential analysis reveals that the tank heater and air heater

Table 8.3: Accumulated rated power P_g, absolute energy demand W_g, share of energy demand φ_g and criticality for every process step g, based on [Fuhr23b].

Process step	P_g in kW	W_g in Wh	φ_g	Rating
Spray cleaning	3.43	571.67	0.775	yellow
Convection drying	9.35	160.42	0.217	yellow
Suction	0.8	2.22	0.003	-
Blowing	0.8	2.22	0.003	-
Dripping	0.25	0.69	0.001	-
Homing	0.43	0.59	0.001	-

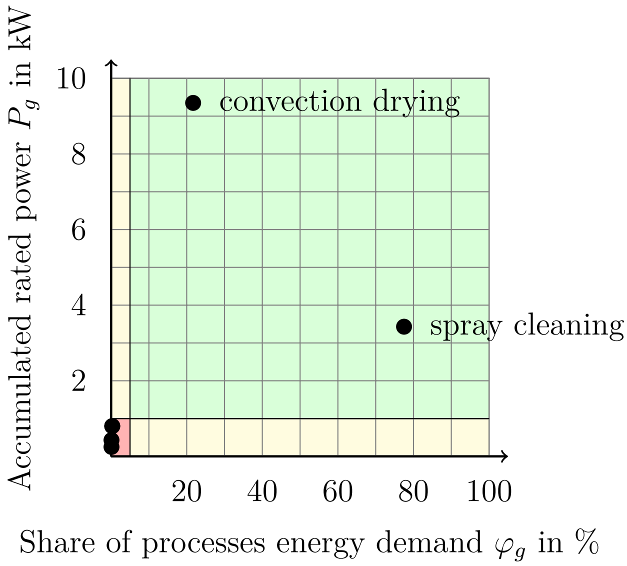


Figure 8.4: Accumulated rated power P_g and share of process energy demand φ_g of all process steps. The thresholds are $P_g = 1\,\text{kW}$ and $\varphi_g = 0.05$. Only process steps with values above one threshold, in the green section, are examined further. The data values are shown in table 8.3. Figure based on [Fuhr23b].

have a high DR potential for *store energy inherently* and the process steps *spray cleaning* and *convection drying* have a high DR potential for *interrupt process*.

8.3 Implementation of the demand response automation architecture

To utilise the identified DR potential in this section a DR automation architecture is implemented using the framework presented in Chapter 6. Section 8.3.1 presents the implemented object-oriented DR automation program. The IT-framework that used for the implementation of the DR control algorithm is shown in Section 8.3.2. Section 8.3.3 explains the IT-OT-communication using OPC UA as IT-OT-data exchange service and the OPC UA-connector of the selected *eta_utility* IT-framework.

8.3.1 Implementation of the object-oriented demand response automation program

The object-oriented automation program is implemented in IndraWorks ML 14V22 P10 using the IEC 6113-3 programming language Structured Text [IEC61131-3]. To model the APCM in the automation program, the APCM is separated into three logical systems:

- The system *tank* that includes the tank heater, the oil separation system, a temperature sensor and sensors to control quality and level of the cleaning liquid,
- the *inlet air heating* system which includes air heater, fan, a valve, a temperature sensor and an air flow sensor and
- the system *cleaning chamber* which includes the pump, the motors for basket and nozzle frame rotation, the exhaust air fan, the pressurised air system, sensors to detect the homing position of the basket and nozzle frame as well as safety sensors.

The systems are implemented by extending the base classes presented in Section 6.3.1 by the three classes `TankSystem`, `InletAirheatingSystem` and `CleaningChamberSystem`, see Figure 8.5. For a better understanding, the Unified Modeling Language (UML) diagram only shows the most important attributes, which are the data structures that must be included in the communication layer, and the methods that are used for DR. Additional attributes and methods that are not described are indicated using "..." in the diagram. This section describes the most important attributes and methods.

The system `KEASystem` represents the complete APCM and includes the three aforementioned systems. All four systems implement the `System` interface. The systems `TankSystem` and `InletAirheatingSystem` with their main machine modules tank heater and air heater, are used for *store energy inherently* and implement the `StoreEnergyInherently` interface. The APCM's main system `KEASystem` implements the `InterruptProcess` interface to be able to execute the *interrupt process* measure. In addition to the basic data structures for safety, *store energy inherently* and *interrupt process*, presented in Section 6.3.1, the following data structures are added:

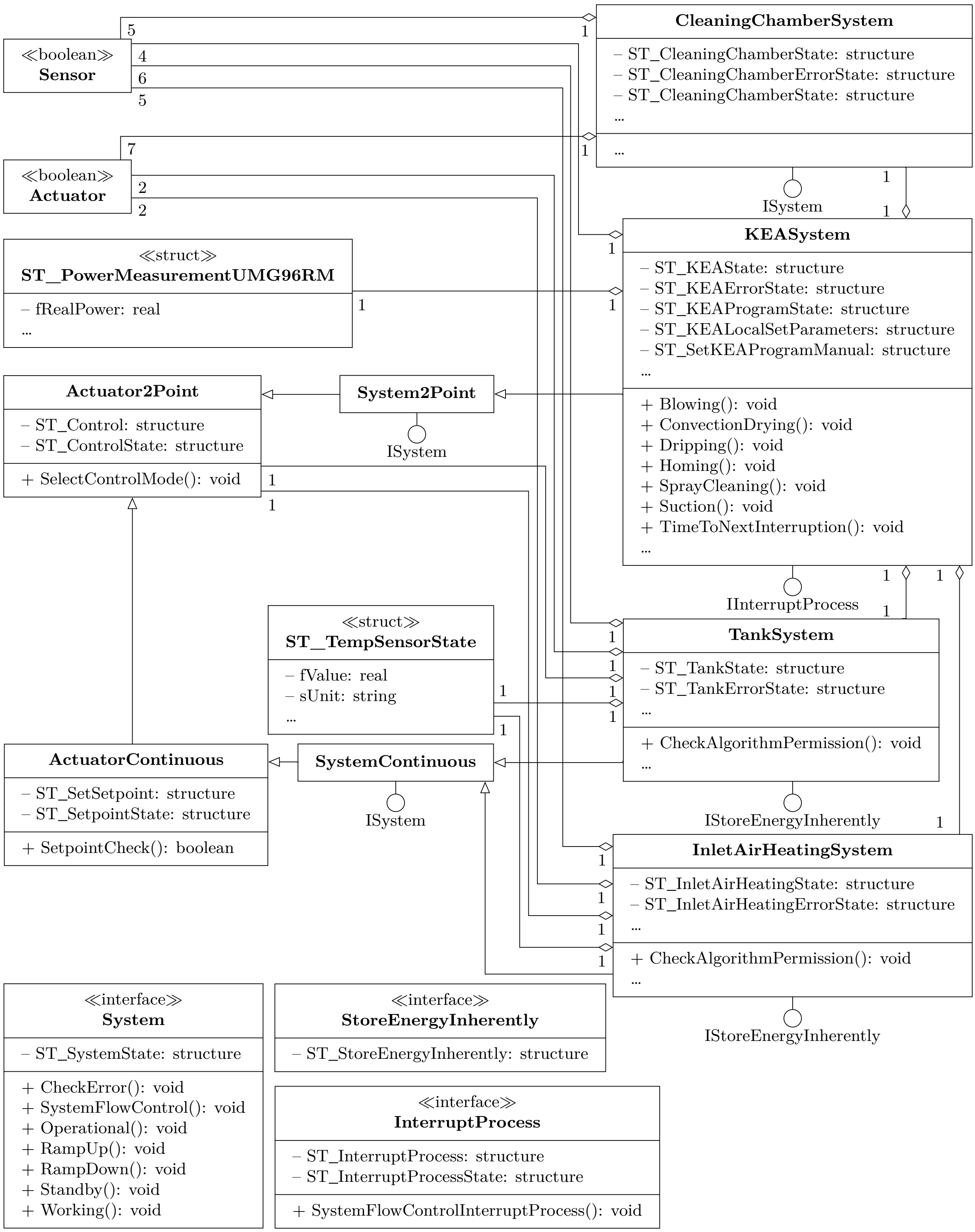


Figure 8.5: Object-oriented DR automation program for the APCM MAFAC KEA. The diagram only shows the attributes that are included in the communication layer and the most important methods needed for DR. The missing attributes and methods are symbolized by "...". The classes `CleaningChamberSystem`, `TankSystem`, `InletAirheatingSystem` and `KEASystem` extend the classes in Figure 6.2. The first three systems represent sub-systems of the APCM, the last system represents the whole APCM.

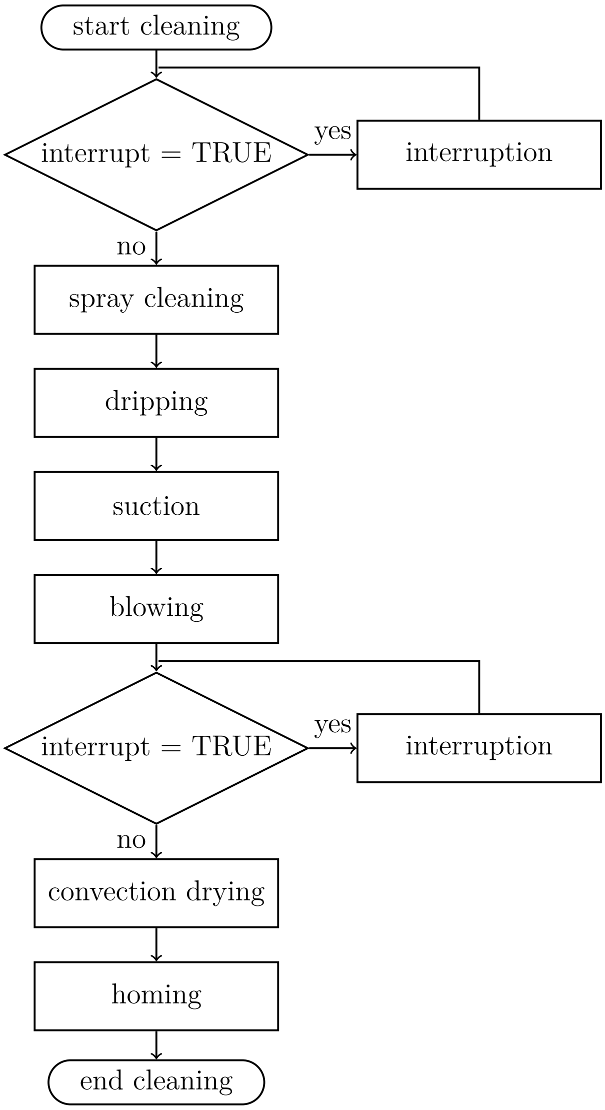


Figure 8.6: Program flow chart of the cleaning process. It is possible to interrupt the cleaning process before the steps *spray cleaning* and *convection drying*. Figure based on [Fuhr23b].

- The four data structures `ST_KEAState`, `ST_TankState`, `ST_CleaningChamberState` and `ST_InletAirheaterState` represent the state of the systems. This includes information such as the setpoint state of all the actuators included in the systems and sensor values.
- The system's error state is communicated by `ST_KEAErrorState`, `ST_TankErrorState`, `ST_CleaningChamberErrorState` as well as `ST_InletAirheaterErrorState`. They include a variable representing the global error of the system and variables, that show which part of the system is in error.
- The parameters of the active cleaning program such as the durations of cleaning process steps are included in the `ST_KEAProgramState` structure.
- The structure `ST_KEALocalSetParameters` is used for the interaction with the machine operator. Here it is possible to select different predefined cleaning programs, to reset the errors, open the door and specify the numbers of parts that are loaded to the APCM.
- Using the `ST_SetKEAProgramManual` it is possible to change the cleaning process parameters manually if the operator does not want to use one of the predefined cleaning programs.

These additional data structures and the basic data structures presented in Section 6.3 are included in the communication layer using OPC UA as the IT-OT-data exchange service. Additional methods need to be added to adapt the automation program to the APCM MAFAC KEA:

- The systems `TankSystem` and `InletAirHeaterSystem` that implement *store energy inherently* include the `CheckAlgorithmPermission()` methods that only allow the usage of the tank heater and the air heater if the system's temperature is in the temperature limits and if the system is not in error state.
- The system `KEASystem` integrates additional methods that model the cleaning process steps *spray cleaning*, *dripping*, *suction*, *blowing*, *convection drying* and *homing* implementing the control procedures for the actuators active in that step.
- The `SystemFlowControlInterruptProcess()` of the `KEASystem` implements the possibility to interrupt the process before *spray cleaning* and *convection drying*. Figure 8.6 shows the program flow of the cleaning process which is executed in the *working* state.
- In the `TimeToNextInterruption()` method of the `KEASystem` the remaining time until the next possible interruption is calculated.

With these additional methods and the basic methods the execution of DR measures on the APCM MAFAC KEA can be ensured.

8.3.2 IT-framework eta_utility

The *eta_utility* framework is the IT-framework used to deploy the DR control algorithm and for the interaction between the DR control algorithm and the APCM. The content and structure of this section follows [Gros22b], where we presented the framework. The framework is implemented in Python and consists of six machine modules illustrated in Figure 8.7:

- The main machine module `eta_x` includes the rolling horizon optimisation functionality `ETAx`, which can execute different optimising algorithms called `agents` and interact with different environments `envs`. The structure of the framework is based on the

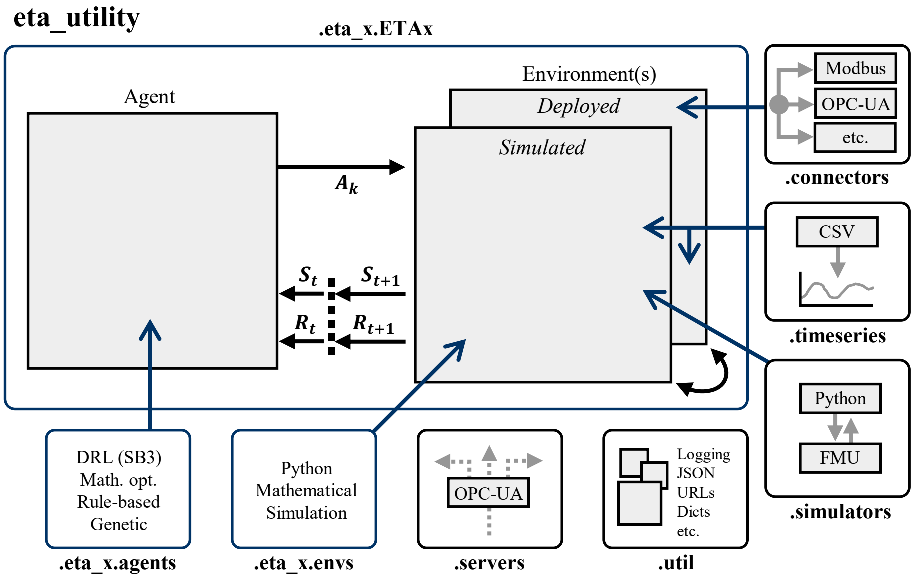


Figure 8.7: Overview of the eta_utility IT-framework. Figure taken from [Gros22b].

stable_baselines3 library [Raff21]. The environment interface is specified in the *Gym* framework developed by OpenAI [Broc16].

- The `connectors` module implements the interaction of the optimisation with external entities. It can establish a connection to different IT-OT-data exchange services such as OPC UA, industrial communication protocols such as Modbus TCP/IP and application programming interfaces (APIs) for the interaction with databases or other IT-frameworks.
- The framework also includes the `timeseries` module to handle time-series data using the Python package *pandas* [Jeff22; McKi10].
- The `simulators` module implements an interface to enable the interaction with simulation models using the Functional Mock-up Unit (FMU) standard [Mode21] based on the Python package *fmpy* [Dass18].
- Using the `servers` module, it is possible to publish data to be used by other systems for example by using the implemented OPC UA server.
- Additional ancillary functions such as logging and data de-serialization are included in the `util` module.

The DR automation architecture only uses the `ETAx` and the `connectors` modules. The DR automation architecture applied to the APCM MAFAC KEA was already presented in our work [Fuhr23a], where we use two interacting `ETAx` environments for the implementation of the DR control algorithm and the OPC UA `connectors` module to implement the IT-OT-communication. We implement the DR optimisation model of the DR control algorithm in the first environment, the second environment represents the APCM. The DR control algorithm consists of the interaction between the agent and the first environment. The DR optimisation model in this environment predicts the behaviour of the APCM and therefore can be seen as a simulation with which the agent interacts.

`ETAx` is based on the idea of performing experiments. An experiment consists of an environment, an agent and a scenario configuration [Gros22b]:

```
{
    "setup": {
        "environment_import": "environment.DRControlAlgorithmKEA",
        "interaction_env_import": "environment.ConnectionKEA",
        "agent_import": "eta_utility.eta_x.agents.MPCBasic"
    },
    "settings": {
        "sampling_time": 10,
        "episode_duration": 1800,
    },
    "environment_specific": {
        "scenario_time_begin": "2021-12-01 06:00",
        "scenario_time_end": "2021-12-01 07:00",
        "scenario_files": [...],
        "model_parameters": {...}, ...
    },
    "interaction_env_specific": {
        "scenario_time_begin": "2021-12-01 06:00",
        "scenario_time_end": "2021-12-01 07:00", ...
    },
    "agent_specific": {
        "timelimit": 8, ...
    }, ...
}
```

Figure 8.8: Parts of the JSON configuration file that contains parameters to configure an experiment in `ETAx`. `setup` defines the used environments and agent, `settings` includes general experiment information and the objects `environment_specific`, `interaction_env_specific` and `agent_specific` contain parameters for the two used environments and the agent. Content based on [Fuhr23a].

- An agent in the `agents` module is an algorithm that enables the control of environments. There are three types of agents available: rule-based control agents, deep reinforcement learning agents and a mathematical agent. We use the mathematical agent for the implementation of the DR control algorithm. The mathematical agent consists of an algorithmic solver and a Python interface that enables the execution of the solver in *eta_utility*. The interface is based on the Python library *Pyomo* [Hart11; Bynu21]. As the solver we use the industrial solver CPLEX by IBM [IBM 19] which can solve MILP using different solving strategies.
- An environment of the `envs` module in `ETAx` is an abstract subclass of OpenAI *Gym* environments [Broc16]. Environments implement a representation of physical systems for example an APCM or they can be used for the interaction of the agent and simulation models. If the environment represents a physical system, `ETAx` uses the `connectors` to communicate with the system. For the interaction with simulation models the `simulators` module is used. We include a `StateVar` object representing each environment variable, such as the APCM's current power consumption or the duration of a cleaning process step. The `StateVar` contains all relevant information about this variable. This includes

the information whether the agent should set the variable as an action, such as a setpoint value, or whether it should be read from scenario data or a simulation model, such as a temperature state value.

- The scenario configuration is based on scenario files which could include price data from an electricity market as an example.

To configure an experiment we use JSON files. A part of an exemplary JSON configuration file as used in [Fuhr23a] can be seen in Figure 8.8. In the object `setup`, the `envs` classes `DRServiceKEA` is selected as the environment that includes the DR optimisation model of the DR control algorithm and `ConnectionKEA` as the environment representing the physical APCM. `MPCBasic` is the selected agent. `settings` defines the sampling time and the duration of the experiment in seconds. `environment_specific` and `interaction_env_specific` define the start and end times of the scenario information to be used, the location of the scenario file and specific model parameters for our DR, amongst others. In `agent_specific` special parameters for the agent can be set for instance a time limit for the execution of the solver.

8.3.3 OPC UA connector for IT-OT-communication

To implement the IT-OT-communication, we use the OPC UA connector from the `connectors` module of the *eta_utility* framework to establish a communication to the APCM's OPC UA server. Thereby we can enable the interaction of the DR control algorithm and the APCM's control system. The *connectors* module is based on `Node` and `Connection` objects [Gros22b].

`Node` objects represent one data point or variable of the system that the environment represents and that `ETAx` interacts with. The `Node` includes all the necessary information to enable a connection to the data source. In the case of the OPC UA connector this includes the server and OPC UA Node ID.

The `Connection` objects represent the connection to one data source, for example an OPC UA server. They can include `Node` objects and implement basic communication methods like reading from a `Node` or writing to a `Node`. In the `Connection` object we specify the needed information to establish a connection, in the case of an OPC UA server, the IP address, user name and password.

The OPC UA connector is parametrised using the automation data dictionary. The automation data dictionary is implemented in the connector configuration file in JSON format, see Figure 8.9. For the parametrisation of the `Connection` object, we include its `name` and specify the `servers` with the information needed to establish a connection. One `Connection` object can include connections to multiple servers. Besides the connection to the APCM, we establish a second connection to the building automation system of the ETA Research Factory to receive the current hall temperature for determining the heat losses of the cleaning liquid to the environment.

For each `Node` object we include its `name`, the `server` where the `Node` is located, the OPC UA identifier (`opc_id`) and the data type (`dtype`). The `name` and `dtype` attributes are the variable name and the data type of the variable in the `ETAx` environment that corresponds to the `Node` object. Thereby we implement a name and data-type mapping between both.

The `observe` attribute defines the `Node` objects with which the connector should interact. This includes reading and writing data. In `actions` we can define that the connector communicates

```
{
    "name": "cleaning machine MAFAC KEA",
    "servers": {
        "KEA": {
            "url": "<IP address>",
            "protocol": "opcua",
            "usr": "",
            "pwd": ""}, ...
    },
    "nodes": [
        {
            "name": "fTankTemperatureKEA",
            "server": "KEA",
            "opc_id": "ns=2;s=KEA.tank.localState.fTankTemperature",
            "dtype": "float"
        }, ...
      ],
    "observe": [
        "fTankTemperatureKEA", ...
    ],
    "actions": {
        "init": {
            "bTankHeaterAlgorithmModeActivated": true, ...
        },
        "close": {
            "bTankHeaterAlgorithmModeActivated": false, ...
        }
    }
}
```

Figure 8.9: Excerpt of the automation data dictionary implemented in JSON file format. It includes the necessary information to establish a connection to the APCM's OPC UA server like the IP address, user name and password. It also holds information to identify the OPC UA nodes that should be used and actions to be executed at the start and end of an experiment. Figure based on [Gros22a].

Table 8.4: The automation data dictionary includes all the information that is needed by the DR control algorithm for *store energy inherently*, based on [Fuhr23a]. For each item the object, the data structure and the variable of the automation program are shown as well as their corresponding symbols in the DR optimisation model of the DR control algorithm.

Object / data structure / variable	Symbol	Information
KEASystem		
ST_KEAProgramState		
fMassWorkpiece	m_{parts}	mass of workpiece
ST_KEAState		
nWorkpieces	$\mathcal{N}_{\text{parts}}$	number of workpieces
TankSystem		
ST_StoreEnergyInherentlyState		
fRatedPower	P_{heat}	tank heater's rated power
fProcessValue	T_{start}	current tank temperature
aFlexibilityLimits[1]	T_{lb}	tank temperature's lower limit
aFlexibilityLimits[2]	T_{ub}	tank temperature's upper limit
ST_TankState		
fCleaningFluidSpecificHeatCapacity	$c_{\text{p,fluid}}$	cleaning fluid's specific heat capacity
fTankVolume	V_{tank}	tank volume
fCleaningFluidDensity	ρ_{fluid}	cleaning fluid's density
TankSystem/TankHeater		
ST_Control		
bSetStatusOnAlgorithm	h_k	Boolean setpoint of the tank heater
bAlgorithmModeActivated	-	activate algorithm mode
BuildingAutomation/Ambient*		
ST_LocalState		
fHallTemperature	T_{env}	current hall temperature

*The hall temperature is communicated by the building automation via OPC UA.

Table 8.5: The automation data dictionary includes all the information that is needed by the DR control algorithm for *interrupt process*, based on [Fuhr23a]. For each item, the object, the data structure and the variable of the automation program are shown as well as their corresponding symbols in the DR optimisation model of the DR control algorithm.

Object / data structure / variable	Symbol	Information
KEASystem		
ST_Control		
bAlgorithmModeActivated	-	activate algorithm mode
ST_InterruptProcess		
bInterruptProcessAlgorithm	i_k	Boolean setpoint for interruption
ST_InterruptProcessState		
nInterruptedState	n_{start}*	interrupted process step
stepDuration		
fDurationSprayCleaning	d_{clean}**	duration of *spray cleaning*
fDurationDripping	d_{clean}**	duration of *dripping*
fDurationSuction	d_{clean}**	duration of *suction*
fDurationBlowing	d_{clean}**	duration of *blowing*
fDurationConvectionDrying	d_{dry}	duration of *convection drying*
fInterruptionCountdown	d_{start}	remaining event duration
stepPowerConsumption		
fPowerConsumptionOperational	P_{int}	power consumption *operational*
fPowerConsumptionSprayCleaning	P_{clean}	power consumption *spray cleaning*
fPowerConsumptionConvectionDrying	P_{dry}	power cons. *convection drying*
ST_KEAState		
nKEAOperatingState	n_{start}*	current process step

*The current and interrupted process step values are both used by the DR control algorithm to calculate n_{start}.
**The event *cleaning*, with the duration d_{clean}, combines the process steps *spray cleaning*, *dripping*, *suction* and *blowing*, because an interruption is not possible between these steps.

specific values to `Node` objects. In the example, we set `bTankHeaterAlgorithmModeActivated` to true before the execution of the experiment (`init` in Figure 8.9) and to false after its end (`close` in Figure 8.9). The content of the automation data dictionary for the use case of the APCM MAFAC KEA for *store energy inherently* are shown in Table 8.4 and for *interrupt process* in Table 8.5.

8.4 Adaptation and implementation of the demand response control algorithm

To implement the DR service, the general model presented in Section 7.2 needs to be adapted to the APCM MAFAC KEA. This chapter uses the equations of the model presented in our work [Fuhr23a], where we also conducted the experimental parameter identification presented

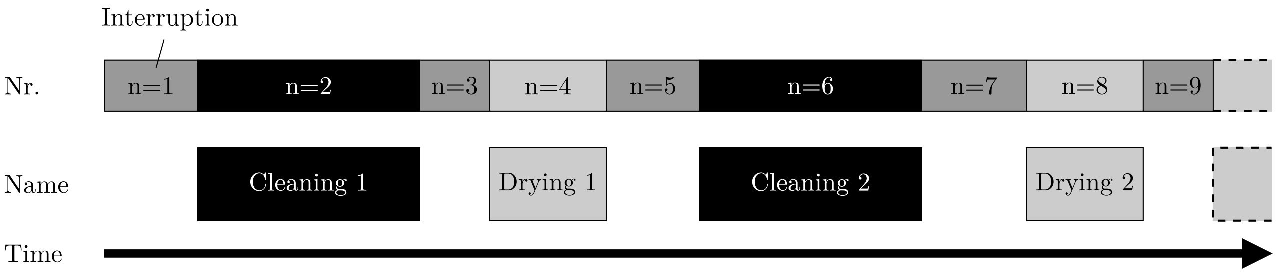


Figure 8.10: Process events as part of the event-based model. A cleaning process starts with an interruption ($n = 1$) before the cleaning is executed. Other interruptions are possible between cleaning and drying ($n = 3, 7$) and after drying before the start of a new cleaning process ($n = 5, 9$).

in Section 8.4.4. Comparing to [Fuhr23a], in this work some variable symbols are modified to maintain consistency in the present work and for a better understanding.

Since only the tank heater is selected for *store energy inherently*, the DR optimisation model includes only one machine module, such that $L = 1$, and (7.1) can be simplified to

$$f(\mathbf{d}, \mathbf{h}) = \sum_{n=1}^{N} \sum_{k=s_n}^{s_n+d_n-1} P_n C_k + P_{\text{heat}} \sum_{k=0}^{K} h_k C_k \tag{8.1}$$

where $P_{\text{heat}} \in \mathbb{R}$ is the rated power of the tank heater, $h_k \in \{0, 1\}$ is the tank heater state at time step k and $\mathbf{h} = (h_0, \dots, h_K) \in \{0, 1\}^{K+1}$ are all tank heater states. Since only one machine module is considered the vector $\mathbf{h}$ replaces the matrix $\mathbf{H}$.

8.4.1 Event-based sub-model of the cleaning process for interrupt process

The potential analysis in Section 8.2.2 shows that the DR potential for *interrupt process* is high before the process steps *spray cleaning* and *convection drying*. Section 8.3 presents how to implement the possibility to interrupt the process before these two steps. For the event-based sub-model, the event *cleaning* combines the process steps *spray cleaning*, *dripping*, *suction* and *blowing*, since an interruption is not possible between these steps. The process step *convection drying* is modelled in one event *drying*.

Figure 8.10 illustrates the event sequence. It starts with an interruption, followed by the *cleaning* event. Before *drying* there is another interruption. The last event $n = 5$ of the first cleaning process is the interruption before the *cleaning* event of the second cleaning process. In the model it is possible to connect multiple cleaning processes in a row as shown in the figure. The first, last and every uneven event $n = 1, 3, \dots, N$ are interruptions. The *cleaning* event $n = 2, 6, \dots, N-3$ is the second event and the *drying* event $n = 4, 8, \dots, N-1$ the fourth event of each process. Both are repeated every four events. Note that $N \in \{5, 9, 13, \dots\}$, since each process consists of four events and the first event of the first process ($n = 1$) is always an interruption.

Every process event has a specific power consumption

$$P_n = \begin{cases} P_{\text{int}} & , \forall n = 1, 3, \ldots, N \\ P_{\text{clean}} & , \forall n = 2, 6, \ldots, N-3 \\ P_{\text{dry}} & , \forall n = 4, 8, \ldots, N-1 \end{cases}, \tag{8.2}$$

where $P_{\text{int}} \in \mathbb{R}_{\geq 0}$ is the APCM's power consumption during interruption, $P_{\text{clean}} \in \mathbb{R}_{\geq 0}$ during *cleaning* and $P_{\text{dry}} \in \mathbb{R}_{\geq 0}$ during *drying*.

Since *cleaning* and *drying* cannot be interrupted the event duration is fixed by

$$d_n = \begin{cases} 0 & , \forall n < n_{\text{start}} \\ d_{\text{start}} & , \forall n = 2, 4, \ldots, N-1 \text{ with } n = n_{\text{start}} \\ d_{\text{clean}} & , \forall n = 2, 6, \ldots, N-3 \text{ with } n > n_{\text{start}} \\ d_{\text{dry}} & , \forall n = 4, 8, \ldots, N-1 \text{ with } n > n_{\text{start}} \end{cases} \tag{8.3}$$

to either the remaining duration $d_{\text{start}} \in \mathbb{N}$ of the active event, the total duration $d_{\text{clean}} \in \mathbb{N}$ of the process steps combined in the *cleaning* event or the duration of *convection drying* $d_{\text{dry}} \in \mathbb{N}$.

The starts s_n and durations d_n of past events are zero, following (7.8) and (7.9). The starts of all other events are calculated by (7.10) and all modelled cleaning processes must be terminated before or at the time step S and in the optimisation horizon K, based on (7.11). At the end of each cleaning process a minimum duration for unloading and loading of the APCM $d_{\text{load}} \in \mathbb{N}$ must be included, such that

$$d_n \geq \min(d_{\text{load}}, S), \forall n = 5, 9, \ldots, N \text{ with } n \geq n_{\text{start}}. \tag{8.4}$$

The last event is always an interruption.

8.4.2 Time-based sub-model of the tank heating system for store energy inherently

The DR potential analysis for *store energy inherently* in Section 8.2.1 shows that the tank and air heater have a high potential for DR. However, the air heater can only be used for short-term DR measures such as internal load shifting because it is only active during the short process step *convection drying*. Since the DR control algorithm models the interaction with an electricity market, we only consider the tank heater to be used for *store energy inherently* by the DR control algorithm. The tank heating system's process value (see Section 7.2.2) is the temperature of the cleaning liquid in the tank, which we refer to as the tank temperature $T_k \in \mathbb{R}$ in the following. The tank temperature is calculated by

$$T_{k+1} = T_k + \Delta T_k^+ - \Delta T_k^-, \forall k = 0, \ldots, K-1, \tag{8.5}$$

where $\Delta T_k^+ \in \mathbb{R}_{\geq 0}$ is the temperature increase if the tank heater is on and $\Delta T_k^- \in \mathbb{R}_{\geq 0}$ is the temperature loss during the cleaning process and in stand-by. We set

$$T_0 = T_{\text{start}}, \tag{8.6}$$

where $T_{\text{start}} \in \mathbb{R}_{\geq 0}$ denotes the current tank temperature. The tank temperature is bounded by

$$T_{\text{lb}} \leq T_k \leq T_{\text{ub}}, \forall k = 0, \ldots, K, \tag{8.7}$$

where $T_{\mathrm{lb}} \in \mathbb{R}_{\geq 0}$ is the lower and $T_{\mathrm{ub}} \in \mathbb{R}_{\geq 0}$ is the upper bound of the tank temperature.

To obtain the behaviour of the tank temperature during the cleaning process, we need to model the temperature losses of the cleaning liquid in the APCM. This includes all the cleaning liquid in the APCM, including the cleaning liquid in the tank as well as in the cleaning chamber. We assume that the cleaning liquid is a closed homogeneous system, since the losses of cleaning liquid to the environment by evaporation and the amount of fresh water introduced to the APCM are very small and hence negligible. Since the tank is not hermetically sealed, we assume that the heat exchange between the tank heater and the liquid is isobaric. The work applied to the cleaning liquid by the pump is negligible, as the temperature influence is minimal, so that we assume that there is no exchange of work between the system and its environment.

We calculate the temperature rise ΔT_k^+ if the tank heater is on based on the first law of thermodynamics for closed systems [Step13, p 84]

$$\frac{\mathrm{d}E}{\mathrm{d}t} = \frac{\mathrm{d}Q}{\mathrm{d}t} + \frac{\mathrm{d}W}{\mathrm{d}t}, \tag{8.8}$$

where $E \in \mathbb{R}$ is the system's energy, $Q \in \mathbb{R}$ is the energy supplied to the system as heat, $W \in \mathbb{R}$ is the work done on the system by its surroundings and $t \in \mathbb{R}$ is the time. As already mentioned, we assume that there is no work applied to the cleaning liquid such that $\frac{\mathrm{d}W}{\mathrm{d}t} = 0$. Considering this and that the mechanical energy of the system does not change, we can simplify (8.8) to

$$\frac{\mathrm{d}U}{\mathrm{d}t} = \frac{\mathrm{d}Q}{\mathrm{d}t}, \tag{8.9}$$

where $U \in \mathbb{R}$ is the inner energy of the system. If we have a continuous process, as in this use case, based on [Step13, p 85], we can modify (8.9) to

$$\frac{\mathrm{d}U}{\mathrm{d}t} = \dot{Q}, \tag{8.10}$$

where the heat flow $\dot{Q} \in \mathbb{R}$ is constant in the time period $\mathrm{d}t$. To calculate a change of the inner energy from one state to the other we use the integrated form of (8.10), that is

$$\Delta U = \dot{Q}\Delta t, \tag{8.11}$$

where $\Delta U \in \mathbb{R}$ is the system's change of inner energy and $\Delta t \in \mathbb{R}$ is the considered time interval. For liquids and solids

$$\Delta U = c_{\mathrm{p}} m \Delta T \tag{8.12}$$

applies [Step13, p 112], where $c_{\mathrm{p}} \in \mathbb{R}$ is the specific heat capacity of the system's material, $m \in \mathbb{R}_{\geq 0}$ is its mass and $\Delta T \in \mathbb{R}$ is its temperature change.

Assuming that no heat is supplied from outside and the tank heater has an efficiency of nearly one, we can set the heat flow to $\dot{Q} \cong P_{\mathrm{heat}}$. Considering $m_{\mathrm{fluid}} = V_{\mathrm{tank}}\rho_{\mathrm{fluid}}$ and using (8.11) and (8.12) we can calculate the temperature rise by

$$\Delta T_k^+ = \frac{P_{\mathrm{heat}}\delta}{c_{\mathrm{p,fluid}} V_{\mathrm{tank}} \rho_{\mathrm{fluid}}} h_k, \; \forall k = 0, \ldots, K, \tag{8.13}$$

where $\delta \in \mathbb{N}$ is the duration of one time step in seconds, $c_{\mathrm{p,fluid}} \in \mathbb{R}$ is the specific heat capacity of the cleaning fluid, $m_{\mathrm{fluid}} \in \mathbb{R}_{\geq 0}$ is the mass of the cleaning fluid, $V_{\mathrm{tank}} \in \mathbb{R}_{\geq 0}$ is the volume of the tank and $\rho_{\mathrm{fluid}} \in \mathbb{R}_{\geq 0}$ is the density of the cleaning fluid.

We want to estimate the temperature loss ΔT_k^- experimentally. The total temperature loss ΔT_k^- consists of the temperature loss to the APCM's environment $\Delta T_{\mathrm{env},k}^- \in \mathbb{R}_{\geq 0}$ and the heat loss during cleaning $\Delta T_{\mathrm{clean},k}^- \in \mathbb{R}_{\geq 0}$, which is composed of the specific heat loss to the parts $\Delta T_{\mathrm{parts},k}^- \in \mathbb{R}_{\geq 0}$ and the general heat loss $\Delta T_{\mathrm{spray},k}^- \in \mathbb{R}_{\geq 0}$ to the APCM.

To determine the temperature loss to the environment $\Delta T_{\mathrm{env},k}^-$, we use the simplified heat exchange equation, based on [VDI13, p 33],

$$\dot{Q} = \frac{T_{\mathrm{f1}} - T_{\mathrm{f2}}}{R}, \tag{8.14}$$

where $T_{\mathrm{f1}} \in \mathbb{R}$ is the temperature of the first fluid, $T_{\mathrm{f2}} \in \mathbb{R}$ is the temperature of the second fluid and $R \in \mathbb{R}$ is the thermal resistance. For the heat flow $\dot{Q}_{\mathrm{env}} \in \mathbb{R}$ from the cleaning liquid to the environment (8.14) becomes

$$\dot{Q}_{\mathrm{env}} = \frac{T_k - T_{\mathrm{env}}}{R_{\mathrm{env}}}, \tag{8.15}$$

where $T_{\mathrm{env}} \in \mathbb{R}$ is the temperature of the production hall and $R_{\mathrm{env}} \in \mathbb{R}$ is the thermal resistance between liquid and the production hall. Using (8.11), (8.12) and (8.15), yields

$$\Delta U_{\mathrm{env},k} = c_{\mathrm{p,fluid}} m_{\mathrm{fluid}} \Delta T_{\mathrm{env},k}^- = \frac{T_k - T_{\mathrm{env}}}{R_{\mathrm{env}}} \delta, \tag{8.16}$$

where $\Delta U_{\mathrm{env},k} \in \mathbb{R}$ is the system's change of inner energy caused by the temperature loss to the environment. Then we can calculate the temperature loss to the environment by

$$\Delta T_{\mathrm{env},k}^- = \frac{T_k - T_{\mathrm{env}}}{c_{\mathrm{p,fluid}} m_{\mathrm{fluid}} R_{\mathrm{env}}} \delta. \tag{8.17}$$

The determination of the thermal resistance R is not straight forward since it includes determining the thermal resistance between the cleaning fluid, the tank and the environment. These thermal resistances depend on multiple unknown factors, for example the exact properties of the material the tank is made of. To facilitate the estimation of the temperature loss to the environment $\Delta T_{\mathrm{env},k}^-$, we define $\beta_{\mathrm{env}} \in \mathbb{R}_{\geq 0}$ to

$$\beta_{\mathrm{env}} \cong \frac{1}{c_{\mathrm{p,fluid}} m_{\mathrm{fluid}} R_{\mathrm{env}}} \tag{8.18}$$

and thereby can approximate the temperature loss to the environment by

$$\Delta T_{\mathrm{env},k}^- = \beta_{\mathrm{env}} \delta \left(T_k - T_{\mathrm{env}} \right). \tag{8.19}$$

Using (8.19) it is possible to estimate β_{env} experimentally, see Section 8.4.4.

We determine the temperature loss to the parts by the change of the inner energy $\Delta U_{\mathrm{parts}} \in \mathbb{R}$ based on (8.12) and assume that the parts are heated up until they have the same temperature as the cleaning liquid, such that $T_k = T_{\mathrm{parts},k}$. The temperature loss can then be calculated by

$$\dot{Q}_{\mathrm{parts}} \delta = \Delta U_{\mathrm{parts},k} = c_{\mathrm{p,parts}} \mathcal{N}_{\mathrm{parts}} m_{\mathrm{parts}} (T_{\mathrm{parts}} - T_k), \tag{8.20}$$

where $\dot{Q}_{\mathrm{parts}} \in \mathbb{R}$ is the heat flow from the cleaning liquid to the parts, $T_{\mathrm{parts}} \in \mathbb{R}$ is the temperature of the parts before loading, outside of the APCM, $c_{\mathrm{p,parts}} \in \mathbb{R}$ is the parts' specific heat capacity, $\mathcal{N}_{\text{parts}} \in \mathbb{N}$ is the number of parts in the cleaning tray and $m_{\text{parts}} \in \mathbb{R}_{\geq 0}$ is the mass of one part.

We assume that the parts have the same temperature as the environment when loaded, such that $T_{\text{parts}} = T_{\text{env}}$. By using (8.12) and (8.20), leading to

$$\Delta U_{\text{parts},k} = c_{\text{p,fluid}} m_{\text{fluid}} \Delta T^{-}_{\text{parts},k} = c_{\text{p,parts}} \mathcal{N}_{\text{parts}} m_{\text{parts}} (T_{\text{env}} - T_k), \tag{8.21}$$

we can calculate the temperature loss to the parts by

$$\Delta T^{-}_{\text{parts},k} = -\frac{c_{\text{p,parts}}}{c_{\text{p,fluid}} V_{\text{tank}} \rho_{\text{fluid}}} \left(T_k - T_{\text{env}}\right) \mathcal{N}_{\text{parts}} m_{\text{parts}}. \tag{8.22}$$

For the temperature loss to the parts $\Delta T^{-}_{\text{parts},k}$ we define $\beta_{\text{parts}} \in \mathbb{R}_{\geq 0}$ such that

$$\beta_{\text{parts}} \cong -\frac{c_{\text{p,parts}}}{c_{\text{p,fluid}} V_{\text{tank}} \rho_{\text{fluid}}}. \tag{8.23}$$

We approximate the temperature loss from the cleaning luiquid to the parts by

$$\Delta T^{-}_{\text{parts},k} = \beta_{\text{parts}} \left(T_k - T_{\text{env}}\right) \mathcal{N}_{\text{parts}} m_{\text{parts}}. \tag{8.24}$$

We define the temperature loss of the cleaning liquid to the APCM itself while the spray cleaning is running based on (8.14) by

$$\dot{Q}_{\text{spray}} = \frac{T_k - T_{\text{env}}}{R_{\text{spray}}}, \tag{8.25}$$

where $R_{\text{spray}} \in \mathbb{R}$ is the thermal resistance between the liquid and the APCM. For the temperature loss to the APCM $\Delta T^{-}_{\text{spray},k}$ analogous to (8.18) we define $\beta_{\text{spray}} \in \mathbb{R}_{\geq 0}$ such that

$$\beta_{\text{spray}} \cong \frac{1}{c_{\text{p,fluid}}\ V_{\text{tank}} \rho_{\text{fluid}} R_{\text{spray}}}. \tag{8.26}$$

Analogous to (8.19), we approximate the temperature loss from liquid to the APCM by

$$\Delta T^{-}_{\text{spray},k} = \beta_{\text{spray}} \delta \left(T_k - T_{\text{env}}\right). \tag{8.27}$$

Using (8.22) and (8.27) we calculate the temperature loss during cleaning by

$$\Delta T^{-}_{\text{clean},k} = \left(\beta_{\text{spray}} \delta + \beta_{\text{parts}} \mathcal{N}_{\text{parts}} m_{\text{parts}}\right) \left(T_k - T_{\text{env}}\right), \forall k = 0, \dots, K. \tag{8.28}$$

Thus, the total temperature losses can be summarized by

$$\Delta T^{-}_{k} = \beta_{\text{env}} \delta \left(T_k - T_{\text{env}}\right) + \begin{cases} \Delta T^{-}_{\text{clean},k}, & \text{if } s_n \leq k < s_n + d_n,\ \forall n = 2, 6, \dots, N-3 \\ 0, & \text{else} \end{cases}, \tag{8.29}$$

for all $k = 0, \dots, K$, such that we can calculate the temperature losses during cleaning (where $s_n \leq k < s_n + d_n$, $\forall n = 2, 6, \dots, N-3$ applies) and during the other process steps and machine states.

Table 8.6: Starts and durations of the five events of an exemplary cleaning process.

n	s_n	d_n
1	0	1
2	1	4
3	5	2
4	7	2
5	9	1

8.4.3 Adaptation of the demand response control algorithm for implementation

For the implementation of the DR control algorithm we use the Python library *Pyomo* [Hart11; Bynu21]. This library cannot handle an objective function with variable summation limits as in (8.1). Therefore, we introduce the binary variable $a_{n,k} \in \{0,1\}$ which has value one during the execution of a process step n at time step k and value zero otherwise. We include $a_{n,k}$ in the objective function and modify (8.1) to

$$f(\mathbf{A},\mathbf{h}) = \sum_{n=1}^{N}\sum_{k=0}^{K} a_{n,k}P_nC_k + P_{\text{heat}}\sum_{k=0}^{K} h_kC_k \tag{8.30}$$

such that the sum $\sum_{k=s_n}^{s_n+d_n-1} P_nC_k$ with variable limits is replaced by $\sum_{k=0}^{K} a_{n,k}P_nC_k$. Further, the optimisation variable $\mathbf{d}$ is replaced by

$$\mathbf{A} = \begin{bmatrix} a_{1,0} & \cdots & a_{1,K} \\ \vdots & \ddots & \vdots \\ a_{N,0} & \cdots & a_{N,K} \end{bmatrix} \in [0,1]^{N\times(K+1)}, \tag{8.31}$$

which includes the starts and durations of all N process events.

We use an example to illustrate the construction of $\mathbf{A}$. Assuming the starts and durations of five events given in Table 8.6, we construct the matrix

$$\mathbf{A} = \begin{bmatrix} 1&0&0&0&0&0&0&0&0&0 \\ 0&1&1&1&1&0&0&0&0&0 \\ 0&0&0&0&0&1&1&0&0&0 \\ 0&0&0&0&0&0&0&1&1&0 \\ 0&0&0&0&0&0&0&0&0&1 \end{bmatrix}, \tag{8.32}$$

where the five rows represent the five events and the ten columns represent the time steps from $k = 0$ to $k = 10$. Using this exemplary matrix we illustrate that, by construction of the matrix, the sum of the n-th row, representing an event n, is equal to the sum from start s_n to the end $s_n + d_n - 1$ of the event and also equal to the event's duration d_n, meaning

$$\sum_{k=0}^{K} a_{n,k} = \sum_{k=s_n}^{s_n+d_n-1} a_{n,k} = d_n,\ \forall n = 1,\dots,N. \tag{8.33}$$

To construct $a_{n,k}$ we introduce a second binary variable $\tilde{a}_{n,k} \in \{0,1\}$ which is true during and before the execution of a process step n at the time step k and can be combined in the matrix

$$\tilde{\mathbf{A}} = \begin{bmatrix} \tilde{a}_{1,0} & \cdots & \tilde{a}_{1,K} \\ \vdots & \ddots & \vdots \\ \tilde{a}_{N,0} & \cdots & \tilde{a}_{N,K} \end{bmatrix} \in [0,1]^{N\times(K+1)}. \tag{8.34}$$

Using $\tilde{\mathbf{A}}$ we guarantee the correct order of the events. For our example we get the matrix

$$\tilde{\mathbf{A}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}. \tag{8.35}$$

We set

$$\sum_{k=0}^{K} \tilde{a}_{n,k} = \sum_{i=1}^{n} d_i,\ \forall n = 1,\dots,N, \tag{8.36}$$

such that $\tilde{a}_{n,k}$ is always true during and before the execution of a process event n. With

$$\tilde{a}_{n,0} = \begin{cases} b_n, & \forall n = n_{\text{start}},\dots,N \\ 0, & \forall n < n_{\text{start}} \end{cases}, \tag{8.37}$$

where $b_n \in \{0,1\}$, and

$$\tilde{a}_{n,k} \leq \tilde{a}_{n,k-1},\ \forall n = 1,\dots,N; k = 0,\dots,K, \tag{8.38}$$

we guarantee $\tilde{a}_{n,k} = 0$ for past events. The variable $\tilde{a}_{n,k}$ can be zero if event durations are set to zero such that $b_n = 0$ and can be one if the duration is greater than zero such that $b_n = 1$. Using $\tilde{a}_{n,k}$, we construct $a_{n,k}$ by

$$a_{n,k} = \begin{cases} \tilde{a}_{n,k}, & n = 1 \\ \tilde{a}_{n,k} - \tilde{a}_{n-1,k}, & \forall n = 2,\dots,N \end{cases},\ \forall k = 0,\dots,K. \tag{8.39}$$

The MPC algorithm needs one value that is aligned with the interruption variable of the machine automation. Therefore, we introduce

$$i_k = \sum_{n\in\{1,3,\dots,N\}} a_{n,k},\ \forall k = 0,\dots,K, \tag{8.40}$$

which is true during interruption and selects the second value i_1 of the vector as a control value that is communicated to the APCM.

Including $a_{n,k}$ in the calculation of the temperature losses during cleaning (8.28) results in the bilinear equation

$$\Delta T^{-}_{\text{clean},k} = \sum_{i=0}^{n} \left(\beta_{spray}\delta + \beta_{\text{parts}}\mathcal{N}_{\text{parts}}m_{\text{parts}}\right)\left(T_{n,k} - T_{\text{env}}\right) a_{n,k},\ \forall n = 2,6,\dots,N-3; \forall k = 0,\dots,K, \tag{8.41}$$

which is infeasible for the used solver. To avoid the bilinear product $T_{n,k}a_{n,k}$ and hence restore the feasibility of the optimisation, we linearise it using

$$0 \leq z_{n,k} \leq T_{\text{ub}} a_{n,k} \tag{8.42}$$

and

$$T_k - T_{\text{ub}}(1 - a_{n,k}) \leq z_{n,k} \leq T_k \tag{8.43}$$

where $z_{n,k} \in \mathbb{R}$, such that

$$z_{n,k} = \begin{cases} T_k, & \text{for } a_{n,k} = 1 \\ 0, & \text{else} \end{cases}. \tag{8.44}$$

Then, the temperature loss during cleaning results in

$$\Delta T^{-}_{\text{clean},k} = \sum_{i=0}^{n} \left(\beta_{spray}\delta + \beta_{\text{parts}}\mathcal{N}_{\text{parts}}m_{\text{parts}}\right)\left(z_{n,k} - a_{n,k}T_{\text{env}}\right), \forall n = 2, 6, \ldots, N{-}3, \forall k = 0, \ldots, K, \tag{8.45}$$

which is feasible.

Finally, considering the objective function (8.30), we obtain the specific optimisation problem

$$\begin{aligned} \min_{\mathbf{A},\mathbf{h}} \quad & f\left(\mathbf{A}, \mathbf{h}\right) \\ \text{such that} \quad & (7.8),\ (7.9),\ (7.10),\ (7.11),\ (8.3),\ (8.4), \\ & (8.7),\ (8.36),\ (8.37),\ (8.38) \text{ and } (8.39) \text{ hold.} \end{aligned} \tag{8.46}$$

8.4.4 Experimental parameter identification

To estimate the temperature loss parameters in (8.18), (8.23) and (8.26) we performed a parameter identification in [Fuhr23a] consisting of three experiments. The production hall's temperature was approximately constant with $T_{\text{env}} \approx 22.5\,°\text{C}$ during all three experiments. We first determined the heat loss to the environment $\Delta T^{-}_{\text{env},k}$ by heating up the tank to $T \approx 64\,°\text{C}$ and letting it cool down for 127 minutes, see the top of Figure 8.11. We then calculated β_{env} with (8.17) to

$$\beta_{\text{env}} = -\frac{\text{avg}(\Delta T_{\text{env}})}{\text{avg}(T_k - T_{\text{env}})} \cong 1.67 \cdot 10^{-5}, \tag{8.47}$$

using the average measured temperature differences per second.

To estimate the temperature loss to the parts $\Delta T^{-}_{\text{parts},k}$ we executed one cleaning process with $\mathcal{N}_{\text{parts}} = 42$ parts and a mass of $m_{\text{parts}} = 0.262\,\text{kg}$ each, visualized in the middle of Figure 8.11. The part is a control plate for a hydraulic pump. We calculate β_{parts} with (8.22) to

$$\beta_{\text{parts}} = -\frac{\text{avg}(\Delta T_{\text{parts}} - \Delta T_{\text{spray}} - \Delta T_{\text{env}})}{\text{avg}(T_k - T_{\text{parts}})\mathcal{N}_{\text{parts}}m_{\text{parts}}} \cong 1.03 \cdot 10^{-5}, \tag{8.48}$$

also using the average temperature differences per second.

Finally, to estimate the temperature loss of the liquid to the APCM $\Delta T^{-}_{\text{spray}}$ we run a cleaning process with an empty tray and calculate β_{spray} with (8.27) to

$$\beta_{\text{spray}} = -\frac{\text{avg}(\Delta T_{\text{spray}} - \Delta T_{\text{env}})}{\text{avg}(T_k - T_{\text{env}})} \cong 1.72 \cdot 10^{-5}. \tag{8.49}$$

We can see this experiment in the bottom of Figure 8.11.

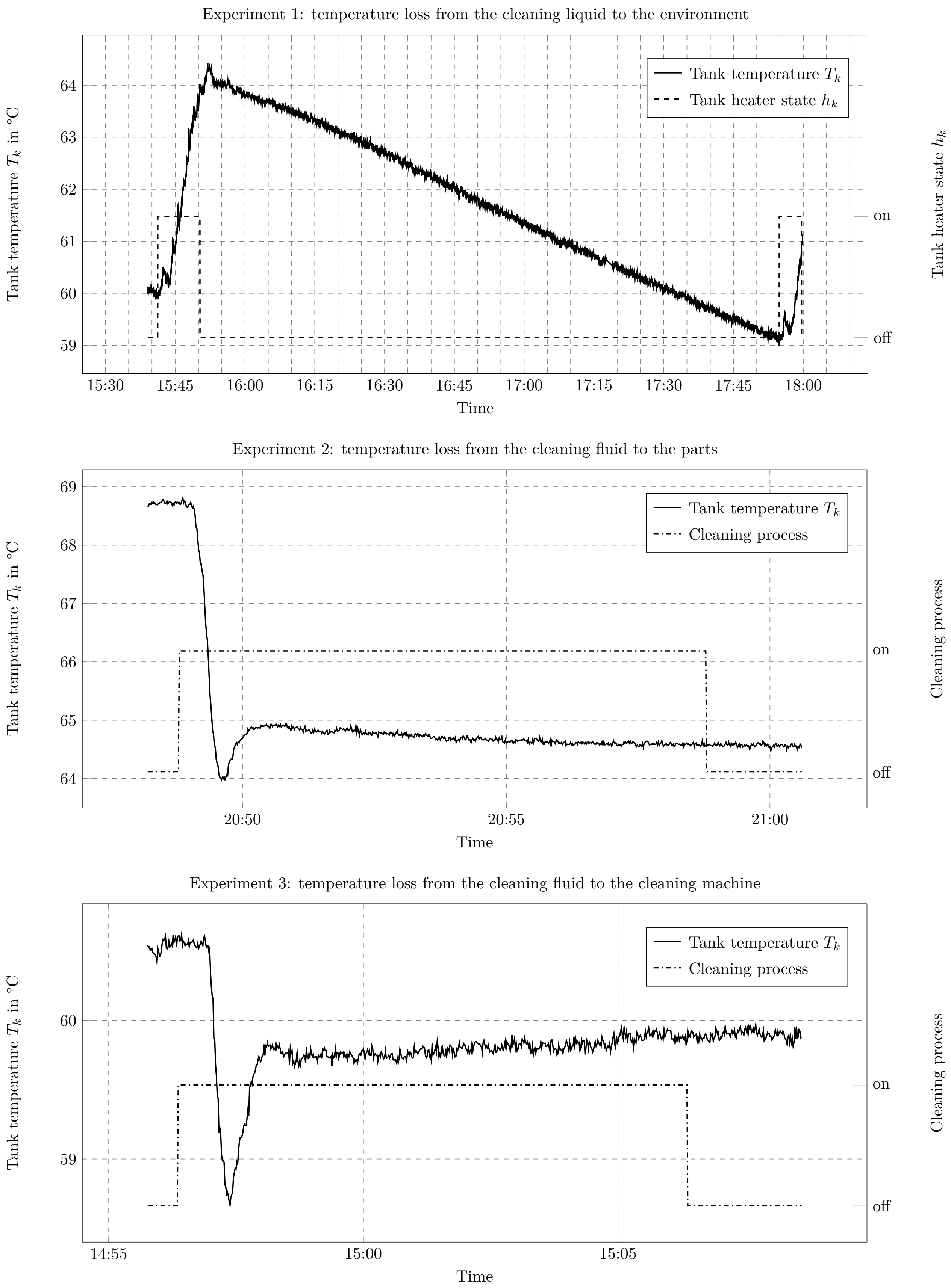


Figure 8.11: Estimation of the temperature loss of the cleaning liquid in the tank to the environment (top), to the parts (middle) and to the APCM (bottom). Data from [Fuhr23a].

Table 8.7: Set-up of the field test. Values taken from [Fuhr23a].

Parameter	Symbol	Value	Unit
Number of process elements	N	5	
Time step duration	δ	10	s
Prediction scope	K	1800	s
Fixed end time of the process	S	1800	s
Duration for loading	d_{load}	120	s
Tank volume	V_{tank}	320	L
Cleaning fluid specific heat capacity	$c_{\text{p,fluid}}$	4.19	$\frac{\text{kJ}}{\text{kg K}}$
Cleaning fluid density	ρ_{fluid}	1	$\frac{\text{kg}}{\text{L}}$
Tank temperature lower bound	T_{lb}	55	°C
Tank temperature upper bound	T_{ub}	65	°C
Number of parts	$\mathcal{N}_{\text{parts}}$	42	
Mass of part	m_{parts}	0.262	kg
Rated power tank heater	P_{heat}	10	kW
Accumulated rated power interruption	P_{int}	0.2	kW
Accumulated rated power cleaning	P_{clean}	3.43	kW
Accumulated rated power drying	P_{dry}	9.35	kW

8.5 Field test: execution of the demand response service

To verify the functionality of the automation structure and the DR control algorithm we execute a field test on the APCM MAFAC KEA. We already published the experiment of this section in [Fuhr23a]. We use the exemplary price data from European Power Exchange (EPEX) Spot [EPEX21] as input for the DR control algorithm. We take the prices of December 1st, 2021, 6:00 am to 9:00 am and scale the interval of price changes from 15 min to 5 min. This results in price data for one hour for the field test.

Since the cleaning process has a total duration of 12 min a price interval of 15 min would probably only lead to shifting the beginning of the complete cleaning process but not result in process interruptions. The assumption of a price change every 5 minutes is reasonable, since the conversion of the European electricity market to a 5-minute interval is currently being discussed [Rolo22]. In [Gros22a] we use a 15-minute interval for the control of the tank heater and show that applying DR measures on APCMs is also possible with the current design of the electricity market.

The DR control algorithm is executed on a PC with Intel Core i-5 6200U CPU and 8 GB of RAM in 10 s intervals. We set the prediction scope of the model to 30 min and examine one single cleaning process that must be terminated within 30 min. We reduce the fixed end time of the process S by 10 s every time the MPC is executed to guarantee the end of the process after 30 min. The part that is used in the field test is the control plate for a hydraulic pump which we mentioned in Section 8.4.4. We run the test with a part tray carrying 42 parts. An overview of the parameters of the field test are shown in Table 8.7.

Figure 8.12 displays the results of the field tests. After five minutes, the energy price, shown in the upper diagram, turns negative, causing the solver to optimise for a rise in total power usage during the negative-price phase. By turning on the interruption, the DR control algorithm

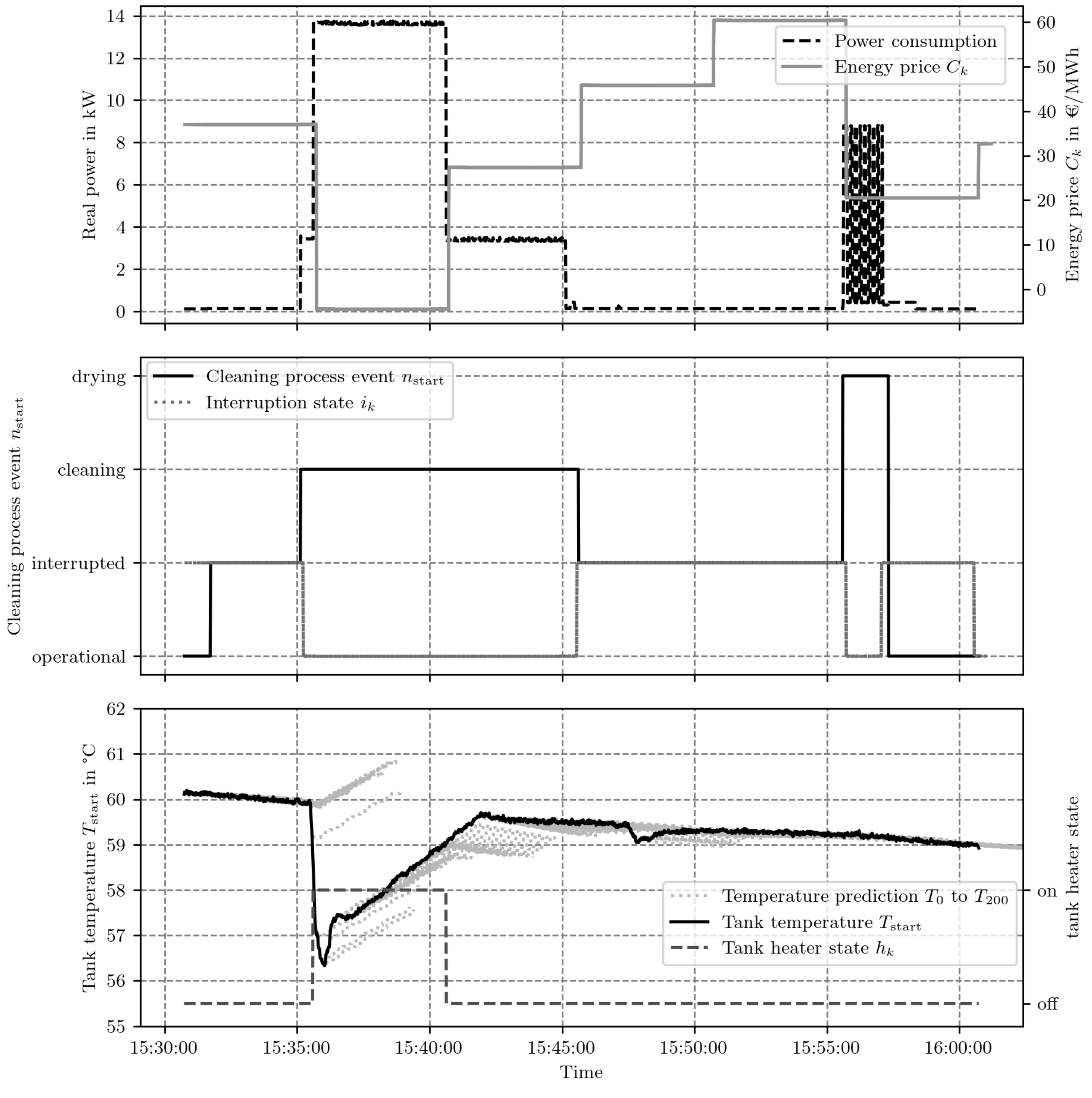


Figure 8.12: Results of the 30-minute field test. The APCM's measured overall power consumption is depicted in the upper diagram along with the energy price C_k. The active step of the cleaning process n_{start} and the boolean interruption variable i_k are shown in the centre diagram. The measured tank temperature T_{start}, the tank temperature prediction T_0 to T_{200} for 200 s as well as the tank heater state h_k are displayed in the lower diagram. Figure adapted from [Fuhr23a].

Table 8.8: Measured activation duration d_{meas}, average power P_{meas} and energy consumption W_{meas} of the tank heater and the process steps as results of the field test. The total energy consumption is 1483.64 Wh. Data from [Fuhr23a].

	d_{meas} in s	P_{meas} in kW	W_{meas} in Wh
Tank heater	302	10.04	842.58
Interruption 1	203	0.12	6.82
Spray cleaning	600	3.19	525.50
Dripping	10	0.12	0.33
Suction	10	0.24	0.67
Blowing	10	0.13	0.35
Interruption 2	598	0.12	20.08
Convection drying	90	3.24	81.10

delays the beginning of *spray cleaning* and uses the negative price, which is depicted in the centre diagram.

As seen in the lower diagram, the tank heater is turned on only shortly after the cleaning process begins. The tank heater is turned off as the energy price rises, and *spray cleaning* continues. When *spray cleaning* is completed, the DR control algorithm again interrupts the process to postpone *convection drying* to a later time window at a lower price. After 30 minutes, the complete cleaning process is finished.

The temperature forecast, based on the parameters identified in Section 8.4.4, is displayed in grey in the lower diagram. The anticipated grey temperature values perfectly match the actual tank temperature shown in black during the first five minutes and at the end of the cleaning process. The temperature increase that occurs during operation of the tank heater is also precisely anticipated (between 15:36 and 15:41).

There are dead times, after the heater is turned on and before the heat transmission finishes, that the model is unable to replicate. The measured temperature drop at the beginning of *spray cleaning* cannot be represented by the tank temperature model. This is because the spray pump's activation causes a transient response. The tank temperature stabilises when the pump starts after around two minutes.

The forecast is accurate again once transient processes (such as the tank heater's dead times and the cleaning fluid's response) have settled. This is also true when considering the cleaning process as a whole.

The field tests shows that the execution of DR measures is possible after applying the DRAAD method. Table 8.8 shows the measured average power and energy consumption of the tank heater, which was used for *store energy inherently*, and the process steps *convection drying* and *spray cleaning*, which were interrupted during *interrupt process*. While the measured power of the tank heater is only 0.04 kW higher than the rated power value, which we have used in the DR potential analysis, its average measured power is 420.36 Wh higher than the value we have used during the DR potential analysis, compare Table 8.2. The DR control algorithm exploits the negative electricity price and turns the tank heater on for 302 s which is almost twice as long as during the average cleaning process.

Table 8.9: Comparison of the absolute and relative technical DR potential for *store energy inherently*, P_{flex} and Φ_{P}, to the measured absolute and relative power change in the field test, P_{meas} and $\Phi_{\mathrm{P,meas}}$. Data from [Fuhr23a].

	P_{flex} in kW	Φ_{P}	P_{meas} in kW	$\Phi_{\mathrm{P,meas}}$
Tank heater	10	0.48	10.04	0.49
Air heater	8	0.39	0	0.00
Total	18	0.87	10.04	0.49

Table 8.10: Comparison of the technical DR potential for *interrupt process*, W_{flex} and Φ_{W}, to the measured energy shift in the field test, W_{meas} and $\Phi_{\mathrm{W,meas}}$. Data from [Fuhr23a].

	W_{flex} in Wh	Φ_{W}	W_{meas} in Wh	$\Phi_{\mathrm{W,meas}}$
Spray cleaning	571.67	0.77	525.5	0.71
Convection drying	160.42	0.22	81.1	0.11
Total	732.09	0.99	606.60	0.82

In comparison to the DR potential analysis, the average power during *spray cleaning* is only 0.24 kW lower and the energy consumption is also only slightly lower, with a difference of 39.97 Wh. Since this process step has a fixed duration implemented in the automation program, there is no difference between the expected duration used in the DR potential analysis and the measured duration in the field test. Hence, the used DR potential is almost as high as expected.

The average power during *convection drying* is 6.11 kW lower and the energy consumption is 79.32 Wh less than the respective values of the DR potential analysis, compare Table 8.3. The DR potential analysis uses the rated power of all machine modules active during *convection drying*, but the air heater, as the main electrical consumer of these machine modules, is only active for 33 s during drying. This is 22 s less than in the reference cleaning process, resulting in lower values for average power and energy consumption. The time difference is due to differences in the automation program. The reference process used for the DR potential analysis in [Fuhr23b] is based on an early version of the DR automation program, which contained an error in the air heater's control logic such that it was switched on longer than in the field test in [Fuhr23a]. However, the lower activation time does not change the qualitative result of the DR potential analysis. Even with an activation time of 33 s, the process step *convection drying* has a high technical DR potential and is suitable to be used for *interrupt process*.

Table 8.9 and Table 8.10 show the comparison of the technical DR potential for *store energy inherently* and *interrupt process* as result of the DR potential analysis and the power change and energy shift measured in the field test. In the field test, using *store energy inherently* a power change of 49 % could be archived in comparison to a technical DR potential of 87 %. Applying *interrupt process* leads to an energy shift of 82 %, comparing a technical DR potential of 99 %. The relative measured power change $\Phi_{\mathrm{P,meas}}$ is relative to the APCM's total rated power $P_{\mathrm{total}} = 20.7\,\mathrm{kV\,A}$. The relative measured energy shift $\Phi_{\mathrm{W,meas}}$ is relative to the total energy consumption of the reference cleaning process $W_{\mathrm{total}} = 739.75\,\mathrm{Wh}$.

The total energy costs of the field tests are 0.0262 €, excluding the *operational* state and only considering the energy consumption during the *working* state, when the cleaning process is running. The electricity price changes five times during the cleaning process, resulting in six different energy prices. To compare the energy costs of the cleaning process during execution of the DR measures with the standard cleaning process, we assume that the energy price is constant for the standard cleaning process.

We take the field test's average electricity price of 116.72 €/MWh as constant electricity price, which is calculated based on a sampling interval of one second. Using the measured average power during the process steps from Table 8.8 and the durations of the reference process the energy demand of the reference process can be calculated. During the reference process the tank heater is only activated for 152 s, the durations of the process steps are equal in the reference process and the field test.

Using the measured power values P_{meas} of the electrical consumers and the activation times d_f in the reference process, results in a total energy consumption of 1038.24 Wh and total energy costs of 0.1211 € for the reference process. This includes the energy consumption of the actuators active during the cleaning process as well as the energy consumption of the tank heater. The field test's total energy consumption of 1483.63 Wh is 42.90 % higher than the energy consumption of the reference process, but the field test's electricity costs of 0.0262 € are 78.45 % lower. This shows, that the DR control algorithm is able to exploit the varying electricity price.

For the field test we used EPEX Spot price data of 2021. While the number of hours with negative electricity prices increased during the last years to 298 in 2020 [Agor21, p 41], there were only 139 hours with negative prices in 2021 [Agor22, p 16] and none in 2022 [Agor23, p 41]. However, the electricity price continued to fluctuate between 186 €/MWh and 319 €/MWh [Agor23, p 34] in the average daily profile in 2022, so there is still an incentive to use DR.

8.6 Summary

This chapter showed the successful application of the DRAAD method to the APCM MAFAC KEA in the ETA Research Factory. This consists of the DR potential analysis of the APCM, the implementation of a DR architecture and the implementation of a DR control algorithm. The DR automation architecture and the DR control algorithm interact to execute the DR measures *store energy inherently* and *interrupt process* simultaneously in a field test.

The DR analysis identified the DR potential of the DR measures *store energy inherently* and *interrupt process*. No energy measurements were needed, since the average activation durations of machine modules that are not time-controlled were measured by running the reference process multiple times. The absolute technical DR potential for *store energy inherently* is $P_{\text{flex}} = 18\,\text{kW}$ and the relative DR potential is $\Phi_{\text{P}} = 0.87$. The absolute DR potential of *interrupt process* is $W_{\text{flex}} = 732.08\,\text{Wh}$ and the DR ratio $\Phi_{\text{W}} = 0.99$.

The DR automation program was implemented in the PLC of the investigated APCM. OPC UA was used as the IT-OT-data exchange service, such that the DR data model is implemented as an OPC UA data model. In the experimental set-up, the IT-framework and the DR control algorithm were deployed on an external computer. The data published by an OPC UA server can also be accessed by an IT-operating system installed on an IPC, as shown in [Fuhr22b].

Hence, the DR automation architecture is adaptable to different IT-architectures. The IT-framework *eta_utility* enables the communication between different IT-systems.

The DR control algorithm implements the DR measures *store energy inherently* and *interrupt process*, which was validated in a field test. The DR control algorithm was adapted to the APCM MAFAC KEA by including a forecast of the cleaning liquid's temperature. In contrast to the generalised DR control algorithm, which uses a white-box model, the implemented DR control algorithm is a grey-box approach, since it includes a statistical parameter estimation for modelling the cleaning liquid's temperature losses.

The field test showed that the DR control algorithm is able to use the full DR potential of the tank heater for *store energy inherently* and exploit the negative electricity price. The energy amount that is used for the DR measure *interrupt process* by interrupting the process step *spray cleaning* matches the results of the DR potential analysis. For the process step *convection drying* only half of the technical DR potential can be exploited.

The energy consumption of the cleaning process in the field test is 42.90 % higher than the energy consumption of the reference process, but 78.45 % energy costs can be saved due to the negative electricity price. Although no negative electricity prices were denoted at EPEX Spot in 2022, the daily price spread is still high and the application of DR highly relevant.

9 Summary, conclusion and outlook

The aim of this thesis was to develop a method to implement an automation architecture that enables the execution of demand response (DR) measures on aqueous parts cleaning machines (APCMs), the so called demand response automation architecture design (DRAAD) method. By implementing DR measures on production machines such as APCMs, and as such facilitating the integration and use of renewable energies, industry can make its contribution to the reduction of greenhouse gas emissions and move towards a sustainable economy.

The DRAAD method consists of a DR potential analysis, developed in Chapter 5, a DR automation architecture, including a DR automation program and a DR data model, specified in Chapter 6, and a DR control algorithm, modelled in Chapter 7. Chapter 8 presented the application of the DRAAD method to the APCM MAFAC KEA in the Energy Technologies and Applications in Production (ETA) Research Factory.

In Section 3.2, 19 design criteria for the DRAAD method were specified. As can be seen in Table 9.1, the concept of the DRAAD method fulfilled all 19 design criteria. The transfer of the DRAAD method to other APCMs was not demonstrated in the frame of this work, since it was only applied to one APCM. However, in [Fuhr22a; Fuhr22b] we used an early version of the DR automation architecture to execute a DR control algorithm on an air-cooled chiller in the ETA research factory and thereby showed that the DR automation architecture as the main part of the DRAAD method is transferable to other machines.

In Section 5.1, prior to developing a DR potential analysis for APCMs the DR measures *store energy inherently* and *interrupt process* were determined to be suitable for the application to APCMs, based on the list of DR measures defined in [VDI5207-1]. The DR potential analysis of APCMs, presented in this thesis, extended the approach of [Abel16], that analysed the DR potential of machine tools for *store energy inherently*. It adapted the DR potential analysis for *store energy inherently* to APCMs in Section 5.2 and added a DR potential analysis for *interrupt process* in Section 5.3.

The cyber-physical production system presented in Section 6.2 was the framework for the development of the DR automation architecture. It included the physical APCM, a digital twin, consisting of a digital master, a digital shadow and digital services, external entities such as an electricity market as well as the cyber-physical interface that represents the communication in the cyber-physical production system.

The object-oriented design of the DR automation program and the structure of the DR data model were specified in the automation data specification, presented in Section 6.3, as part of the digital master. The DR automation program included objects and functionality to enable the safe execution of the DR measures *store energy inherently* and *interrupt process*.

The second part of the digital master was the automation data dictionary described in Section 6.3. This was used by an interpreter to map the variables of the DR data model to the variables of the DR control algorithm and thereby, in combination with an information technology (IT)-operating technology (OT)-data exchange service, enabled the IT-OT-communication between the DR automation program and the DR control algorithm.

Table 9.1: The table shows that all design criteria are fulfilled by the DRAAD method's concept. The application does not verify the transfer of the DRAAD method to multiple APCMs, but the DR automation architecture is used in building automation systems, see [Fuhr22a; Fuhr22b].

	Design criteria		Concept	Application
DR potential analysis	C1.1	machine level	x	x
	C1.2	manual execution	x	x
	C1.3	technical DR potential	x	x
	C1.4	scalable & transferable	x	
	C1.5	process criticality	x	x
DR automation architecture	C2.1	both DR measures implemented*	x	x
	C2.2	adaptable deployment architecture	x	x
	C2.3	observable	x	x
	C2.4	controllable	x	x
	C2.5	IT-communication enabled	x	x
	C2.6	object-oriented (transferable)	x	(x)
	C2.7	functional safety	x	x
	C2.8	IT-OT-communication enabled	x	x
	C2.9	naming convention	x	x
DR control algorithm	C3.1	both DR measures implemented*	x	x
	C3.2	scalable & transferable	x	
	C3.3	process stability	x	x
	C3.4	white- or grey-box model	x	x
	C3.5	integrate electricity prices	x	x

* The DR measures *store energy inherently* and *interrupt process* must be implemented.

Chapter 7 showed a generalised model predictive control (MPC) algorithm, integrating a mixed integer linear programming (MILP) model, as a DR control algorithm for the execution of DR measures on APCMs. The DR control algorithm included two sub-models to implement the DR measures *store energy inherently* and *interrupt process*.

The DR potential analysis in Section 8.2, as part of the application of the DRAAD method to the APCM MAFAC KEA, revealed that the APCM has a high relative DR potential of 87 % for *store energy inherently* and 99 % for *interrupt process*. Section 8.3 showed the design of the DR automation program and its IT-OT-interaction with the DR control algorithm using Open Platform Communications Unified Architecture (OPC UA) as the IT-OT-data exchange service and the IT-framework *eta_utility*.

The DR control algorithm, adapted to the APCM MAFAC KEA in Section 8.4, extended the generalised MPC algorithm by a detailed model to predict the temperature of the cleaning liquid in the APCM's tank. For a stable execution of the cleaning process, this temperature must stay within defined boundaries, which was included in the DR control algorithm using constraints. The model of the liquid's temperature included statistical heat flow parameters identified in experiments.

The field test in Section 8.5 showed the successful execution of the DR measures *store energy inherently* and *interrupt process* simultaneously. The DR control algorithm observed the APCM's machine states, which were integrated in the OPC UA DR data model, and used the DR methods implemented in the DR automation program to control the APCM. Consequently all parts of the DR automation architecture were active during the field test.

This thesis showed that it is possible to apply DR measures to production machines, not only by controlling auxiliary machine modules, but also by controlling process relevant machine modules and interrupting the process. By considering constraints in the DR control algorithm and integrating functional safety functions in the DR automation program, it is possible to execute DR measures while considering process stability and safety at the same time.

Further, it can be concluded that service-oriented communication architectures, such as OPC UA, enable an IT-OT-interaction and facilitate the integration of production machines in IT-systems. This could be used not only to apply DR control algorithms but also to execute other complex monitoring and control algorithms for use cases such as the identification of defect parts during production, forecasts of energy-consumption or optimal production planning.

The full DR potential in industry can be utilised only if many industrial production machines are enabled for DR. Standardised DR data models can make the manual adjustment of DR control algorithms to individual plants unnecessary. In this thesis and in [Fuhr22a; Fuhr22b], we have shown that the DR automation architecture including the DR data model is transferable to different machines. A future standardisation of the DR data model in an OPC UA Companion Specification can massively increase the transferability of the approach to industry.

If machine manufacturers integrate DR in the automation systems of their machines, new business cases arise. Machine manufacturers could sell the integration of DR into their machines as an additional feature. Machine users can save electricity costs by executing DR measures based on varying electricity prices. Note that this requires electricity companies that offer varying prices or that trade the pooled DR potential of multiple production machines on energy markets. It is also possible to use DR such that renewable electricity produced on-site is better utilised, thus reducing the factory's carbon footprint.

Future research should investigate the application of the DRAAD method to other APCMs and examine whether an adaption to other production machines is possible. Also the impact on the cleaning quality by interrupting the cleaning process should be analysed in detail. The DR control algorithm should be tested in field tests with longer durations. Also, multiple APCMs could be integrated. An interaction of the DR automation architecture with other DR control algorithms for example DR control algorithms based on artificial intelligence (AI) is a promising research topic.

Bibliography

[Abel16] E. Abele, P. Schraml and D. Moog. 'Electric Load Management on Machine Tools'. In: *Procedia CIRP* 55 (2016), pp. 164–169.

[Abik17] J. B. Abikarram and K. McConky. 'Real time machine coordination for instantaneous load smoothing and photovoltaic intermittency mitigation'. In: *Journal of Cleaner Production* 142 (2017), pp. 1406–1416.

[Agor21] Agora Energiewende, ed. *Die Energiewende im Corona-Jahr: Stand der Dinge 2020: Rückblick auf die wesentlichen Entwicklungen sowie Ausblick auf 2021.* 2021. URL: `https://www.agora-energiewende.de/veroeffentlichungen/die-energiewende-im-corona-jahr-stand-der-dinge-2020/` (visited on 23/02/2023).

[Agor22] Agora Energiewende, ed. *Die Energiewende in Deutschland: Stand der Dinge 2021: Rückblick auf die wesentlichen Entwicklungen sowie Ausblick auf 2022.* 2022. URL: `https://www.agora-energiewende.de/veroeffentlichungen/die-energiewende-in-deutschland-stand-der-dinge-2021/` (visited on 23/02/2023).

[Agor23] Agora Energiewende, ed. *Die Energiewende in Deutschland: Stand der Dinge 2022: Rückblick auf die wesentlichen Entwicklungen sowie Ausblick auf 2023.* 2023. URL: `https://www.agora-energiewende.de/veroeffentlichungen/die-energiewende-in-deutschland-stand-der-dinge-2022/` (visited on 23/02/2023).

[Alba07] M. H. Albadi and E. F. El-Saadany. 'Demand Response in Electricity Markets: An Overview'. In: *2007 IEEE Power Engineering Society General Meeting.* IEEE, 2007, pp. 1–5.

[Aliz16] M. I. Alizadeh, M. Parsa Moghaddam, N. Amjady, P. Siano and M. K. Sheikh-El-Eslami. 'Flexibility in future power systems with high renewable penetration: A review'. In: *Renewable and Sustainable Energy Reviews* 57 (2016), pp. 1186–1193.

[Ange19] K. Angerer, A. Hohmann, V. Seifert and C. Tammer. *Ökologische und ökonomische Bewertung des Ressourcenaufwands: Reinigungstechnologien in der industriellen Produktion.* Berlin: VDI Zentrum Ressourceneffizienz, 2019. URL: `https://www.ressource-deutschland.de/service/publikationen/detailseite/studie-reinigungstechnologien/` (visited on 14/04/2023).

[Auri09] J. C. Aurich, F. Fallenstein, T. Eyrisch and T. Hielscher. 'Bauteilreinigung im Spritzverfahren'. In: *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb* 104.9 (2009), pp. 775–779.

[Bäns21] K. Bänsch, J. Busse, F. Meisel, J. Rieck, S. Scholz, T. Volling and M. G. Wichmann. 'Energy-aware decision support models in production environments: A systematic literature review'. In: *Computers & Industrial Engineering* 159 (2021).

[Bart18] L. Barth, N. Ludwig, E. Mengelkamp and P. Staudt. 'A comprehensive modelling framework for demand side flexibility in smart grids'. In: *Computer Science - Research and Development* 33.1-2 (2018), pp. 13–23.

[Beck23a] Beckhoff Automation. *Identifiers for variables and instances*. 2023. URL: `https://infosys.beckhoff.com/english.php?content=../content/1033/tc3_plc_intro/12073947403.html&id=` (visited on 14/04/2023).

[Beck23b] Beckhoff Automation. *TF3810: TwinCAT 3 Neural Network Inference Engine*. 2023. URL: `https://www.beckhoff.com/en-us/products/automation/twincat/tfxxxx-twincat-3-functions/tf3xxx-tc3-measurement/tf3810.html` (visited on 13/02/2023).

[Beck23c] Beckhoff Automation. *TF4100: TwinCAT 3 Controller Toolbox*. 2023. URL: `https://www.beckhoff.com/en-us/products/automation/twincat/tfxxxx-twincat-3-functions/tf4xxx-tc3-controller/tf4100.html` (visited on 13/02/2023).

[Biel16] K. Biel and C. H. Glock. 'Systematic literature review of decision support models for energy-efficient production planning'. In: *Computers & Industrial Engineering* 101 (2016), pp. 243–259.

[Bilz13] M. Bilz, J. Mankiewicz and S. Motschmann. *Markt- und Trendanalyse in der industriellen Teilereinigung 2012*. Berlin: Frauenhofer-Allianz Reinigungstechnik, 2013.

[Bles09] L. T. Blessing and A. Chakrabarti. *DRM, a Design Research Methodology*. London: Springer London, 2009.

[Brag23] S. A. Braga dos Santos, J. M. Soares, G. C. Barroso and B. de Athayde Prata. 'Demand response application in industrial scenarios: A systematic mapping of practical implementation'. In: *Expert Systems with Applications* 215 (2023).

[Broc16] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang and W. Zaremba. 'OpenAI Gym'. In: *arXiv* (2016).

[Broy19] M. Broy. 'Grundlagen der Objektorientierung'. In: *Logische und methodische Grundlagen der Programm- und Systementwicklung*. Ed. by M. Broy. Wiesbaden: Springer Vieweg, 2019, pp. 381–433.

[Bruc19] D. Bruckner, M.-P. Stanica, R. Blair, S. Schriegel, S. Kehrer, M. Seewald and T. Sauter. 'An Introduction to OPC UA TSN for Industrial Communication Systems'. In: *Proceedings of the IEEE* 107.6 (2019), pp. 1121–1131.

[Bynu21] M. L. Bynum, G. A. Hackebeil, W. E. Hart, C. D. Laird, B. L. Nicholson, J. D. Siirola, J.-P. Watson and D. L. Woodruff. *Pyomo-optimization modeling in python*. Third. Vol. 67. Springer Science & Business Media, 2021.

[Cama07] E. F. Camacho and C. Bordons. *Model predictive control*. Advanced textbooks in control and signal processing. London: Springer London Limited, 2007.

[CIGR11] CIGRE. *Demand side integration*. Paris: CIGRÉ, 2011.

[Claa11] A. Claassen, S. Rohjans and S. Lehnhoff Member. 'Application of the OPC UA for the Smart Grid'. In: *2011 2nd IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies*. IEEE, 2011, pp. 1–8.

[Conn16] D. Connolly, H. Lund and B. V. Mathiesen. 'Smart Energy Europe: The technical and economic impact of one potential 100% renewable energy scenario for the European Union'. In: *Renewable and Sustainable Energy Reviews* 60 (2016), pp. 1634–1653.

[Dass18] Dassault Systèmes. *FMPy*. GitHub respository. 2018. URL: `https://github.com/CATIA-Systems/FMPy` (visited on 21/10/2018).

[Deng15] R. Deng, Z. Yang, M.-Y. Chow and J. Chen. 'A Survey on Demand Response in Smart Grids: Mathematical Models and Approaches'. In: *IEEE Transactions on Industrial Informatics* 11.3 (2015), pp. 570–582.

[Deve22] B. U. Deveci, H. Bas, E. Ummak, O. Albayrak and P. Unal. 'A Thorough Analysis and Comparison of Data Communication Protocols Used in Industry 4.0: the Case of Smart-CNC'. In: *2022 9th International Conference on Future Internet of Things and Cloud (FiCloud)*. IEEE, 2022, pp. 199–206.

[DIN16593-1] DIN. 16593-1: *Reference Model for Industrie 4.0 Service Architectures: Part 1: Basic Concepts of an Interaction-based Architecture*

[DIN8580] DIN. 8580:2022-12. *Fertigungsverfahren: Begriffe, Einteilung*

[DIN8592] DIN. 8592:2003-09. *Fertigungsverfahren Reinigen: Einordnung, Unterteilung, Begriffe*

[Ditt09] R. Dittmar and B.-M. Pfeiffer. *Modellbasierte prädiktive Regelung: Eine Einführung für Ingenieure*. München, Wien: Oldenbourg, 2009.

[Döbe18] I. Döbel, M. Leis, M. Molina Vogelsang, D. Neustroev, H. Petzka, A. Riemer, S. Rüping, A. Voss, M. Wegele and J. Welz. *Maschinelles Lernen: Eine Analyse zu Kompetenzen, Forschung und Anwendung*. Ed. by Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e.V. 2018. URL: `https://www.bigdata-ai.fraunhofer.de/content/dam/bigdata/de/documents/Publikationen/Fraunhofer_Studie_ML_201809.pdf` (visited on 12/02/2023).

[Duft17] C. Dufter, A. Guminski, C. Orthofer, S. von Roon and A. Gruber, eds. *Lastflexibilisierung in der Industrie - Metastudienanalyse zur Identifikation relevanter Aspekte bei der Potenzialermittlung*. 2017.

[Durk06] J. B. Durkee. *Management of industrial cleaning technology and processes*. 1. ed. Amsterdam: Elsevier, 2006.

[EPEX21] EPEX SPOT SE. *Market Data: Continuous > 15min > DE*. Paris, 2021. URL: `https://www.epexspot.com/en/market-data` (visited on 01/12/2021).

[euro21] eurostat. *Supply, transformation and consumption of electricity*. 2021. URL: `https://ec.europa.eu/eurostat/databrowser/view/nrg_cb_e/` (visited on 14/12/2021).

[Fran11] U. Frank, J. Papenfort and D. Schütz. 'Real-time capable software agents on IEC 61131 systems – Developing a tool supported method'. In: *IFAC Proceedings Volumes* 44.1 (2011), pp. 9164–9169.

[Fuhr21] D. Fuhrländer-Völker, M. Lindner and M. Weigold. 'Design Method for Building Automation Control Programs to Enable the Energetic Optimization of Industrial Supply Systems'. In: *Procedia CIRP* 104 (2021), pp. 229–234.

[Fuhr22a] D. Fuhrländer-Völker, F. Borst, L. Theisinger, H. Ranzau and M. Weigold. 'Modular data model for energy-flexible cyber-physical production systems'. In: *Procedia CIRP* 107 (2022), pp. 215–220.

[Fuhr22b] D. Fuhrländer-Völker, M. Lindner and M. Weigold. 'Umsetzung und Implementierung am Beispiel von Speicher-Wandler-Systemen'. In: *Markt- und Stromsystem, Managementsysteme und Technologien energieflexibler Fabriken.* Ed. by A. Sauer, H. U. Buhl, A. Mitsos and M. Weigold. Energieflexibilität in der deutschen Industrie. Stuttgart: Fraunhofer Verlag, 2022, pp. 176–181.

[Fuhr23a] D. Fuhrländer-Völker, B. Grosch and M. Weigold. 'Modelling And Control Of Aqueous Parts Cleaning Machines For Demand Response'. In: *Proceedings of the Conference on Production Systems and Logistics: CPSL 2023* (2023), pp. 790–800.

[Fuhr23b] D. Fuhrländer-Völker, J. Magin and M. Weigold. 'Automation architecture for harnessing the demand response potential of aqueous parts cleaning machines'. In: *Production Engineering* (2023).

[Gart23a] Gartner Glossary. *Information Technology (IT).* 2023. URL: `https://www.gartner.com/en/information-technology/glossary/it-information-technology` (visited on 08/02/2023).

[Gart23b] Gartner Glossary. *Operational Technology (OT).* 2023. URL: `https://www.gartner.com/en/information-technology/glossary/operational-technology-ot` (visited on 08/02/2023).

[Gil22] S. Gil, G. D. Zapata-Madrigal, R. García-Sierra and L. A. Cruz Salazar. 'Converging IoT protocols for the data integration of automation systems in the electrical industry'. In: *Journal of Electrical Systems and Information Technology* 9.1 (2022).

[Graß14a] M. Graßl and G. Reinhart. 'Evaluating Measures for Adapting the Energy Demand of a Production System to Volatile Energy Prices'. In: *21st CIRP Conference on Life Cycle Engineering* 15 (2014), pp. 129–134.

[Graß14b] M. Graßl, E. Vikdahl and G. Reinhart. 'A Petri-net Based Approach for Evaluating Energy Flexibility of Production Machines'. In: *Enabling Manufacturing Competitiveness and Economic Sustainability.* Ed. by M. F. Zaeh. Cham: Springer International Publishing, 2014, pp. 303–308.

[Graß15] M. Graßl. *Bewertung der Energieflexibilität in der Produktion.* Forschungsberichte / IWB. München: Utz, 2015.

[Gros19] B. Grosch, T. Weitzel, N. Panten and E. Abele. 'A metaheuristic for energy adaptive production scheduling with multiple energy carriers and its implementation in a real production system'. In: *Procedia CIRP* 26th CIRP Life Cycle Engineering Conference, West Lafayette, IN (USA).80 (2019), pp. 203–208.

[Gros22a] B. Grosch, D. Fuhrländer-Völker, J. Stock and M. Weigold. 'Cyber-physical production system for energy-flexible control of production machines'. In: *Procedia CIRP* 107 (2022).

[Gros22b] B. Grosch, H. Ranzau, B. Dietrich, T. Kohne, D. Fuhrländer-Völker, J. Sossenheimer, M. Lindner and M. Weigold. 'A framework for researching energy optimization of factory operations'. In: *Energy Informatics* 5.S1 (2022).

[Hahn16] A. Hahn. 'Operational Technology and Information Technology in Industrial Control Systems'. In: *Cyber-security of SCADA and Other Industrial Control Systems*. Ed. by E. J. M. Colbert and A. Kott. Vol. 66. Advances in Information Security. Cham: Springer International Publishing, 2016, pp. 51–68.

[Hart11] W. E. Hart, J.-P. Watson and D. L. Woodruff. 'Pyomo: modeling and solving mathematical programs in Python'. In: *Mathematical Programming Computation* 3.3 (2011), pp. 219–260.

[Heus09] H. Heuser. *Lehrbuch der Analysis*. 17., aktualisierte Auflage. Studium. Wiesbaden: Vieweg + Teubner, 2009.

[Heut19] T. Heutmann, B. Zhou, C. Schommer and R. H. Schmitt. 'Integrated Energy Efficiency in Production'. In: *IFAC-PapersOnLine* 52.13 (2019), pp. 135–140.

[IBM 19] IBM Corporation. *IBM ILOG CPLEX Optimization Studio*. 2019. URL: `https://www.ibm.com/docs/en/icos/` (visited on 05/04/2022).

[IEC61131-1] IEC. 61131-1: *Part 1: General information: Programmable controllers - German Version EN 61131-1:2003*

[IEC61131-3] IEC. 61131-3: *Programmable controllers: Part 3: Programming languages - German Version EN 61131-3:2013*

[IEC62264-1] IEC. 62264-1:2013. *Enterprise-control system integration: Part 1. Models and terminology - German version EN 62264-1:2013*

[IEC62541-1] IEC. 62541-1: *OPC unified architecture: Part 1: Overview and concepts (IEC/TR 62541-1:2010); German version CLC/TR 62541-1:2010*

[IEC62541-14] IEC. 62541-14: *OPC unified architecture: Part 14: PubSub*

[IEC62541-6] IEC. 62541-6:2020. *OPC Unified Architecture: Part 6: Mappings*

[ISOIEC20924] ISO/IEC. 20924: *Internet of things (IoT) – Vocabulary*

[ISOIEC30164] ISO/IEC. 30164:1.0. *Internet of things (IoT) - Edge computing*

[Jeff22] Jeff Reback et al. *pandas-dev/pandas: Pandas 1.4.1*. 2022.

[Jung19] F. Junge. *Ein Beitrag zur Energieeffizienzbewertung von wässrigen Bauteilreinigungsanlagen*. Darmstadt: Universitäts- und Landesbibliothek Darmstadt, 2019.

[Karm18] A. Karmarkar and M. Buchheit, eds. *The Industrial Internet of Things Volume G8: Vocabulary*. 2018. URL: `https://www.iiconsortium.org/pdf/IIC_Vocab_Technical_Report_2.1.pdf` (visited on 08/02/2023).

[Kohn20] T. Kohne, H. Ranzau, N. Panten and M. Weigold. 'Comparative study of algorithms for optimized control of industrial energy supply systems'. In: *Energy Informatics* 3.1 (2020), pp. 1–19.

[Kohn21] T. Kohne, M. Burkhardt, L. Theisinger and M. Weigold. 'Technical and digital twin concept of an industrial heat transfer station for low exergy waste heat'. In: *Procedia CIRP* 104 (2021), pp. 223–228.

[Körn19] M.-F. Körner, D. Bauer, R. Keller, M. Rösch, A. Schlereth, P. Simon, T. Bauernhansl, G. Fridgen and G. Reinhart. 'Extending the Automation Pyramid for Industrial Demand Response'. In: *21st CIRP Conference on Life Cycle Engineering* 81 (2019), pp. 998–1003.

[Lamp13] I. Lampropoulos, W. L. Kling, P. F. Ribeiro and J. van den Berg. 'History of demand side management and classification of demand response control schemes'. In: *2013 IEEE Power & Energy Society General Meeting*. IEEE, 2013, pp. 1–5.

[Lara19] P. Lara, M. Sánchez and J. Villalobos. 'OT Modeling: The Enterprise Beyond IT'. In: *Business & Information Systems Engineering* 61.4 (2019), pp. 399–411.

[Li14] F. Li, C. Legat and B. Vogel-Heuser. 'Extension of Electronic Device Description Language for analysing change impacts in modular automation in manufacturing plants'. In: *Journal of Engineering Design* 25.1-3 (2014), pp. 125–149.

[Lieb15] C. Liebl, R. Popp and M. F. Zäh. 'Energieflexibilität von Produktionsanlagen: Energieflexibilitäts-Kennfelder zur Abschätzung der Energieflexibilitäts-Potentiale unterschiedlicher Anlagen'. In: *wt Werkstattstechnik online* 105.H3 (2015), pp. 136–140.

[Litt70] J. D. C. Little. 'Models and Managers: The Concept of a Decision Calculus'. In: *Management Science* 16.8 (1970), pp. 466–485.

[Liu20] Z. Liu and P. Bellot. 'A configuration tool for MQTT based OPC UA PubSub'. In: *2020 RIVF International Conference on Computing and Communication Technologies (RIVF)*. IEEE, 2020, pp. 1–6.

[Löbe19] F. Löber, D. Moog, R. Langner and E. Abele. 'Kennzahlenbasierte Bewertung technischer Energieflexibilitätspotenziale in der Produktion'. In: *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb* 114.3 (2019), pp. 128–133.

[MAFA12] MAFAC. *KEA-Datenblatt: Technische Daten*. 2012.

[McKi10] W. McKinney. 'Data Structures for Statistical Computing in Python'. In: *Proceedings of the 9th Python in Science Conference*. Ed. by Stéfan van der Walt, Jarrod Millman. Proceedings of the Python in Science Conference. SciPy, 2010, pp. 56–61.

[Merr23] Merriam-Webster.com dictionary. *Information technology*. 2023. URL: `www.merriam-webster.com/dictionary/information%20technolog` (visited on 02/08/2023).

[Mode21] Modelica Association. *Functional Mock-up Interface 2: FMI for Model Exchange and Co-Simulation*. Ed. by Modelica Association. 2021. URL: `https://github.com/modelica/fmi-standard/releases/download/v2.0.3/FMI-Specification-2.0.3.pdf` (visited on 23/03/2022).

[Mono16] L. Monostori, B. Kádár, T. Bauernhansl, S. Kondoh, S. Kumara, G. Reinhart, O. Sauer, G. Schuh, W. Sihn and K. Ueda. 'Cyber-physical systems in manufacturing'. In: *CIRP Annals - Manufacturing Technology* 65.2 (2016), pp. 621–641.

[Mono18] L. Monostori. ‘Cyber-Physical Systems’. In: *CIRP Encyclopedia of Production Engineering*. Ed. by Chatti, Sami and Tolio, Tullio. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018, pp. 1–8.

[OPC10000-80] OPC. 10000-80: *OPC Unified Architecture Field eXchange (UAFX): Part 80: UAFX Overview and Concepts*

[OPC30141] OPC. 30141: *OPC UA for Energy Management*

[OPC40001-1] OPC. 40001-1:1.02.0. *OPC UA for Machinery: Part 1: Basic Building Blocks*

[OPC40010-1] OPC. 40010-1:1.00. *OPC UA for Robotics: Part 1: Vertical Integration*

[OPC40501-1] OPC. 40501-1:1.01.1. *OPC UA for Machine Tools: Part 1: Machine Monitoring and Job Overview*

[Pale11] P. Palensky and D. Dietrich. ‘Demand Side Management: Demand Response, Intelligent Energy Systems, and Smart Loads’. In: *IEEE Transactions on Industrial Informatics* 7.3 (2011), pp. 381–388.

[Pant18] N. Panten, N. Strobel, J. Sossenheimer and E. Abele. ‘Framework for an Energy Efficient and Flexible Automation Strategy and Control Optimization Approach of Supply Systems within a Thermally-Linked Factory’. In: *Procedia CIRP* 72 (2018), pp. 526–532.

[Pant19] N. Panten. *Deep Reinforcement Learning zur Betriebsoptimierung hybrider industrieller Energienetze: Dissertation*. Innovation Fertigungstechnik. Aachen: Shaker, 2019.

[Parv13] M. Parvania, M. Fotuhi-Firuzabad and M. Shahidehpour. ‘Optimal Demand Response Aggregation in Wholesale Electricity Markets’. In: *IEEE Transactions on Smart Grid* 4.4 (2013), pp. 1957–1965.

[Pete13] M. K. Petersen, K. Edlund, L. H. Hansen, J. Bendtsen and J. Stoustrup. ‘A taxonomy for modeling flexibility and a computationally efficient algorithm for dispatch in Smart Grids’. In: *2013 American Control Conference*. IEEE, 2013, pp. 1150–1156.

[Pfro16] J. Pfrommer, S. Gruner and F. Palm. ‘Hybrid OPC UA and DDS: Combining architectural styles for the industrial internet’. In: *2016 IEEE World Conference on Factory Communication Systems (WFCS)*. IEEE, 2016, pp. 1–7.

[Popp14] R. S. Popp and M. F. Zaeh. ‘Determination of the Technical Energy Flexibility of Production Systems’. In: *Advanced Materials Research* 1018 (2014), pp. 365–372.

[Prof19] S. Profanter, A. Tekat, K. Dorofeev, M. Rickert and A. Knoll. ‘OPC UA versus ROS, DDS, and MQTT: Performance Evaluation of Industry 4.0 Protocols’. In: *2019 IEEE International Conference on Industrial Technology (ICIT)*. IEEE, 2019, pp. 955–962.

[Raff21] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus and N. Dormann. ‘Stable-Baselines3: Reliable Reinforcement Learning Implementations’. In: *Journal of machine learning research* 22.268 (2021), pp. 1–8.

[Rako19] S. V. Raković and W. S. Levine. *Handbook of Model Predictive Control*. Cham: Springer International Publishing, 2019.

[Rein14] G. Reinhart, M. Graßl and S. Datzmann. 'Methode zur Bewertung der Energieflexibilität'. In: *wt Werkstattstechnik online* 104.H5 (2014), S. 313–319.

[Ritc22] H. Ritchie, M. Roser and P. Rosado. 'Energy'. In: *Our World in Data* (2022). URL: `https://ourworldindata.org/energy` (visited on 16/02/2023).

[Rögn21a] F.-H. Rögner and M. Pfeilschifter. *Markt- und Trendanalyse in der Industriellen Teilereinigung 2020*. Dresden: Fraunhofer-Institut für Organische Elektronik, Elektronenstrahl- und Plasmatechnik, 2021.

[Rögn21b] F.-H. Rögner and U. Vohrer. 'Die Erweiterung des Sinner'schen Kreises'. In: *JOT Journal für Oberflächentechnik* 61.S1 (2021), pp. 8–10.

[Rolo22] N. Roloff. *Entega expects the usage of 5-minute intervals in European electricity market*. Workshop in the project KI4ETA, 2022.

[Rösc19] M. Rösch, D. Bauer, L. Haupt, R. Keller, T. Bauernhansl, G. Fridgen, G. Reinhart and A. Sauer. 'Harnessing the Full Potential of Industrial Demand-Side Flexibility: An End-to-End Approach Connecting Machines with Markets through Service-Oriented IT Platforms'. In: *Applied Sciences* 9.18 (2019), p. 3796.

[Roth18] S. Roth, J. Rappold, S. Braunreuther and G. Reinhart. 'Modulare Simulationsumgebung zur Visualisierung und Analyse der Potenziale von industriellen Energieflexibilitäten'. In: *Neue Energie für unser bewegtes Europa*. Ed. by U. Bachhiesl. Graz: Verlag der Technischen Universität Graz, 2018.

[Roth19] S. Roth, M. Thimmel, J. Fischer, M. Schöpf, E. Unterberger, S. Braunreuther, H. U. Buhl and G. Reinhart. 'Simulation-based analysis of energy flexible factories in a regional energy supply system'. In: *Procedia Manufacturing* 33 (2019), pp. 75–82.

[Sche18] D. Schel, D. Bauer, F. G. Vasquez, F. Schulz and T. Bauernhansl. 'IT Platform for Energy Demand Synchronization Among Manufacturing Companies'. In: *21st CIRP Conference on Life Cycle Engineering* 72 (2018), pp. 826–831.

[Scho19] P. Schott, J. Sedlmeir, N. Strobel, T. Weber, G. Fridgen and E. Abele. 'A Generic Data Model for Describing Flexibility in Power Markets'. In: *Energies* 12.10 (2019).

[Schr18] P. Schraml. *Methode zur Reduktion maximaler elektrischer Lasten spanender Werkzeugmaschinen: Dissertation*. Schriftenreihe des PTW. Shaker, 2018.

[Schu18] J. Schulz, R. S.-H. Popp, V. M. Scharmer and M. F. Zaeh. 'An IoT Based Approach for Energy Flexible Control of Production Systems'. In: *Procedia CIRP* 69 (2018), pp. 650–655.

[Seit19] P. Seitz et al. 'IT-based Architecture for Power Market Oriented Optimization at Multiple Levels in Production Processes'. In: *21st CIRP Conference on Life Cycle Engineering* 81 (2019), pp. 618–623.

[Shil22] M. C. Shilenge and A. Telukdarie. 'Optimization of Operational and Information Technology Integration Towards Industry 4.0'. In: *2022 IEEE 31st International Symposium on Industrial Electronics (ISIE)*. IEEE, 2022, pp. 1076–1081.

[Sinn60] H. Sinner. *Über das Waschen mit Haushaltwaschmaschinen*. Hamburg: Haus + Heim-Verlag, 1960.

[Star18] R. Stark and T. Damerau. 'Digital Twin'. In: *CIRP Encyclopedia of Production Engineering*. Ed. by Chatti, Sami and Tolio, Tullio. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018, pp. 1–8.

[Step13] P. Stephan, K. Schaber, K. Stephan and F. Mayinger. *Thermodynamik: Grundlagen und technische Anwendungen Band 1: Einstoffsysteme*. 19., ergänzte Auflage. Vol. Band 1. Springer-Lehrbuch. Berlin and Heidelberg: Springer Vieweg, 2013.

[Stro20] N. Strobel, D. Fuhrländer-Völker, M. Weigold and E. Abele. 'Quantifying the Demand Response Potential of Inherent Energy Storages in Production Systems'. In: *Energies* 13.16 (2020).

[Stro21] N. Strobel. *Einsatz inhärenter Energiespeicher in Produktionssystemen zum elektrischen Lastmanagement*. Schriftenreihe des PTW: "Innovation Fertigungstechnik". Düren: Shaker Verlag, 2021.

[Tris20] A. Tristán, F. Heuberger and A. Sauer. 'A Methodology to Systematically Identify and Characterize Energy Flexibility Measures in Industrial Systems'. In: *Energies* 13.22 (2020).

[US D06] U.S. Department of Energy. 'Benefits of demand response in electricity markets and recommendations for achieving them: A Report to the United States Congress Pursuant to Section 1252 of the Energy Policy Act of 2005'. In: *US Dept. Energy, Washington, DC, USA, Tech. Rep* 5 (2006). URL: https://www.energy.gov/sites/prod/files/oeprod/DocumentsandMedia/DOE_Benefits_of_Demand_Response_in_Electricity_Markets_and_Recommendations_for_Achieving_Them_Report_to_Congress.pdf (visited on 14/04/2023).

[US E21] U.S. Energy Information Administration. *Monthly Energy Review April 2021: 7. Electricity*. Ed. by U.S. Energy Information Administration. Washington, DC, 2021. URL: https://www.eia.gov/totalenergy/data/monthly/ (visited on 09/05/2021).

[Vard15] J. S. Vardakas, N. Zorba and C. V. Verikoukis. 'A Survey on Demand Response Programs in Smart Grids: Pricing Methods and Optimization Algorithms'. In: *IEEE Communications Surveys & Tutorials* 17.1 (2015), pp. 152–178.

[VDI13] VDI and VDI-Gesellschaft Verfahrenstechnik und Chemieingenieurwesen, eds. *VDI-Wärmeatlas: Mit 320 Tabellen*. 11., bearb. und erw. Aufl. Berlin and Heidelberg: Springer Vieweg, 2013.

[VDI4661] VDI. 4661: *Energetic characteristics: Fundamentals – methodology*

[VDI5207-1] VDI. 5207-1:0. *Energieflexible Fabrik. Blatt 1: Grundlagen*

[VDI5207-2] VDI. 5207-2: *Energieflexible Fabrik. Blatt 2: Identifikation und technische Bewertung*

[VDMA34179] VDMA. 34179:2019-04. *34179, Messvorschrift zur Bestimmung des Energie- und Medienbedarfs von Werkzeugmaschinen in der Serienfertigung*

[Viel18] F. Viel, L. A. Silva, R. Q. Valderi Leithardt and C. A. Zeferino. ‘Internet of Things: Concepts, Architectures and Technologies’. In: *2018 13th IEEE International Conference on Industry Applications (INDUSCON)*. IEEE, 2018, pp. 909–916.

[Voge09] B. Vogel-Heuser, G. Kegel, K. Bender and K. Wucherer. ‘Global Information Architecture for Industrial Automation’. In: *Automatisierungstechnische Praxis (atp)* 51.1 (2009), pp. 108–115.

[Vogt22] M. Vogt, C. Buchholz, S. Thiede and C. Herrmann. ‘Energy efficiency of Heating, Ventilation and Air Conditioning systems in production environments through model-predictive control schemes: The case of battery production’. In: *Journal of Cleaner Production* 350 (2022).

[Walt22] J. Walther, B. Dietrich, B. Grosch, M. Lindner, D. Fuhrländer-Völker, N. Strobel and M. Weigold. ‘A Methodology for the Classification and Characterisation of Industrial Demand-Side Integration Measures’. In: *Energies* 15.3 (2022).

[Wang22] Y. Wang, N. Guo, G. Yan and J. Liu. ‘Design of Integrated Energy System Based on IEC 61499 and OPC UA’. In: *2022 41st Chinese Control Conference (CCC)*. IEEE, 2022, pp. 4270–4275.

[Webe18] T. Weber, N. Strobel, T. Kohne, J. Wolber and E. Abele. ‘Realistic modeling of a combined heat and power plant in the context of mixed integer linear programming’. In: *Energy Informatics* 1.S1 (2018).

[Xu19] X. Xu, Z. Wei, Q. Ji, C. Wang and G. Gao. ‘Global renewable energy development: Influencing factors, trend predictions and countermeasures’. In: *Resources Policy* 63 (2019).

[Zhu22] C. Zhu, B. Li, Z. Lv and X. Zhao. ‘Design and Implementation of the Substation Intelligent Auxiliary Control System Based on OPC UA’. In: *2022 5th International Conference on Renewable Energy and Power Engineering (REPE)*. IEEE, 2022, pp. 265–269.

Publications

2023 D. Fuhrländer-Völker, B. Grosch, and M. Weigold. ‘Modelling And Control Of Aqueous Parts Cleaning Machines For Demand Response’. In: *Proceedings of the Conference on Production Systems and Logistics: CPSL 2023* (2023).

D. Fuhrländer-Völker, J. Magin, and M. Weigold. ‘Automation Architecture for Harnessing the Demand Response Potential of Aqueous Parts Cleaning Machines’. In: *Production Engineering* (2023)

2022 D. Fuhrländer-Völker, M. Lindner, and M. Weigold. ‘Umsetzung und Implementierung am Beispiel von Speicher-Wandler-Systemen’. In: *Markt- und Stromsystem, Managementsysteme und Technologien energieflexibler Fabriken.* Ed. by A. Sauer, H. U. Buhl, A. Mitsos, and M. Weigold. Energieflexibilität in der deutschen Industrie. Stuttgart: Fraunhofer Verlag, 2022.

D. Fuhrländer-Völker, F. Borst, L. Theisinger, H. Ranzau, and M. Weigold. ‘Modular data model for energy-flexible cyber-physical production systems’. In: *Procedia CIRP* 107 (2022).

B. Grosch, D. Fuhrländer-Völker, J. Stock, and M. Weigold. ‘Cyber-physical production system for energy-flexible control of production machines’. In: *Procedia CIRP* 107 (2022).

B. Grosch, H. Ranzau, B. Dietrich, T. Kohne, D. Fuhrländer-Völker, J. Sossenheimer, M. Lindner, and M. Weigold. ‘A framework for researching energy optimization of factory operations’. In: *Energy Informatics* 5.S1 (2022).

J. Walther, B. Dietrich, B. Grosch, M. Lindner, D. Fuhrländer-Völker, N. Strobel, and M. Weigold. ‘A Methodology for the Classification and Characterisation of Industrial Demand-Side Integration Measures’. In: *Energies* 15.3 (2022).

2021 D. Fuhrländer-Völker, M. Lindner, and M. Weigold. ‘Design Method for Building Automation Control Programs to Enable the Energetic Optimization of Industrial Supply Systems’. In: *Procedia CIRP* 104 (2021).

2020 N. Strobel, D. Fuhrländer-Völker, M. Weigold, and E. Abele. ‘Quantifying the Demand Response Potential of Inherent Energy Storages in Production Systems’. In: *Energies* 13.16 (2020)

A. Weyand, S. Seyfried, B. Grosch, D. Fuhrländer-Völker, N. Panten, E. Abele. ‘Method for Transforming Energy and Production Systems Into Mobile Demonstrators’. In: *Procedia Manufacturing* 45 (2020)