

## Reihe 8

Mess-,  
Steuerungs- und  
Regelungstechnik

Nr. 1267

M.Sc. Haitham Elfahaam,  
Aachen

## A Runtime Adaptation Concept to reinforce Versatility in Industrial Automation

**ACPLT**  
**AACHENER**  
**PROZESSLEITTECHNIK**

Lehrstuhl für  
Prozessleittechnik  
der RWTH Aachen



# **A Runtime Adaptation Concept to reinforce Versatility in Industrial Automation**

Der Fakultät für Georessourcen und Materialtechnik der  
Rheinisch-Westfälischen Technischen Hochschule Aachen

zur Erlangung des akademischen Grades eines

**Doktors der Ingenieurwissenschaften**

vorgelegte Dissertation

von

**Haitham Elfahaam, M. Sc.**

aus Giza, Ägypten

Berichter: Univ.-Prof. Dr.-Ing. Ulrich Epple  
Univ.-Prof. Dr.-Ing. Birgit Vogel-Heuser



# Fortschritt-Berichte VDI

## Reihe 8

Mess-, Steuerungs-  
und Regelungstechnik

M.Sc. Haitham Elfahaam,  
Aachen

## Nr. 1267

A Runtime Adaptation  
Concept to reinforce  
Versatility in  
Industrial Automation



Lehrstuhl für  
Prozessleittechnik  
der RWTH Aachen

Elfahaam, Haitham

## **A Runtime Adaptation Concept to reinforce Versatility in Industrial Automation**

Fortschr.-Ber. VDI Reihe 08 Nr. 1267. Düsseldorf: VDI Verlag 2019.

130 Seiten, 63 Bilder, 10 Tabellen.

ISBN 978-3-18-526708-6 ISSN 0178-9546,

€ 52,00/VDI-Mitgliederpreis € 46,80.

**Für die Dokumentation:** Prozessleittechnik – Laufzeitadaption – Redeployment – Lastverteilung, Agentensysteme – Dezentrale Systeme – Industrie 4.0 – Wandelbarkeit – Optimierung – Stabilität

Unter dem Stichwort Wandelbarkeit wird die Fähigkeit verstanden, Industrieanlagen in die Lage zu versetzen, auf ungeplante Änderungen zu reagieren. Daher muss das Automatisierungssystem bereit sein auf Änderungen auf allen Ebenen in der Automatisierungspyramide reagieren zu können. In dieser Dissertation wird ein Konzept zur Laufzeitadaption vorgestellt. Das Konzept adressiert die Prozessleitebene und stellt ein Adaptionssystem vor, das die Softwarekomponenten im Netzwerk nach den verschiedenen Optimierungskriterien stabil verteilt und damit das Automatisierungssystem zur Wandelbarkeit befähigt.

### **Bibliographische Information der Deutschen Bibliothek**

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliographie; detaillierte bibliographische Daten sind im Internet unter [www.dnb.de](http://www.dnb.de) abrufbar.

### **Bibliographic information published by the Deutsche Bibliothek**

(German National Library)

The Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliographie (German National Bibliography); detailed bibliographic data is available via Internet at [www.dnb.de](http://www.dnb.de).

D82 (Diss. RWTH Aachen University, 2019)  
Tag der mündlichen Prüfung: 03. Juni 2019

© VDI Verlag GmbH · Düsseldorf 2019

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder vollständigen Wiedergabe (Fotokopie, Mikrokopie), der Speicherung in Datenverarbeitungsanlagen, im Internet und das der Übersetzung, vorbehalten.

Als Manuskript gedruckt. Printed in Germany.

ISSN 0178-9546

ISBN 978-3-18-526708-6

---

# Preface

This dissertation was written during my employment as an academic researcher at the Chair of Process Control Engineering of RWTH Aachen University under the supervision of Prof. Dr.-Ing. Ulrich Epple and Prof. Dr.-Ing. Birgit Vogel-Heuser. The research was supported by the German Federal Ministry of Education and Research in the framework of the project BaSys4.0 (Förderkennzeichen 01—S16022).

I thank Prof. Epple for his mentoring, guidance and above all his trust that offered me freedom to think innovatively, encounter great experiences and gain exposure. His great insight and deep understanding of process automation and interdisciplinary research offered me a great chance to learn and grow as a researcher, for that I am and will be forever grateful. I also thank Prof. Vogel-Heuser for co-supervising my dissertation, for her keen remarks and careful revision.

I thank the former and current team members of the Chair of Process Control Engineering namely (in alphabetical order) Mahyar Azarmipour, Torben Deppe, Dr. Lars Evertz, Julian Grothoff, Holger Jeromin, Dr. David Kampert, Lars Nothdurft, Florian Palm, Christian von Trotha, Constantin Wagner for the fruitful discussions and cooperation. I also thank Margarete Milesescu, Martina Uecker for their organization and efforts that helped me present this work.

I would like to express my deepest appreciation for the great work contributed by my student assistants Mariia Anapolska, Zolboo Erdenebayar and Michael Thies. Furthermore, I would like to express my sincere appreciation for the collaboration and fruitful discussions with project partners, Dr. Sten Grüner and Tarik Terzimehic.

I am immensely grateful to my parents, my brother, my grandmother and my family for their unconditional love, support and for being there through thick and thin.

Finally, I dedicate my dissertation to my late grandfather who helped and wished to see this work come to light. May your memory be eternal.

Haitham Elfaham  
Aachen, March 2019





---

# Contents

<b>Abstract</b>	<b>IX</b>
<b>Kurzfassung</b>	<b>XI</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.1.1 Applications and Scenarios . . . . .	2
1.1.2 Problem Definition . . . . .	3
1.2 Objective of this Work . . . . .	3
1.3 Structure of the Dissertation . . . . .	4
<b>2 State of the Art</b>	<b>6</b>
2.1 Dynamic RunTime Environments (RTE) . . . . .	6
2.2 Virtual Machines . . . . .	6
2.3 Container Technology . . . . .	7
2.3.1 Docker-Daemon . . . . .	8
2.3.2 Load Distribution . . . . .	8
2.3.3 Compatibility . . . . .	8
2.4 Migration and States Synchronization . . . . .	9
2.4.1 Service Migration in Automation . . . . .	9
2.4.2 Redundancy Migration . . . . .	10
2.5 Components in Automation . . . . .	10
2.5.1 Single Control Unit (SCU) . . . . .	11
2.5.2 Group Control Unit (GCU) . . . . .	11
2.5.3 Procedures . . . . .	11
2.5.4 Inner Structure of a Process Control Component . . . . .	12
2.5.5 Adaptation in Industrial Automation Systems . . . . .	13
2.6 Methodological Fundamentals - Graph Theory . . . . .	13
2.6.1 Bipartite Graph . . . . .	14
2.6.2 Adjacency Matrix . . . . .	14
2.6.3 Star Topology . . . . .	14
2.6.4 Hub and Spoke Topology . . . . .	14
2.6.5 Mesh Topology . . . . .	14
2.6.6 $K_n - K_n$ Topology . . . . .	14
2.6.7 Neighbor . . . . .	14
2.6.8 Valency . . . . .	14
2.7 Methodology - Load Distribution . . . . .	15
2.7.1 Load Distribution Algorithms . . . . .	15
2.7.2 Multi-Core Processing Analysis . . . . .	15

2.7.3	Consensus Networks	17
2.7.4	Semi-Definite Programming	17
2.7.5	MATLAB-YALMIP	18
2.7.6	Simulated Annealing	18
2.7.7	Optimal Distribution Solvers - Z3 SMT Solver	18
2.7.8	Deployment in Automotive Open System Architecture (AUTOSAR)	19
2.8	Agents Systems	19
2.9	Market-based Multi-Agent-System Approach	20
2.10	Agents Systems Hierarchy in Automation	20
2.11	Agent-Based Planning of Production Sequences	21
2.12	Recipes Definitions	21
2.13	Tools - Discovery	21
2.13.1	Bonjour Protocol	22
2.13.2	Mechanism of Operation	22
2.13.3	Reconfiguration of Real-Time Fieldbus	22
<b>3</b>	<b>Runtime Adaptation Concept</b>	<b>23</b>
3.1	Terminology and Definitions	23
3.2	Concept Overview	24
3.3	Process	26
3.3.1	Component redeployment	27
3.3.2	Container Redeployment	27
3.4	Controller	27
3.4.1	Optimization Criteria and Constraints	28
3.4.2	Boundary Conditions	31
3.4.3	Stability, Performance Analysis and Performance Enhancement	31
3.5	Actuator	31
3.5.1	Load Distribution Executor	31
3.6	Sensor	33
3.6.1	Resources & Component Manifestation	33
3.7	Disturbance	35
3.8	Architecture Overview	35
<b>4</b>	<b>Modeling Fundamentals</b>	<b>37</b>
4.1	The Load Balancing Model	37
4.1.1	Network Model	37
4.1.2	Load Model	37
4.1.3	Mathematical Model	38
<b>5</b>	<b>Methodology Investigation - Analytical Approach - Linear Model</b>	<b>40</b>
5.1	Modeling of the Adaption Algorithms	40
5.2	Model Characteristics	40
5.3	Model	40
5.3.1	Stability and Convergence Analysis	42
5.3.2	System Dynamic and Performance Analysis	42
5.4	Modeling of Multidimensional Loads	44
5.4.1	MD Problem Classification	44

5.4.2	MD Problem Modeling . . . . .	44
5.5	Performance Enhancement via Regression Models . . . . .	48
5.5.1	Optimizing the Transfer Coefficient . . . . .	48
5.5.2	Ring Hub and Spoke Networks . . . . .	49
5.5.3	Regression Model . . . . .	51
<b>6</b>	<b>Methodology Investigation - Empirical Approach - Non-Linear Model</b>	<b>55</b>
6.1	Model Characteristics . . . . .	55
6.2	Performance Assessment . . . . .	55
6.3	KPIs Preliminaries . . . . .	55
6.3.1	Network Topology . . . . .	56
6.3.2	Load Description . . . . .	56
6.3.3	Initial Conditions . . . . .	56
6.3.4	Node Capacity . . . . .	56
6.3.5	Probabilistic Algorithms . . . . .	56
6.3.6	Foreknowledge of Terminating Conditions . . . . .	56
6.4	KPIs . . . . .	57
6.4.1	Qualitative KPIs . . . . .	57
6.4.2	Quantitative KPIs . . . . .	57
6.4.3	Modular Benchmark . . . . .	60
6.4.4	Benchmark Testing . . . . .	60
<b>7</b>	<b>Use-Case - Implementation Approach</b>	<b>65</b>
7.1	Demonstrator - SMS-SEMG Cold Rolling Mill . . . . .	65
7.1.1	Devices . . . . .	69
7.1.2	Single and Group Function Units (SFU and GFU) . . . . .	70
7.2	Load Balancing . . . . .	71
<b>8</b>	<b>Implementation - Reality Approach</b>	<b>73</b>
8.1	Decentral Algorithm (Resources Perspective) . . . . .	73
8.1.1	Preliminaries . . . . .	74
8.1.2	Objective . . . . .	74
8.1.3	The BRAD Algorithm - Mechanism of Operation . . . . .	75
8.1.4	Simulation Assessment . . . . .	76
8.2	Agents Systems Approach (Components Perspective) . . . . .	83
<b>9</b>	<b>Scenario 1 - Decentralized Algorithm</b>	<b>85</b>
9.1	Realization . . . . .	85
9.1.1	Sync. State Machine BRAD . . . . .	86
9.1.2	Application Monitor . . . . .	86
9.1.3	Node to Node (N2N) Discovery . . . . .	86
9.1.4	Device Resources Monitoring . . . . .	88
9.1.5	Neighbor Informer . . . . .	90
9.1.6	Neighbor Data Bank . . . . .	90
9.1.7	TSE Optimizer - Sender End . . . . .	90
9.1.8	Request Sender . . . . .	92
9.1.9	Request Receiver . . . . .	92

9.1.10	TSE Optimizer - Receiver End . . . . .	92
9.1.11	Acceptance Notifier . . . . .	92
9.1.12	Send Initiator . . . . .	92
9.1.13	Redeployer . . . . .	92
9.2	Performance Assessment . . . . .	93
9.2.1	Setup . . . . .	93
9.2.2	Scenario . . . . .	95
<b>10</b>	<b>Scenario 2 - Agents System</b>	<b>97</b>
10.1	Realization . . . . .	97
10.1.1	Agent Load Balancing Algorithm . . . . .	100
10.2	Performance Assessment . . . . .	100
10.2.1	Setup . . . . .	100
10.2.2	Scenario . . . . .	101
<b>11</b>	<b>Conclusion and Outlook</b>	<b>105</b>
11.1	Outlook . . . . .	105
11.1.1	Algorithm Enhancement . . . . .	105
11.1.2	Synchronization . . . . .	106
11.1.3	Improvements for Load Model . . . . .	108
11.1.4	Improvements in the Decentralized Algorithm . . . . .	108
11.2	Improvements in the Infrastructure . . . . .	109
11.3	Further Utilizations of Agents Systems Approach . . . . .	109
<b>Bibliography</b>		<b>110</b>

---

# Abstract

In the process control engineering domain, various initiatives around the world (e.g. “Industry 4.0” in Germany) play a crucial role in directing the research and development. In the new generation of industrial automation, a new architecture is introduced where the communication hierarchy of the automation pyramid is dissolved in order to increase the flexibility of the production systems. One of the objectives of this architecture is to achieve “Adaptability” or in other words to enable industrial plants to react to unplanned changes. Furthermore, design principles like decentral decision making and interconnectiveness are widely promoted. In order to achieve the aforementioned goals, various new functionalities (e.g., Self-X functionalities and optimizations) and information (e.g., asset administration shell) are being introduced to the current production systems which did not exist in the conventional ones thus causing a dynamic overhead to the available resources (computation, communication, dynamic memory, etc.).

In the conventional systems, during the engineering phase, control logic and functionalities are designed and then deployed to the computation nodes in the automation network. In some cases, an optimized distribution profile for the loads are computed prior to the initial deployment and accordingly the load is distributed amongst the network endpoints. However, the dynamic aspect of the load variations introduced in the newly introduced automation paradigm is not taken into consideration.

System adaptation to the varying loads is required to readjust the loads and balance the resources consumption in the network. In industrial automation, safety aspects play a crucial role. Hence, a prerequisite for this framework is to not compromise the stability of the production system.

The objective of this dissertation is to establish a framework for a seamless integration of a deployment platform that can, through redeployment and adjustment of software components, balance the resources consumption overhead amongst the automation network participants, establish redundancy of the different components, improve the communication quality of service and adapt the system according to the rapid and dynamic changes imposed.

Thorough analysis and investigations for stability and the production system dynamics are conducted. The goal of these investigations is to ensure that the introduced framework does not affect the performance in any undesired manner, e.g., causing the loads to oscillate in the network or affecting the system performance with a non converging redeployment processes of the software components. Hence, additional to these investigations, a multi optimization criteria load balancing model is constructed to investigate the behavior or multidimensional optimizations. Moreover, performance enhancements analysis is conducted through investigating the automation networks and constructing regression models to compute the optimal parameters for load redeployment.

Furthermore, a prototype implementation to reinforce the presented concepts and validate the conducted investigations is realized. The prototype considers an aluminum cold rolling mill use-case and utilizes two different approaches namely decentral algorithms and

agent systems approaches to perform the load balancing from two different perspective namely resources and component perspectives respectively. In the former approach, the algorithm uses mathematical formulas (e.g., total square error) to compute the optimum load balancing profile from a decentral perspective and cooperates with other network participants to achieve the optimum load distribution profile on a global scale. On the other hand, in the latter approach, the components are considered as independent agents that wander the network. The information incubated within an agent is used (e.g. optimal routed path according to a given recipe) to anticipate the load distribution in the network and thus adjust the placement of the components (agents) accordingly.

The presented prototype implementation uses the runtime environment ACPLT/RTE and acts as extension library to provide the load balancing functionalities. The implementation uses the demonstrator from the SMS-group that simulates a cold rolling mill plant.

---

# Kurzfassung

Die Prozessleittechnik als ein Teil der Automatisierungstechnik erfährt durch Initiativen wie „Industrie 4.0“. In der Prozessleittechnik sind verschiedene Initiativen auf der ganzen Welt (z. B. „Industrie 4.0“ in Deutschland) derzeit in Forschung und Entwicklung richtungsweisend. Die Auflösung der Kommunikationshierarchie in der Automatisierungspyramide wird als zielführend erachtet, um den zunehmenden Anforderungen an Flexibilität in Produktionssystemen gerecht zu werden. Unter dem Stichwort „Wandelbarkeit“ wird die Fähigkeit verstanden, Industrieanlagen in die Lage zu versetzen, auf ungeplante Änderungen zu reagieren. Ansätze dazu sind Gestaltungsprinzipien wie dezentrale Entscheidungsfindung und vollständige Vernetzung. Um die zuvor genannten Ziele zu erreichen, werden verschiedene neue Funktionalitäten (z. B. Self-X-Funktionalitäten und -Optimierungen) und Informationen (z. B. die Verwaltungsschale) in die derzeitigen Produktionssysteme eingeführt, wodurch eine Dynamik und ein Overhead zu den verfügbaren Ressourcen (Berechnung, Kommunikation, dynamischer Speicher usw.) erzeugt wird.

In den konventionellen Systemen werden während der Engineering-Phase Steuerlogik und Funktionalitäten entworfen und dann an die Rechenknoten im Automatisierungsnetzwerk verteilt. In einigen Fällen wird ein optimiertes Verteilungsprofil für die Lasten vor der ersten Bereitstellung berechnet, und dementsprechend wird die Last auf die Netzwerkendpunkte verteilt. Der dynamische Aspekt der Lastschwankungen des neu eingeführten Automatisierungsparadigmas wird jedoch nicht berücksichtigt. Eine Systemanpassung an die unterschiedlichen Lasten ist erforderlich, um die Lasten neu verteilen und den Ressourcenverbrauch im Netzwerk auszugleichen. In der industriellen Automatisierung spielen Sicherheitsaspekte eine entscheidende Rolle. Eine Voraussetzung für diesen Rahmen ist daher, die Stabilität des Produktionssystems nicht zu beeinträchtigen.

Das Ziel dieser Dissertation ist die Schaffung eines Rahmens für die nahtlose Integration einer Implementierungsplattform, die durch die erneute Bereitstellung und Anpassung von Softwarekomponenten den Ressourcenverbrauch zwischen Teilnehmern eines Automatisierungsnetzwerks ausgleicht, die Redundanz der verschiedenen Komponenten herstellt und die Kommunikation verbessert in Bezug auf Servicequalität und sowie die Anpassung des Systems an die schnellen und kontinuierlichen Änderungen.

Gründliche Analysen und Untersuchungen zur Stabilität und zur Dynamik des Produktionssystems werden durchgeführt. Das Ziel dieser Untersuchungen besteht darin, sicherzustellen, dass das eingeführte Framework die Leistung nicht auf unerwünschte Weise beeinflusst, z. B. indem die Lasten im Netzwerk oszillieren oder die Systemleistung durch nicht konvergierende Umverteilungsprozesse der Softwarekomponenten beeinflusst wird. Zusätzlich zu diesen Untersuchungen wird daher ein Lastausgleichsmodell für mehrere Optimierungskriterien erstellt, um das Verhalten oder mehrdimensionale Optimierungen zu untersuchen. Darüber hinaus wird die Analyse der Leistungsverbesserungen durchgeführt, indem die Automatisierungsnetzwerke untersucht und Regressionsmodelle erstellt werden, um die optimalen Parameter für die Lastumschichtung zu berechnen. Darüber hinaus wird eine Prototyp-Implementierung realisiert, um die vorgestellten Konzepte zu verstärken und

die durchgeführten Untersuchungen zu validieren. Der Prototyp betrachtet den Anwendungsfall eines Aluminium-Kaltwalzwerks und verwendet zwei unterschiedliche Ansätze, nämlich dezentrale Algorithmen und Ansätze von Agentensystemen, um den Lastausgleich unter zwei verschiedenen Gesichtspunkten durchzuführen, nämlich Ressourcen- und Komponentenperspektiven.

Die vorgestellte Prototypimplementierung verwendet die Laufzeitumgebung ACPLT/RTE und fungiert als Erweiterungsbibliothek, um die Lastenausgleichsfunktionen bereitzustellen. Die Implementierung verwendet den Demonstrator der SMS-Gruppe, der eine Aluminium-Kaltwalzanlage simuliert.