

Dennis Bauer

»Lernende ereignisbasierte Optimierung
der Produktionssteuerung für die komplexe
Werkstattfertigung«



Fraunhofer
IPA



Universität Stuttgart

Dennis Bauer

»Lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe
Werkstattfertigung«

Herausgeber

Univ.-Prof. Dr.-Ing. Thomas Bauernhansl^{1,2}

Univ.-Prof. Dr.-Ing. Dipl.-Kfm. Alexander Sauer^{1,3}

Univ.-Prof. Dr.-Ing. Kai Peter Birke⁴

Univ.-Prof. Dr.-Ing. Marco Huber^{1,2}

¹Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA, Stuttgart

²Institut für Industrielle Fertigung und Fabrikbetrieb (IFF) der Universität Stuttgart

³Institut für Energieeffizienz in der Produktion (EEP) der Universität Stuttgart

⁴Institut für Photovoltaik (*ipv*) der Universität Stuttgart

Kontaktadresse:

Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA
Nobelstr. 12
70569 Stuttgart
Telefon 0711 970-1101
info@ipa.fraunhofer.de
www.ipa.fraunhofer.de

Bibliographische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliographie; detaillierte bibliografische Daten sind im Internet über <http://dnb.de> abrufbar.

Zugl.: Stuttgart, Univ., Diss., 2022

D 93

2022

Druck und Weiterverarbeitung:

Fraunhofer Verlag, Mediendienstleistungen, Stuttgart, 2022
Für den Druck des Buches wurde chlor- und säurefreies Papier verwendet.

Dieses Werk ist einschließlich aller seiner Teile urheberrechtlich geschützt. Alle Rechte, insbesondere die der Übersetzung, des Nachdrucks, der Wiedergabe, sind vorbehalten.

Lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung

**Von der Fakultät Konstruktions-, Produktions- und Fahrzeugtechnik
der Universität Stuttgart
zur Erlangung der Würde eines Doktor-Ingenieurs (Dr.-Ing.)
genehmigte Abhandlung**

Vorgelegt von

Dennis Bauer

aus Heilbronn

Hauptberichter: Univ.-Prof. Dr.-Ing. Thomas Bauernhansl

Mitberichter: Univ.-Prof. em. Dr.-Ing. Gunther Reinhart

Univ.-Prof. Dr.-Ing. Dipl.-Kfm. Alexander Sauer

Tag der mündlichen Prüfung: 02.06.2022

Institut für Industrielle Fertigung und Fabrikbetrieb der Universität Stuttgart

2022

Vorwort

Die vorliegende Dissertation entstand während meiner Tätigkeit am Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA.

Mein besonderer Dank gilt Herrn Univ.-Prof. Dr.-Ing. Thomas Bauernhansl, dem Leiter des Instituts für Industrielle Fertigung und Fabrikbetrieb (IFF) an der Universität Stuttgart und des Fraunhofer-Instituts für Produktionstechnik und Automatisierung IPA, für das Vertrauen und die Unterstützung meines Promotionsvorhabens, für die wissenschaftliche Begleitung und die persönlichen Entwicklungsmöglichkeiten, die er mir eröffnet hat. Für die Übernahme des Mitberichts bedanke ich mich herzlich bei Herrn Univ.-Prof. em. Dr.-Ing. Gunther Reinhart, dem ehemaligen Leiter des Instituts für Werkzeugmaschinen und Betriebswissenschaften (iwb) an der Technischen Universität München und des Fraunhofer-Instituts für Gießerei-, Composite- und Verarbeitungstechnik IGCV. Bei Herrn Univ.-Prof. Dr.-Ing. Dipl.-Kfm. Alexander Sauer, dem Leiter des Instituts für Energieeffizienz in der Produktion (EEP) an der Universität Stuttgart und des Fraunhofer-Instituts für Produktionstechnik und Automatisierung IPA, bedanke ich mich herzlich für die Übernahme des zweiten Mitberichts sowie die wertvollen Diskussionen und die wissenschaftliche Begleitung meiner Arbeit. Besonders bedanke ich mich auch für die vertrauensvolle Zusammenarbeit sowie die Möglichkeit, die Gruppe Energieflexible Produktion & Energiedatenanalyse am Institut aufzubauen und mit dem Kopernikus-Projekt SynErgie eines der bis dahin größten deutschen Forschungsprojekte zur erfolgreichen Gestaltung der Energiewende in der Industrie zu koordinieren. Herrn Univ.-Prof. Dr.-Ing. Dr. h.c. mult. Alexander Verl, Leiter des Instituts für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen (ISW) an der Universität Stuttgart, danke ich für die Übernahme des Vorsitzes der Prüfungskommission.

Bei meinen Kolleginnen und Kollegen am Institut, insbesondere aus meiner Abteilung und Gruppe, bedanke ich mich für die spannende, abwechslungsreiche und schöne gemeinsame Zeit. Mein Dank gilt insbesondere Herrn Dr.-Ing. Sebastian Weckmann, Herrn Can Kaymakci und Herrn Daniel Umgelter für die konstruktiven Diskussionen und das wertvolle Feedback zu meiner Arbeit sowie Frau Lena Hitzberger für den sehr hilfreichen Style-Guide für die

Grafiken. Mein Dank gilt darüber hinaus allen Studierenden, die mich bei meiner Arbeit unterstützt haben, insbesondere Herrn Andreas Schlereth und Herrn Daniel Umgelter, die anschließend Kollegen am Institut wurden, sowie Frau Lina Baier und Herrn Manuel von Hochmeister. Ebenso bedanke ich mich bei Herrn Univ.-Prof. Dr.-Ing. Marco Huber, Herrn Raphael Lamprecht und Herrn Martin Tschöpe für die gewinnbringenden Reinforcement-Learning-Diskussionen. Bei den Mitgliedern des Forschungsprojekts Sem140, insbesondere Herrn Dr.-Ing. Thomas Ponsignon, bedanke ich mich für die wertvollen Workshops und Diskussionen. Meiner Kollegin Dr. phil. Birgit Spaeth, meinen Freunden Dr. Julian Klepp, Dr. Sarah Legendre und Dr. Benjamin Ziegler sowie meiner Frau Dr. Sonja Bauer danke ich herzlich für die sorgfältige Durchsicht meiner Arbeit.

Der größte und wichtigste Dank gilt meinen Freunden und meiner Familie, die auf diesem Weg immer an mich geglaubt haben. Insbesondere danke ich meinen Eltern Christine und Steffen sowie meinem Bruder Philipp Hardy für die mentale Unterstützung und den Rückhalt bei der Erstellung meiner Arbeit. Von ganzem Herzen danke ich meiner Frau Sonja für die Unterstützung, die kontinuierliche Motivation und das Verständnis für die vielen entgangenen gemeinsamen Stunden. Ohne dich wäre diese Arbeit so nicht möglich gewesen! Ich freue mich sehr darauf, all diese Stunden zukünftig wieder gemeinsam mit dir zu verbringen.

Stuttgart, im Juni 2022

Dennis Bauer

Kurzfassung

Für produzierende Unternehmen steht heute der Kundennutzen und damit auch die Liefertermineinhaltung im Fokus der unternehmerischen Aktivitäten. Parallel steigt die Komplexität des Umfelds und der Märkte sowie innerhalb des Unternehmens. Deshalb ist bei der Leistungserstellung in Wertschöpfungsnetzwerken ein systematischer Umgang mit unvorhergesehenen Ereignissen im Fokus. Eine besondere Herausforderung ist dies für komplexe Werkstattfertigungen mit ihrer immanenten statischen und dynamischen Komplexität.

Die vorliegende Arbeit adressiert die Forschungsfrage, wie eine lernende, auf Ereignissen aus dem Wertschöpfungsnetzwerk basierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung gestaltet werden kann. Die in der vorliegenden Arbeit entwickelte Lösung optimiert die Liefertermineinhaltung bei Ereignissen aus dem Wertschöpfungsnetzwerk durch Korrekturmaßnahmen auf der Ebene der Produktionssteuerung in komplexen Werkstattfertigungen. Hierfür ist ein lernfähiges Entscheidungsmodell notwendig, das eine kontinuierliche Anpassung über die Zeit ermöglicht. Bislang bei Ereignissen aus dem Wertschöpfungsnetzwerk übliche manuelle Eingriffe in die Produktionssteuerung werden dadurch vermieden und die Resilienz des Produktionssystems gesteigert. Ergebnisse der vorliegenden Arbeit sind die Artefakte der lernenden ereignisbasierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung in Form des zugehörigen Regelkreises, der Methode und der Architektur sowie der Implementierung als Softwareartefakt.

Die Halbleiterfertigung ist die industriell wichtigste Anwendung der komplexen Werkstattfertigung. Hier zeigt sich, dass durch die lernende ereignisbasierte Optimierung der Produktionssteuerung die Liefertermineinhaltung bei Ereignissen aus dem Wertschöpfungsnetzwerk signifikant verbessert werden kann. Eine hohe Liefertermineinhaltung sorgt dabei für eine höhere Kundenzufriedenheit, der systematische Umgang mit Ereignissen sorgt für Resilienz. Beides ermöglicht perspektivisch einen Abbau von Sicherheitsbeständen im Wertschöpfungsnetzwerk.

Abstract

For manufacturing companies, the focus of their business activities today is on customer value and thus also on meeting delivery dates. At the same time, the complexity of the environment and the markets as well as within the company is increasing. For this reason, the focus is on systematic handling of events within the value creation process in supply networks. This is a particular challenge for complex job shop manufacturing with its immanent static and dynamic complexity.

This thesis addresses the research question of how a learning, on events from the supply network based optimization of manufacturing control for a complex job shop can be designed, optimizing the adherence to delivery dates in case of events from the supply network by corrective measures on the level of manufacturing control. This requires an adaptive decision model that enables continuous adaptation over time. Manual interventions in manufacturing control, which were previously common in the case of events from the supply network, are thus avoided and the resilience of the manufacturing system is increased. Results of this thesis are the artifacts of learning event-based optimization of manufacturing control for complex job shop manufacturing in the form of the associated control loop, method and architecture, and implementation as a software artifact.

Semiconductor manufacturing is the most important application of complex job shops in industry. Here it is shown that the learning event-based optimization of manufacturing control can significantly improve the adherence to delivery dates in the case of events from the supply network. A high level of adherence to delivery dates ensures higher customer satisfaction, while the systematic handling of events ensures resilience. In the long term, both of these factors will make it possible to reduce safety stocks in the supply network.

Inhaltsverzeichnis

Abkürzungsverzeichnis	X
Symbolverzeichnis	XI
Abbildungsverzeichnis	XVI
Tabellenverzeichnis	XX
1 Einleitung	1
1.1 Ausgangssituation	1
1.2 Problemstellung	4
1.3 Zielsetzung und Forschungsfragen	8
1.4 Wissenschaftstheoretische Positionierung und Forschungskonzeption	9
1.5 Aufbau der Arbeit	13
2 Begriffe und Grundlagen	15
2.1 Komplexe Werkstattfertigung und Wertschöpfungsnetzwerk	15
2.1.1 Die Charakteristik der komplexen Werkstattfertigung	16
2.1.2 Die komplexe Werkstattfertigung als Teil des Wertschöpfungsnetzwerks	23
2.1.3 Ereignisse als Komplexitätstreiber	26
2.1.4 IT-Unterstützung für das Management von Wertschöpfungsnetzwerken	29
2.2 Produktionsplanung und -steuerung	31
2.2.1 Die Rolle der Produktionsplanung und -steuerung	32
2.2.2 Ziele und Messgrößen der Produktionsplanung und -steuerung	35
2.2.3 Produktionsplanung und -steuerung in der komplexen Werkstattfertigung	39
2.2.4 IT-Unterstützung in der Produktionsplanung und -steuerung	42

2.3	Lernende Systeme	44
2.3.1	Die Charakteristik lernender Systeme	45
2.3.2	Maschinelles Lernen zum Aufbau lernender Systeme	46
2.3.3	Regelkreise als Handlungsmöglichkeit lernender Systeme	53
2.3.4	Regelkreise in der Produktionsplanung und -steuerung	55
2.4	Zwischenfazit und Handlungsbedarf	57
3	Stand der Wissenschaft und Technik	61
3.1	Methodisches Vorgehen	61
3.2	Reaktion auf Ereignisse aus Wertschöpfungsnetzwerken	62
3.2.1	Fokus auf die Ereignisverarbeitung	63
3.2.2	Fokus auf die Integration der Reaktion in die Produktionsplanung und -steuerung	66
3.3	Ereignisverarbeitung in der Produktionssteuerung	68
3.3.1	Fokus auf die Architektur zur Ereignisverarbeitung	68
3.3.2	Fokus auf die Anpassung des Ablaufplans	70
3.4	Nutzung von bestärkendem Lernen in der Produktionssteuerung der Werk- stattfertigung	72
3.5	Zwischenfazit	77
4	Anforderungen	79
4.1	Anforderungsdefinition	79
4.1.1	Anforderungen aus dem Anwendungszusammenhang	80
4.1.2	Anforderungen an das Entscheidungsmodell	81
4.1.3	Anforderungen an die Systemarchitektur	83
4.2	Einordnung bestehender Ansätze	83
4.2.1	Reaktion auf Ereignisse aus Wertschöpfungsnetzwerken	84
4.2.2	Ereignisverarbeitung in der Produktionssteuerung	87
4.2.3	Nutzung von bestärkendem Lernen in der Produktionssteuerung der Werkstattfertigung	90
4.3	Zwischenfazit und Handlungsbedarf	92

5	Konzeption der lernenden ereignisbasierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung	93
5.1	Methodisches Vorgehen	93
5.2	Definition des Systemkonzepts	94
5.3	Detailbeschreibung des Systemkonzepts	97
5.3.1	Regelkreis	97
5.3.2	Methode	101
5.3.2.1	Phase I: Design	103
5.3.2.2	Phase II: Training	121
5.3.2.3	Phase III: Applikation	133
5.3.3	Architektur	140
5.4	Zwischenfazit	143
6	Applikation der lernenden ereignisbasierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung	147
6.1	Charakteristik der Anwendungsumgebung der Halbleiterfertigung	147
6.1.1	Das Wertschöpfungsnetzwerk der Halbleiterindustrie	148
6.1.2	Die Fertigung in der Halbleiterindustrie	149
6.2	Implementierung des Systemkonzepts in der Halbleiterfertigung	152
6.2.1	Simulationsmodell des Produktionssystems	152
6.2.2	Regler des Adaptiven Regelkreises für ein integriertes Wertschöpfungsnetzwerk	155
6.2.3	Auswahl eines Algorithmus	157
6.3	Anwendung und Validierung in der Halbleiterfertigung	159
6.3.1	Design des Reglers für die Anwendungsumgebung	159
6.3.2	Training des Regler-Entscheidungsmodells	161
6.3.2.1	Hyperparameter-Optimierung	162
6.3.2.2	Parametervariation der Agentenparameter	165
6.3.2.3	Trainingsbewertung	167
6.3.3	Applikation des Reglers	168
6.3.3.1	Liefertermineinhaltung	170

6.3.3.2	Grenzbetrachtung	171
6.3.3.3	Ergebnisbewertung	175
6.4	Zwischenfazit	176
7	Reflexion	179
7.1	Bewertung der Erfüllung der Anforderungen	179
7.1.1	Anforderungen aus dem Anwendungszusammenhang	179
7.1.2	Anforderungen an das Entscheidungsmodell	180
7.1.3	Anforderungen an die Systemarchitektur	182
7.2	Beantwortung der Forschungsfrage	182
8	Zusammenfassung und Ausblick	185
8.1	Zusammenfassung	185
8.2	Ausblick	187
	Literatur	190
A	Anhang	233
A.1	Klassifikation von Ereignissen	233
A.2	Daten für Supply Chain Management und Produktionsplanung und -steuerung	234
A.3	IT-Unterstützung in der Produktionsplanung und -steuerung	235
A.4	Klassifikation von Ansätzen der Fertigungsregelung	238
A.5	Kurzfristige, operative Maßnahmen der Produktionssteuerung	240
A.6	Modellparameter	243
A.7	Auswahl eines Algorithmus	247
A.8	Hyperparameter-Optimierung	253
A.9	Parametervariation der Agentenparameter	261
A.10	Grenzbetrachtung	262
A.11	Studentische Arbeiten	271
A.12	Thematisch bezogene Veröffentlichungen des Autors	272

Abkürzungsverzeichnis

A3C	Asynchronous Advanced Actor Critic
Ape-X	Distributed Prioritized Experience Replay
API	Application Programming Interface
APPO	Asynchronous Proximal Policy Optimization
APS	Advanced Planning and Scheduling System
ARIW	Adaptiver Regelkreis für ein integriertes Wertschöpfungsnetzwerk
Back-End	Back End of Line
BDE	Betriebsdatenerfassung
CEP	Complex Event Processing
CoJoSim	Complex Job Shop Simulation
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture
DES	Discrete Event Simulation
DQN	Deep Q-Network
EDD	Earliest Due Date
ERP	Enterprise Resource Planning
FIFO	First In First Out
FOUP	Front Opening Unified Pod
Front-End	Front-End-of-Line
GAE	Generalized Advantage Estimator
GPU	Graphics Processing Unit

IMPALA	Importance Weighted Actor-Learner Architectures
IT	Informationstechnologie
KI	Künstliche Intelligenz
KNN	Künstliches Neuronales Netz
LSTM	Long Short-Term Memory
MDE	Maschinendatenerfassung
MES	Manufacturing Execution System
MIMAC	Measurement and Improvement of Manufacturing Capacity
ML	Maschinelles Lernen
PPO	Proximal Policy Optimization
PPS	Produktionsplanung und -steuerung
RL	Reinforcement Learning
SARSA	State-Action-Reward-State-Action
SCEM	Supply Chain Event Management
SCM	Supply Chain Management
SPT	Shortest Processing Time
TRPO	Trust Region Policy Optimization

Symbolverzeichnis

Griechische Symbole

Symbol	Beschreibung
α	Lernrate
α_0	Anfängliche Lernrate zum Zeitpunkt t_0
α_{Final}	Finale Lernrate zum Zeitpunkt $t_0 + n_\alpha$
β	Entropiefaktor
γ	Diskontierungsfaktor
Δ	Differenz
θ	Modellgewichte
θ_π	Modellgewichte der Strategie
θ_v	Modellgewichte der Wertfunktion
λ	Verhältnis zwischen Varianz und Bias
μ_P	Erwartungswert einer Poisson-Verteilung
π	Strategie eines Agenten
π^*	Optimale Strategie eines Agenten
τ	Zustands-Aktions-Belohnungs-Trajektorie

Lateinische Symbole

Symbol	Beschreibung
A	Aktionsraum
A_t	Aktion zu einem Zeitpunkt t
a	Aktion aus dem Aktionsraum unabhängig von einem Zeitpunkt
a_0	Aktion „keine Handlung“
$a_{\uparrow}$	Aktion zur Prioritätserhöhung eines Loses
$a_{\downarrow}$	Aktion zur Prioritätsverringering eines Loses
$A^{\pi}(S_t, A_t)$	Advantage-Funktion
$A^*(S_t, A_t)$	Optimale Advantage-Funktion
$A_t^{GAE(\gamma, \lambda)}$	Schätzer für die Advantage-Funktion
B_m	Mittlerer Bestand einer Fertigung
D_{KNN}	Struktur eines Künstlichen Neuronalen Netzes
E	Ereignisraum
E_t	Ereignis zu einem Zeitpunkt t
F	Fertigstellungsgrad
FF	Flussfaktor
$FF_{Freigabe}$	Freigabe-Flussfaktor (Flussfaktor zum Zeitpunkt der Auftragsfreigabe)
FF_{Ist}	Ist-Flussfaktor
FF_{Plan}	Plan-Flussfaktor
FF_{Steuer}	Steuer-Flussfaktor
G_t	Rückgabe
L	Losnummer

Symbol	Beschreibung
L_m	Mittlere Leistung einer Fertigung
LTA	Lieferterminabweichung eines Auftrags
LTE	Liefertermineinhaltung
M_t	Modellparameter zu einem Zeitpunkt t
$M_{Prod,t}$	Modellparameter für den Produktivbetrieb zu einem Zeitpunkt t
MG	Maschinengruppe
n_A	Anzahl der Aufträge
n_α	Anzahl der Schritte zum Absenken der Lernrate α
n_{Akt}	Anzahl der Aktoren
n_L	Anzahl der Lose
$n_{L,A}$	Anzahl der abgeschlossenen Lose
$n_{L,B}$	Anzahl der Lose im Bestand
n_M	Anzahl der Maschinen
n_{MG}	Anzahl der Maschinengruppen
n_P	Anzahl der Produkttypen
n_{Stk}	Stückzahl in einem Los
n_τ	Anzahl der Schritte in einer Zustands-Aktions-Belohnungs-Trajektorie
P	Produkttyp
$P(S_{t+1} S_t, A_t)$	Übergangswahrscheinlichkeit eines Markov-Entscheidungsprozesses
$Q^\pi(S_t, A_t)$	Qualitätsfunktion
$Q^*(S_t, A_t)$	Optimale Qualitätsfunktion
r	Rückführgröße

Symbol	Beschreibung
$R(S_t, A_t, S_{t+1})$	Belohnungsfunktion eines Markov-Entscheidungsprozesses
$R_A(S_t, A_t, S_{t+1})$	Belohnungsfunktion für abgeschlossene Lose
$R_B(S_t, A_t, S_{t+1})$	Belohnungsfunktion für Lose im Bestand eines Produktionssystems
R_t	Belohnung zu einem Zeitpunkt t
S	Zustandsraum
S_{Los}	Zustandsraum eines Loses
S_t	Zustand zu einem Zeitpunkt t
t	Zeitpunkt
t_0	Start-Zeitpunkt
T	End-Zeitpunkt einer Episode
T_{Epoche}	End-Zeitpunkt einer Epoche
t_g	Zeitpunkt eines globalen Zählers
t_e	Zeitintervall Poisson-verteilter Ereignisse
$T(S_t, A_t, S_{t+1})$	Übergangsfunktion eines Markov-Entscheidungsprozesses
TAB	Bearbeitungsbeginn eines Auftrags
TAE	Bearbeitungsende eines Auftrags
TL_{Ist}	Ist-Liefertermin eines Auftrags
TL_{Plan}	Plan-Liefertermin eines Auftrags
$V^\pi(S_t)$	Wertfunktion (auch Zustands-Wert-Funktion)
$V^*(S_t)$	Optimale Wertfunktion (auch optimale Zustands-Wert-Funktion)
w	Führungsgröße
x	Regelgröße
y	Stellgröße

Symbol	Beschreibung
z	Störgröße
ZDA	Durchlaufzeit eines Auftrags
ZDA_m	Mittlere Durchlaufzeit eines Auftrags
ZDF	Durchführungszeit eines Arbeitsvorgangs
ZDL	Durchlaufzeit eines Arbeitsvorgang
ZUE	Übergangszeit

Abbildungsverzeichnis

1.1	Umfeld und Herausforderungen für produzierende Unternehmen	1
1.2	Fehlerkreis der Produktionssteuerung	5
1.3	Wissenschaftssystematik	10
1.4	Information Systems Research Framework	12
1.5	Aufbau der Arbeit	14
2.1	Aufbau und Elemente der Produktion	16
2.2	Unterscheidung der Fertigungsprinzipien	18
2.3	Kombinationen von Fertigungsarten und Fertigungsprinzipien	19
2.4	Charakteristika komplexer Systeme	20
2.5	Planungs- und Entscheidungsaufgaben in Wertschöpfungsnetzwerken	25
2.6	IT-Systeme des SCEM	30
2.7	Aufgaben nach dem Aachener PPS-Modell	33
2.8	Komponenten der Durchlaufzeit	37
2.9	Durchlaufdiagramm eines Produktionssystems	39
2.10	Hierarchie der PPS-Aufgaben in der komplexen Werkstattfertigung	41
2.11	IT-Systeme der PPS in der komplexen Werkstattfertigung	42
2.12	Referenzarchitektur für eine ereignisbasierte PPS	44
2.13	Theorien des Lernens in der Fertigung	46
2.14	Interaktion eines Agenten mit seiner Umgebung	50
2.15	Algorithmen-Familien für bestärkendes Lernen	52
2.16	Regelkreis mit Störgröße	54
2.17	Grundstruktur der PPS als Regelkreis	56
2.18	Klassifikation von Ansätzen der Fertigungsregelung	57
3.1	Methodisches Vorgehen zur Analyse des Stands der Wissenschaft und Technik	62
3.2	Charakterisierung des Entscheidungsprozesses	63

5.1	Gestaltungszyklus für die lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung	94
5.2	Artefakte der lernenden ereignisbasierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung	95
5.3	Der adaptive Regelkreis für ein integriertes Wertschöpfungsnetzwerk	98
5.4	Der adaptive Regelkreis für ein integriertes Wertschöpfungsnetzwerk eingebettet in die PPS	99
5.5	Methode zur Gestaltung des adaptiven Regelkreises für ein integriertes Wertschöpfungsnetzwerk	102
5.6	Phase I: Design der Methode zur Gestaltung des adaptiven Regelkreises für ein integriertes Wertschöpfungsnetzwerk	104
5.7	Zusammensetzung des Zustandsraums des Agenten	108
5.8	Zusammensetzung des Aktionsraums des Agenten	112
5.9	Phase II: Training der Methode zur Gestaltung des adaptiven Regelkreises für ein integriertes Wertschöpfungsnetzwerk	121
5.10	Interaktion eines Agenten mit einem Simulationsmodell über die Abstraktionsschicht der Umgebung	123
5.11	Prüfung der Abbruchbedingungen von Episoden und der Epoche	129
5.12	Exemplarischer Verlauf einer Belohnung über die Schritte einer Epoche	130
5.13	Exemplarisches Durchlaufdiagramm eines Produktionssystems	131
5.14	Exemplarische Liefertermineinhaltung eines Produktionssystems	132
5.15	Prüfung der Modellevaluation im Training	132
5.16	Phase III: Applikation der Methode zur Gestaltung des adaptiven Regelkreises für ein integriertes Wertschöpfungsnetzwerk	133
5.17	Interaktion eines Agenten mit einem Produktionssystem über die Abstraktionsschicht der Umgebung	134
5.18	Prüfung der Modellevaluation im Betrieb	140
5.19	Architektur zur Einbettung des adaptiven Regelkreises für ein integriertes Wertschöpfungsnetzwerk in die Produktions- IT-Landschaft	141
5.20	Architektur des adaptiven Regelkreises für ein integriertes Wertschöpfungsnetzwerk	143

6.1	Stufen der Halbleiterherstellung	149
6.2	Prozessschritte der Front-End-Fertigung in der Halbleiterindustrie	150
6.3	Schritte zur Modellierung und Simulation	153
6.4	Struktur des Konzeptmodells für das untersuchte Produktionssystem	154
6.5	Implementierung der Umgebung und des Agenten	156
6.6	Durchlaufdiagramm des Produktionssystems der Anwendungsumgebung	161
6.7	Einfluss der Variation der Lernrate	164
6.8	Einfluss der Variation des Entropiefaktors	165
6.9	Einfluss der Variation der Flussfaktor-Anpassung	167
6.10	Vergleich der Basislinie und der Trainingslinie	169
6.11	Liefertermineinhaltung mit Ereignissen 1-28800-0,5	171
6.12	Einfluss des Zeitintervalls von Ereignissen auf die Liefertermineinhaltung	172
6.13	Einfluss der Lieferterminänderung von Ereignissen auf die Liefertermineinhaltung	173
6.14	Liefertermineinhaltung mit Ereignissen 1-14400-0,75	174
A.1	Verlauf der Belohnung für die Algorithmen DQN und DQN Rainbow	247
A.2	Verlauf der Belohnung für die Algorithmen Ape-X und Ape-X Rainbow	248
A.3	Verlauf der Belohnung für die Algorithmen A3C und IMPALA	249
A.4	Verlauf der Belohnung für die Algorithmen PPO und APPO	249
A.5	Liefertermineinhaltung mit den Algorithmen PPO, APPO, A3C und IMPALA	252
A.6	Einfluss der Variation der Lernrate, höhere Werte als in der Basislinie	253
A.7	Einfluss der Variation der Lernrate, niedrigere Werte als in der Basislinie und absinkende Werte	254
A.8	Einfluss der Variation des Entropiefaktors	255
A.9	Einfluss der Variation des Diskontierungsfaktors	256
A.10	Einfluss der Variation der Struktur des KNN, Anzahl der Neuronen	257
A.11	Einfluss der Variation der Struktur des KNN, Anzahl der versteckten Schichten	258
A.12	Einfluss der Variation der Struktur des KNN, Ergänzung eines LSTM	260
A.13	Einfluss der Variation der Flussfaktor-Anpassung	261
A.14	Einfluss der Variation der Schrittweite	262
A.15	Liefertermineinhaltung mit Ereignissen 1-86400-0,5	263

A.16 Liefertermineinhaltung mit Ereignissen 1-14400-0,5	264
A.17 Liefertermineinhaltung mit Ereignissen 1-3600-0,5	265
A.18 Liefertermineinhaltung mit Ereignissen 1-900-0,5	266
A.19 Liefertermineinhaltung mit Ereignissen 1-28800-0,25	268
A.20 Liefertermineinhaltung mit Ereignissen 1-28800-0,75	268
A.21 Liefertermineinhaltung mit Ereignissen 1-28800-1,0	269
A.22 Liefertermineinhaltung mit Ereignissen 1-28800-1,25	269
A.23 Trainingslinie trainiert mit Ereignissen 1-14400-0,75	270

Tabellenverzeichnis

2.1	Zuordnung der Aufgaben zwischen komplexer Werkstattfertigung und allgemeiner PPS	42
2.2	Gliederung verschiedener Ansätze des ML	48
4.1	Anforderungen an eine lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung	80
4.2	Bewertung der Ansätze zur Reaktion auf Ereignisse aus Wertschöpfungsnetzwerken mit Fokus auf die Ereignisverarbeitung	85
4.3	Bewertung der Ansätze zur Reaktion auf Ereignisse aus Wertschöpfungsnetzwerken mit Fokus auf die Integration der Reaktion in die PPS	86
4.4	Bewertung der Ansätze zur Ereignisverarbeitung in der Produktionssteuerung mit Fokus auf die Architektur zur Ereignisverarbeitung	88
4.5	Bewertung der Ansätze zur Ereignisverarbeitung in der Produktionssteuerung mit Fokus auf die Anpassung des Ablaufplans	89
4.6	Bewertung der Ansätze zur Nutzung von bestärkendem Lernen in der Produktionssteuerung der Werkstattfertigung	91
5.1	Schritt 1: Ereignisraum definieren	104
5.2	Schritt 2: Zustandsraum definieren	107
5.3	Schritt 3: Aktionsraum definieren	111
5.4	Schritt 4: Simulationsmodell vorbereiten	114
5.5	Schritt 5: Daten integrieren	116
5.6	Schritt 6: Belohnungsfunktion definieren	119
5.7	Schritt 7: Training initialisieren	122
5.8	Übersicht der Modellparameter	125
5.9	Schritt 8: Episodenschritt ausführen	125
5.10	Schritt 9: Belohnung berechnen	126
5.11	Schritt 10: Modellparameter anpassen	127

5.12 Schritt 11: Modell evaluieren	129
5.13 Schritt 12: Betrieb initialisieren	134
5.14 Schritt 13: Aktion ausführen	136
5.15 Schritt 14: Belohnung berechnen	137
5.16 Schritt 15: Modellparameter anpassen	138
5.17 Schritt 16: Modell evaluieren	138
6.1 Übersicht der betrachteten Algorithmen	158
6.2 Untersuchte Variation der Hyperparameter	163
6.3 Untersuchte Variation der Agentenparameter	166
6.4 Vergleich der Hyperparameter und Agentenparameter der Basislinie und der Trainingslinie	168
6.5 Untersuchte Variation der Ereignisse zur Grenzbetrachtung	171
A.1 Klassifikation von Ereignissen für produzierende Unternehmen in Wertschöpfungsnetzwerken	234
A.2 Einordnung der Maßnahmen der Kapazitätssteuerung anhand der Charakteristik der komplexen Werkstattfertigung	241
A.3 Einordnung der Maßnahmen der Reihenfolgebildung anhand der Charakteristik der komplexen Werkstattfertigung	242
A.4 Hyperparameter	244
A.5 Agentenparameter	245
A.6 Trainingsparameter	246

1 Einleitung

1.1 Ausgangssituation

Unternehmen bewegen sich heute in einem Umfeld, das, wie in Abbildung 1.1 dargestellt, geprägt ist durch den ökologischen, gesellschaftlichen, wirtschaftlichen und technologischen Wandel (Bauernhansl et al. 2018, S. 6). Es ist charakterisiert durch eine zunehmende Volatilität, Unsicherheit, Komplexität und Mehrdeutigkeit¹ (Mack & Khare 2016, S. 5). Gleichzeitig haben das Umfeld sowie die Veränderungen einen maßgeblichen Einfluss darauf, mit welchen Ressourcen zukünftig welche Produkte für welche Kunden und Märkte produziert werden sowie welche Technologien hierfür zum Einsatz kommen (Bauernhansl et al. 2018, S. 6).

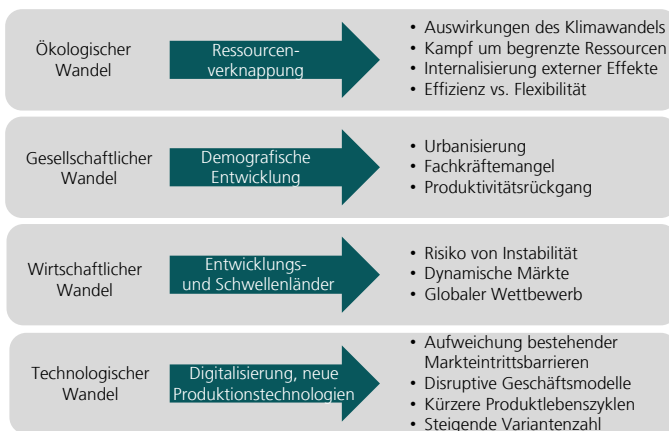


Abbildung 1.1: Umfeld und Herausforderungen für produzierende Unternehmen (eigene Darstellung in Anlehnung an Schiffer et al. 2020, S. 13; Bauernhansl et al. 2018, S. 6; Sauer 2018, S. 7; Reinhart & Zühlke 2017, S. XXV; Abele & Reinhart 2011, S. 10)

¹Aus dem Militär existiert für diese Charakterisierung das englische Akronym VUCA, welches sich aus Volatility, Uncertainty, Complexity und Ambiguity zusammensetzt (Mack & Khare 2016, S. 5).

Seit den 1980er Jahren wird der Markt mehr und mehr vom Verkäufermarkt zum Käufermarkt. Es geht nicht mehr darum, nur zu produzieren, da die Produkte definitiv verkauft werden, sondern darum, exakt auf Kundenwünsche einzugehen (Bauernhansl 2017a, S. 3). Daraus resultiert bei zunehmender Volatilität und Unsicherheit der Märkte ein Übergang vom Mengen- zum Variantenwachstum (Schiffer et al. 2020, S. 18; Reinhart & Zühlke 2017, S. XXXVII; Schuh & Riesener 2017, S. 3). Das Konzept der Mass Personalization, hochpersonalisierte Leistungen zu geringen Kosten nahe der Massenproduktion, geht noch einen Schritt weiter und stellt den personalisierten Nutzen für den Kunden in den Mittelpunkt (Bauernhansl et al. 2018, S. 22; Bauernhansl 2017b, S. 107; Hu 2013, S. 6). Zu diesem Kundennutzen zählt auch die Erfüllung logistischer Zielgrößen (Lödding 2016, S. 1; Mönch et al. 2011, S. 583). Im Rahmen einer Studie der Technischen Universität München gaben 170 befragte Unternehmen an, dass die Liefertreue für ihr Unternehmen die wichtigste logistische Zielgröße ist (Nyhuis et al. 2016, S. 9).

Die Komplexität der Märkte und des Unternehmensumfelds nimmt stetig zu (Bauernhansl et al. 2014, S. 347). Für produzierende Unternehmen in einem komplexen Umfeld wird es deshalb zunehmend wichtiger, resilient gegenüber äußeren Einflüssen zu sein (Schiffer et al. 2020, S. 35; Ivanov & Dolgui 2020, S. 2906; Ponomarov & Holcomb 2009, S. 128). Unternehmen reagieren auf diese Veränderungen des Unternehmensumfelds einerseits durch die Beeinflussung von unternehmensinternen Stellgrößen und andererseits durch die Organisation in Wertschöpfungsnetzwerken (Schuh et al. 2011b, S. 473–476; Schenk et al. 2012, S. 426). Das Agieren in Wertschöpfungsnetzwerken ermöglicht es Unternehmen, sich auf die eigenen Kernkompetenzen zu konzentrieren (Schuh et al. 2011b, S. 476; Schenk et al. 2012, S. 426). In den Leistungserstellungsprozess sind deshalb eine Vielzahl unterschiedlicher Partner eingebunden, welche durch deren Know-how sowie deren Produkt- und Prozesstechnologien charakterisiert sind (Schuh & Riesener 2017, S. 3; Schuh et al. 2011b, S. 476). Für Unternehmen wird es deshalb zunehmend wichtiger, in Geschäftsmodellen zu denken, aber in Kooperationen zu handeln (Bauernhansl 2016, S. 23). Die derzeitige politische und wirtschaftliche Entwicklung lässt annehmen, dass der Trend zur globalen Verteilung der Wertschöpfung weiter anhalten wird (Schenk et al. 2012, S. 426–427).

Wertschöpfungsnetzwerke sind der Schlüssel für nachhaltigen Erfolg. Das Management der Netzwerke stellt Unternehmen aber vor Herausforderungen (Friedli & Schuh 2012,

S. 187). Insbesondere die zunehmende Volatilität und Komplexität der Märkte sowie die steigende Kundenorientierung tragen zu diesen Herausforderungen bei und führen häufig zu Unsicherheiten im Wertschöpfungsnetzwerk. Unsicherheiten wiederum bedingen Fehler bei Prognosen sowie höhere Sicherheitsbestände, um diese Fehler ausgleichen zu können. Je größer dabei die Unsicherheit ist, desto wichtiger ist es für Wertschöpfungsnetzwerke, eine hohe Reaktionsfähigkeit zu besitzen (Chopra 2018, S. 24-28). Bei einer innovativen Auseinandersetzung mit den Problemen liegt deshalb in der Reaktion auf Herausforderungen eine Chance für Unternehmen, Wettbewerbsvorteile zu generieren, die zu unternehmerischem Erfolg führen (Möller 2011, S. 149). Dies gilt einerseits für globale Herausforderungen, andererseits auch für Problemstellungen in Wertschöpfungsnetzwerken.

Bedingt durch den Aufbau globaler Wertschöpfungsnetzwerke nehmen standortübergreifende logistische Prozesse zu. Dies führt zu Abstimmungsbedarf in der Planung und Steuerung auf verschiedenen Ebenen (Schenk et al. 2012, S. 427). Die Verfügbarkeit der notwendigen Informationen ist hier der Schlüssel, da Unsicherheit nichts anderes als das Fehlen der erforderlichen Informationen ist (Chopra 2018, S. 33). Ganzheitliche und durchgängige Informationsflüsse über alle Partner eines Wertschöpfungsnetzwerks sind in der industriellen Praxis heute jedoch selten vorhanden (Schiffer et al. 2020, S. 36; Schenk et al. 2012, S. 427–428). Die Verknüpfung des Wertschöpfungsnetzwerks mit der Produktionsplanung und -steuerung (PPS) erfolgt heute im Wesentlichen durch Planungsobjekte, beispielsweise Produktionsmengen oder Produktionstermine (Schenk et al. 2012, S. 429).

Die Diskussion um die Erreichung logistischer Ziele wird deshalb derzeit von der Produktionsplanung dominiert. Ein Produktionsplan führt jedoch nur zum Erfolg, wenn die Produktionssteuerung den Plan auch umsetzen kann. Die Bedeutung der Produktionssteuerung nimmt deshalb zu, da Prognosen fehleranfällig sind und daher eine schnelle Reaktion auf Planänderungen notwendig ist. Eine gute Produktionssteuerung sorgt folglich für eine robuste Produktion (Wiendahl & Wiendahl 2019, S. 245; Löffding 2016, S. 2; Schuh et al. 2011a, S. 63).

1.2 Problemstellung

Die Produktionssteuerung hat die Aufgabe, das Produktionsprogramm auch bei unvermeidbaren Ereignissen möglichst wirtschaftlich zu realisieren (Wiendahl & Wiendahl 2019, S. 245; Lööding 2016, S. 3). Ereignisse umfassen dabei einerseits regelmäßige Vorgänge, beispielsweise den Eingang eines Kundenauftrags. Andererseits werden auch unvorhergesehene Vorgänge mit negativen Auswirkungen, also Störungen, als Ereignisse bezeichnet (Lööding 2016, S. 3; Fleischmann et al. 2015, S. 74; Günthner 2008, S. 377). Des Weiteren kann unterschieden werden zwischen Ereignissen mit unternehmensinternen sowie unternehmensexternen Ursachen (Westkämper et al. 2000, S. 203).

Externe Ereignisse umfassen jene Quellen, welche von außerhalb des Unternehmens, also über das Wertschöpfungsnetzwerk, auf die Produktion einwirken. Auslöser hierfür sind der Markt sowie die Kunden des Unternehmens (Westkämper et al. 2000, S. 203). Bedingt durch deren zunehmende Dynamik wächst auch die Anzahl der externen Ereignisse, auf die Unternehmen reagieren müssen (vgl. Abschnitt 1.1). Gleichzeitig erfordert der Kundennutzen als Mittelpunkt der unternehmerischen Aktivitäten trotz Störeignissen die Einhaltung von Lieferterminen und die Erfüllung benötigter Produktionsmengen als Schlüssel für die Kundenzufriedenheit (Lööding 2016, S. 1; Mönch et al. 2011, S. 583). Für Partner in Wertschöpfungsnetzwerken steht deshalb die effiziente Reaktion auf Ereignisse im Fokus (Stadtler 2015, S. 20). So ist bei Unternehmen, die sehr flexibel auf Kundenwünsche reagieren, ein Trend zu immer mehr kurzfristigen Umplanungen und Steuerungseingriffen zu beobachten. Das stört den Produktionsablauf und erhöht die Durchlaufzeit der anderen Aufträge. Konsequenterweise entstehen dann immer neue Eilaufträge, die höher priorisiert und eingeplant werden müssen (Schuh et al. 2011a, S. 64; Zäpfel 2001, S. 226). Bereits in geringer Anzahl beeinflussen diese Eilaufträge die Produktion negativ (Waschneck et al. 2016, S. 7). Wiendahl & Wiendahl (2019, S. 247) beschreiben diesen Zusammenhang im Fehlerkreis der Produktionssteuerung (vgl. Abbildung 1.2).

Im Produktionsablauf ist immer mit der Möglichkeit des Auftretens der beschriebenen Ereignisse zu rechnen, die aus unvorhergesehenen Zuständen und Aktivitäten resultieren (Corsten & Gössinger 2016, S. 586). Traditionelle Ansätze für die PPS unterstützen jedoch keine schnelle Reaktion auf plötzlich auftretende Ereignisse (Schuh et al. 2011a, S. 66;



Abbildung 1.2: Fehlerkreis der Produktionssteuerung (eigene Darstellung in Anlehnung an Wiendahl & Wiendahl 2019, S. 247)

Zäh et al. 2010, S. 300–301; Kuhn & Hellingrath 2002, S. 130). Praktisch liegt dies häufig an langen Berechnungsläufen der Informationstechnologie (IT)-Systeme. Konzeptuell liegt dies jedoch auch darin begründet, dass eine Abstimmung über Unternehmensgrenzen hinweg in der PPS oftmals nicht vorgesehen ist (Kuhn & Hellingrath 2002, S. 130). Starre Prozesse können die zunehmende Dynamik somit nur unzureichend abbilden (Zäh et al. 2010, S. 300–301; Günthner 2008, S. 377).

Die Reaktion der PPS auf unvorhergesehene Ereignisse ist durch drei Defizite geprägt: eine mangelnde Datenqualität, eine fehlende Anpassung von Steuerungsmodellen sowie zu viele manuelle Eingriffe in die Steuerung.

Eine hohe *Datenqualität* ist für die PPS von großer Wichtigkeit (Reinhart & Gyger 2008, S. 302). Liegen Daten nicht in der notwendigen Zeit- und Objektgranularität sowie mit dem richtigen Inhalt am richtigen Ort vor, führt dies zu hohen Kosten im Unternehmen, die sich über höhere Bestände, zusätzliches Personal und Fehlerfolgekosten auswirken (Fleisch et al. 2005, S. 16–17). Insbesondere auch durch manuelle Datenerfassung und eine daraus resultierende mangelnde Datenqualität, ist bis heute eine mangelhafte Integration der realen und digitalen Welt in Unternehmen zu beobachten (Lödding 2020, S. 15–16;

Fleisch et al. 2005, S. 3–4). Zeitlich ist dies damit begründet, dass bestehende IT-Systeme trotz fortgeschrittener Entwicklung Daten oftmals nicht regelmäßig aktualisieren (Schuh et al. 2011a, S. 63). Inhaltlich bestehen die Defizite in den Schwächen durchgängiger Informationsflüsse sowie einer Verknüpfung zwischen Wertschöpfungsnetzwerk und PPS nur auf Planungsebene (vgl. Abschnitt 1.1). Der Integration von Daten aus dem Wertschöpfungsnetzwerk in die Produktionssteuerung und den darauf aufbauenden automatischen Entscheidungen wird jedoch großes Potenzial zugeschrieben (Bauernhansl 2017a, S. 27; Waschneck et al. 2016, S. 7).

Heute werden Elemente der Produktion typischerweise einmalig aufgenommen und digitalisiert, da die *Anpassung von Steuerungsmodellen* in IT-Systemen häufig nur mit hohem manuellen Aufwand möglich ist (Siegert et al. 2017, S. 128; Schuh et al. 2011a, S. 67). Änderungen in der Planungs- und Steuerungslogik werden deshalb oft nicht umgesetzt, Unternehmen arbeiten mit einer sogenannten Planungsrealität (Schuh et al. 2011a, S. 63–67). Die Informationsnachführung zwischen dem aktuellen Zustand und dessen Abbild in den IT-Systemen ist damit nicht gegeben (Schuh et al. 2011a, S. 67). Durch die zunehmenden Abweichungen wird die Komplexität des Systems größer (Siegert et al. 2017, S. 128). Um korrekte Entscheidungen in IT-Systemen zu treffen, ist daher eine kontinuierlich aktualisierte datenbasierte Abbildung notwendig (Siegert et al. 2018, S. 180). Des Weiteren werden Sachverhalte in digitalen Modellen oftmals zu sehr vereinfacht, da Entwickler nicht alle potenziell möglichen Situationen und Änderungen über die Zeit antizipieren können (Russell & Norvig 2021, S. 651; Schuh et al. 2011a, S. 63). Ansätze für die Nachführung von Modellen sowie die Abbildung komplexer Sachverhalte bieten Technologien aus dem Feld der Künstlichen Intelligenz (KI), also die Verbesserung der Leistung für zukünftige Aufgaben basierend auf Einflüssen der Umwelt und damit eine kontinuierliche Erweiterung der Wissensbasis (Russell & Norvig 2021, S. 651; Westkämper & Löffler 2016, S. 185). Das zunehmende Interesse an diesen Technologien zeigt sich laut Russell & Norvig (2021, S. 27) anhand einer Verzwanzigfachung der wissenschaftlichen Veröffentlichungen zwischen 2010 und 2019. In einer Studie von adesso sehen 82 Prozent der Befragten Wettbewerbsvorteile für Unternehmen durch KI innerhalb der nächsten fünf Jahre (adesso 2021, S. 13). Manufuture.de und eine Expertenkommission der Europäischen Union sehen in KI eine Schlüsseltechnologie zur Sicherung der Wettbewerbsfähigkeit der europäischen Industrie

(Bauernhansl et al. 2018, S. 22; Rüttgers et al. 2018, S. 22). Nach einer Studie des Leibniz-Zentrums für Europäische Wirtschaftsforschung trägt KI in Unternehmen, die sie einsetzen, bereits etwa ein Fünftel zur Umsatzrendite bei (Rammer 2020, S. 24).

Die Zunahme *manueller Eingriffe in die Steuerung* ist bedingt durch die fehlende Modellanpassung in IT-Systemen. Mitarbeiter nutzen deshalb Werkzeuge wie beispielsweise Microsoft Excel-Tabellen, um Produktionsdaten zu visualisieren und Steuerungsentscheidungen abzuleiten und damit die Produktionssteuerung zu übersteuern (Lödding 2020, S. 28–32; Schuh et al. 2011a, S. 64). Manuelle Eingriffe in die Produktionssteuerung führen jedoch wegen der Komplexität der Produktion oft zu seltsamem oder unerwartetem Verhalten sowie zum Unter- oder Übersteuern (Mönch et al. 2011, S. 584; Lödding 2016, S. 357; Robinson 2004, S. 6). Bereits geringe Änderungen und Steuereingriffe können dabei, insbesondere bei einer längeren Wirkperiode, große Auswirkungen haben (Schuh et al. 2013, S. 482). Trotzdem gab in einer Studie der Technischen Universität München nur ein geringer Anteil der befragten Unternehmen an, auf Störungen systemunterstützt zu reagieren. Der überwiegende Anteil reagiert in der Produktionssteuerung manuell (Niehues 2017, S. 233). In der Mehrheit der produzierenden Unternehmen werden Mitarbeiter auch nicht durch automatisch abgeleitete Entscheidungsalternativen unterstützt (Lödding 2020, S. 19).

Zusammenfassend kann festgestellt werden, dass trotz Fortschritten in der IT das Potenzial der Produktionssteuerung hinsichtlich der Erreichung logistischer Zielgrößen und damit auch der Kompensation von Ereignissen aus dem Wertschöpfungsnetzwerk nicht ausgeschöpft wird (Lödding 2016, S. 1). Dies ist insbesondere darin begründet, dass Daten in mangelnder Qualität vorliegen, Steuerungsmodelle nicht angepasst werden und zu viele manuelle Eingriffe in die Steuerung vorgenommen werden (Reinhart & Gyger 2008, S. 302; Schuh et al. 2011a, S. 63–64; Lödding 2016, S. 357). Dabei wird erwartet, dass durch neue IT-Technologien zur Integration von Wertschöpfungsnetzwerk und Produktion die Bestandskosten um 30 bis 40 Prozent reduziert werden könnten (Bauernhansl 2017a, S. 27; Lödding 2016, S. 140). Erreicht wird dies über eine IT-gestützte Reaktion auf Abweichungen, bevor diese Störungen auslösen und nachgelagerte Prozessschritte negativ beeinflussen (Reinhart & Zühlke 2017, S. XXXIX). Während zur Verbesserung der Datenqualität bereits Ansätze vorhanden sind, existieren bisher trotz dieses Potenzials insbesondere zur gemeinsamen

Abbildung der Defizite der Anpassung von Steuerungsmodellen und manuellen Eingriffen in die Steuerung keine Lösungsansätze.

1.3 Zielsetzung und Forschungsfragen

Die Zielsetzung im Rahmen der vorliegenden Arbeit ist deshalb die Optimierung der Liefertermineinhaltung bei Ereignissen aus dem Wertschöpfungsnetzwerk in der Produktionssteuerung der komplexen Werkstattfertigung. Hierfür ist ein lernfähiges Entscheidungsmodell notwendig, um dessen kontinuierliche Anpassung über die Zeit zu ermöglichen. Gleichzeitig sollen damit bislang übliche manuelle Eingriffe in die Produktionssteuerung vermieden werden. Diese Zielsetzung wird durch die konsequente Beantwortung der Forschungsfragen erreicht.

Nach Kubicek (1977) ist eine zentrale Leistung im Forschungsprozess die Formulierung von Forschungsfragen. Ausgerichtet auf das gewählte Forschungsdesign führen diese Forschungsfragen zu Erfahrungswissen, um die Entwicklung von Theorien zu ermöglichen. Ihre Legitimation erhalten Forschungsfragen dabei aus dem mit der Beantwortung erzielbaren Erkenntnisgewinn (Kubicek 1977, S. 14–16). Zur Formulierung der relevanten Forschungsfragen ist die Arbeit zunächst jedoch einzugrenzen, denn jede Theorie muss einige Dinge vernachlässigen, um sich auf andere zu konzentrieren (Malone & Crowston 1994, S. 100). In Anlehnung an das Vorgehen nach Braun (2016) lässt sich das vorliegende Forschungsthema wie folgt eingrenzen:

- **Untersuchungsgegenstand** ist die automatisierte Entscheidungsfindung in der Produktionssteuerung für den Umgang mit Ereignissen aus dem Wertschöpfungsnetzwerk durch die Anwendung des Konzepts adaptiver Regelkreise auf die Schnittstelle zwischen Wertschöpfungsnetzwerk und Produktionssteuerung.
- **Untersuchungsmotivation** ist die zunehmende Relevanz der Einhaltung von Lieferterminen für die unternehmerische Exzellenz, insbesondere aus Kundenperspektive.
- **Einflussfaktoren** mit signifikanten Auswirkungen auf die Untersuchung sind die Charakteristika der komplexen Werkstattfertigung, deren Wertschöpfungsnetzwerke sowie die veränderten Anforderungen an die Produktionssteuerung.

- **Untersuchungsziel** ist die Entwicklung einer automatisierten, situationsabhängigen Reaktion auf Ereignisse aus dem Wertschöpfungsnetzwerk in der Produktionssteuerung der komplexen Werkstattfertigung.

Aus dieser Eingrenzung lässt sich folgende grundlegende Forschungsfrage ableiten:

Wie kann eine **lernende**, auf **Ereignissen aus dem Wertschöpfungsnetzwerk** basierte Optimierung der **Produktionssteuerung** für die **komplexe Werkstattfertigung** gestaltet werden?

Die Eingrenzung sowie die grundlegende Forschungsfrage präzisieren die Problemstellung und erlauben die Entwicklung einer adaptiven Reaktion, um abhängig von der jeweiligen Situation auf Ereignisse des Wertschöpfungsnetzwerks in der Produktionssteuerung der komplexen Werkstattfertigung zu reagieren. Die grundlegende Forschungsfrage kann in die nachfolgenden Teilforschungsfragen untergliedert werden:

1. Was sind die relevanten Bestandteile der komplexen Werkstattfertigung und ihres Wertschöpfungsnetzwerks, um die Produktionssteuerung basierend auf Ereignissen aus dem Wertschöpfungsnetzwerk zu optimieren?
2. Welche Anforderungen werden an eine lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung gestellt?
3. Wie kann das Konzept einer lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung aufgebaut werden?
4. Wie kann eine lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung implementiert und simulativ getestet werden?

1.4 Wissenschaftstheoretische Positionierung und Forschungskonzeption

Wissenschaft bezeichnet menschliches Denken, das nach bestimmten Regeln stattfindet und umfasst Aktivitäten, welche zum Verständnis von Phänomenen beitragen (Vaishnavi & Kuechler 2015, S. 9; Ulrich & Hill 1976, S. 305). Ergebnisse wissenschaftlicher Tätigkeit,

die solchen Regeln, abhängig unter anderem vom Forschungsgebiet, genügen, werden als wissenschaftliche Erkenntnisse bezeichnet (Ulrich & Hill 1976, S. 305). Grundlegend systematisiert werden können Wissenschaften durch die Gliederung in Formal- und Realwissenschaften (vgl. Abbildung 1.3).

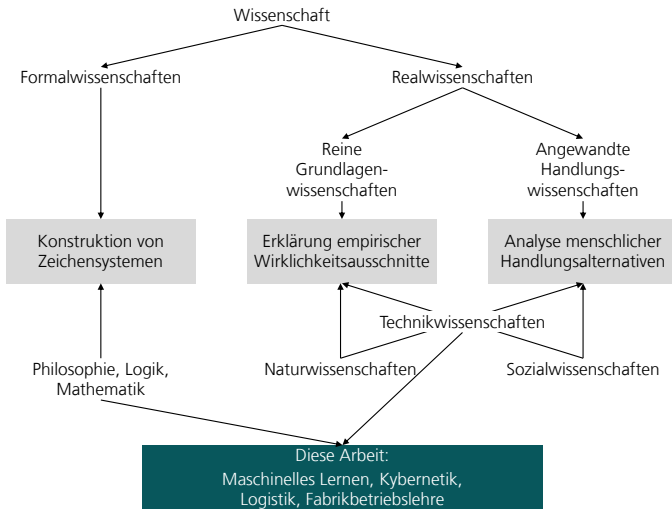


Abbildung 1.3: Wissenschaftssystematik (eigene Darstellung in Anlehnung an Ulrich & Hill 1976, S. 305)

Formalwissenschaften umfassen die Konstruktion von Zeichensystemen und Regeln zur Verwendung dieser Zeichen (Ulrich & Hill 1976, S. 305). Sie treffen also analytische Aussagen (Schanz 1987, S. 2039). Formalwissenschaften sind nur auf ihre logische Wahrheit hin überprüfbar. Die Wissenschaftsdisziplinen der Philosophie, Logik und Mathematik werden den Formalwissenschaften zugerechnet (Ulrich & Hill 1976, S. 305). Aussagen der Realwissenschaften sind synthetisch (Schanz 1987, S. 2039). Sie beschreiben, erklären und gestalten sinnlich wahrnehmbare Wirklichkeitsausschnitte (Ulrich & Hill 1976, S. 305). Realwissenschaften können weiter untergliedert werden in reine Grundlagenwissenschaften und angewandte Handlungswissenschaften. Grundlagenwissenschaften versuchen, Erklärungsmodelle zu bilden. Hierzu zählen die Naturwissenschaften (Ulrich & Hill 1976, S. 305). Angewandte Handlungswissenschaften hingegen analysieren menschliche Handlungsalternativen mit dem Fokus auf die Gestaltung sozialer und technischer Systeme.

Hierfür werden Entscheidungsmodelle und Entscheidungsprozesse entwickelt. Die Sozialwissenschaften zählen zu den angewandten Handlungswissenschaften (Ulrich & Hill 1976, S. 305). Angewandte Handlungswissenschaften sind dabei auf einen praktischen Nutzen oder eine praktische Zielsetzung hin ausgerichtet (OECD 2002, S. 30). An der Schnittstelle zwischen reinen Grundlagenwissenschaften und angewandten Handlungswissenschaften sind die Technikwissenschaften anzusiedeln. Sie zielen auf die Erforschung und Entwicklung künstlicher Artefakte mit dem Ziel der Erprobung einer praktischen Funktionsfähigkeit ab² (Ropohl 2009, S. 161–162).

Die vorliegende Problemstellung lässt sich den Technikwissenschaften zuordnen, wohingegen zur Lösung auch Methoden aus den Formalwissenschaften eingesetzt werden. Konkret sind dies Methoden des Maschinellen Lernens, einer Wissenschaftsdisziplin, welche sich mit der automatischen Verbesserung im Laufe der Zeit sowie mit Inferenz und Entscheidungsfindung unter Unsicherheit beschäftigt (Jordan & Mitchell 2015, S. 255-256). Aus den Technikwissenschaften kommen daneben Methoden der Kybernetik, der Wissenschaftsdisziplin zur Erforschung des Verhaltens sowie der Steuerung und Regelung technischer und biologischer Systeme, zum Einsatz (Wiener 1961, S. 11). Der kybernetische Managementansatz leitet dabei aus der Analogie zwischen Unternehmen und Regelkreisen Grundsätze ab, ohne das Unternehmen mathematisch als Regelkreis zu beschreiben oder zu berechnen (Warnecke 1989, S. 111). Die Problemstellung folgt aus der Schnittstelle zwischen den Wissenschaftsdisziplinen der Logistik und der Fabrikbetriebslehre. Das integrale Logistikmanagement umfasst dabei das Management des Wertschöpfungsnetzwerks entlang des gesamten Produktlebenszyklus, unternehmensintern und unternehmensübergreifend (Schönsleben 2016, S. 11). Die Fabrikbetriebslehre befasst sich mit der systematischen Gestaltung von wirtschaftlichen Strukturen und Prozessen produzierender Unternehmen aus organisatorischen und technologischen Gesichtspunkten sowie dem Einfluss des Menschen (Bauernhansl & Mieke 2020, S. 3).

Forschung kann als Lernprozess der beteiligten Forscherinnen und Forscher begriffen werden, in welchem zu untersuchende Problemstellungen basierend auf dem Erfahrungswissen theoretisch abgegrenzt und präziser definiert werden (Kubicek 1977, S. 14). Tomczak

²Technikwissenschaften werden in verschiedenen Quellen auch als Ingenieurwissenschaften bezeichnet. Da diese jedoch weniger Ingenieurinnen und Ingenieure, sondern deren Tätigkeitsfeld zum Gegenstand haben, wird zunehmend von Technikwissenschaften gesprochen (Ropohl 2009, S. 161).

(1992, S. 83–84) beschreibt den Lernprozess in seinem explorativen Forschungszyklus als iterativen Lernprozess. Die Ergebnisse eines solchen Lernprozesses können in der Lehre sowie der Beratung der Praxis Verwendung finden (Kubicek 1977, S. 16).

Design Science beschreibt ein Problemlösungsparadigma, welches darauf abzielt, Innovationen zu schaffen, welche Ideen, Praktiken, technische Fähigkeiten und Produkte definieren, durch die die Analyse, das Design, die Implementierung, das Management und die Nutzung von Artefakten, meist Informationssysteme, effektiv und effizient durchgeführt werden können (Hevner et al. 2004, S. 76). Artefakte, durch Menschen geschaffen, werden dabei in einer strukturierten Form beschrieben, die von Software, formaler Logik und Mathematik bis hin zu informellen Beschreibungen in natürlicher Sprache reichen kann (Vaishnavi & Kuechler 2015, S. 9; Hevner et al. 2004, S. 77; Simon 1996, S. 2–5).

Das Information Systems Research Framework nach Hevner et al. (2004) umfasst analog zum Forschungszyklus nach Tomczak (1992) ebenfalls ein iteratives Vorgehen zur Lösungsentwicklung in Form eines Gestaltungszyklus zwischen Entwicklung und Validierung (vgl. Abbildung 1.4). Dieser Gestaltungszyklus zielt auf die Schaffung von Artefakten ab, welche für die jeweilige Problemstellung anwendbar sind, und schließt damit den Kreis zu den Technikwissenschaften (Hevner et al. 2004, S. 80).

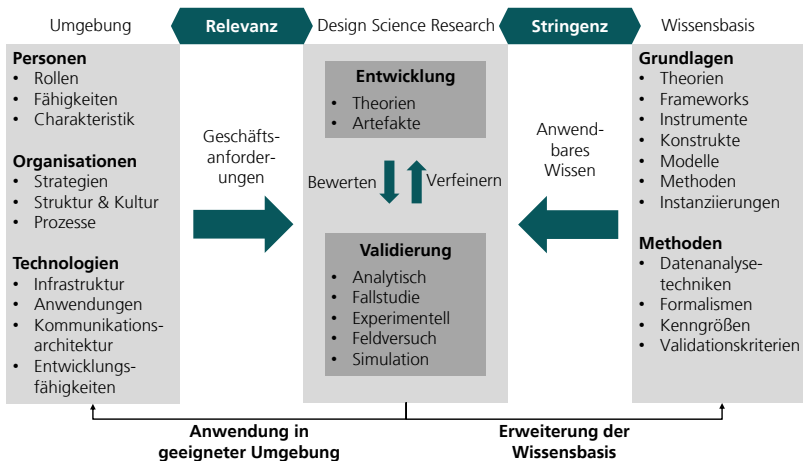


Abbildung 1.4: Information Systems Research Framework (eigene Darstellung in Anlehnung an Hevner et al. 2004, S. 80)

Die Problemstellung resultiert aus der kontextuellen Umgebung, konkreter den Personen, Organisationen und Technologien mit ihren Zielen, Aufgaben, Problemen und Möglichkeiten. Die Ausrichtung der Forschungsaktivitäten auf die Bedürfnisse dieser Umgebung stellt die Relevanz der Forschung sicher (Hevner et al. 2004, S. 79). Der sogenannte Relevanzzyklus verbindet dabei die kontextuelle Umgebung mit den Forschungsaktivitäten (Hevner 2007, S. 88). Die Wissensbasis mit Grundlagen und Methoden auf der anderen Seite bildet den theoretischen Bezugsrahmen und das Fundament für die Forschungsaktivitäten. Wissenschaftliche Erkenntnisse, welche aus den Forschungsaktivitäten resultieren, fließen wieder in die Wissensbasis ein (Vaishnavi & Kuechler 2015, S. 15; Hevner et al. 2004, S. 80–81). Dies beschreibt der sogenannte Stringenzzyklus (Hevner 2007, S. 88). Das Information Systems Research Framework beinhaltet damit drei inhärente Zyklen (Hevner 2007, S. 88).

1.5 Aufbau der Arbeit

Der Aufbau der Arbeit, dargestellt in Abbildung 1.5, orientiert sich an den Phasen der angewandten Forschung nach Ulrich (1981), welche sich auch in den Design-Science-Zyklen nach Hevner (2007) widerspiegeln. Abschnitt 1 umfasst die Konkretisierung der Ausgangssituation und Problemstellung, die Ableitung einer Zielsetzung und darauf aufbauend die Ableitung der Forschungsfragen und Gestaltung eines wissenschaftlichen Rahmens. Begriffe und Grundlagen der komplexen Werkstattfertigung, der Wertschöpfungsnetzwerke, der Produktionsplanung und -steuerung und der lernenden Systeme beschreibt Abschnitt 2. Neben dem methodischen Vorgehen zur Analyse des Stands der Wissenschaft und Technik zeigt Abschnitt 3 bestehende Ansätze zur Reaktion auf Ereignisse aus Wertschöpfungsnetzwerken, zur Ereignisverarbeitung in der Produktionssteuerung und zur Nutzung von bestärkendem Lernen in der Produktionssteuerung der Werkstattfertigung. In Abschnitt 4 werden Anforderungen definiert und basierend darauf die bestehenden Ansätze des Stands der Wissenschaft und Technik eingeordnet. Die Konzeption der lernenden ereignisbasierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung folgt in Abschnitt 5. Diese umfasst die Darstellung des methodischen Vorgehens, die Definition des Konzepts und die Detailbeschreibung der einzelnen Elemente. Implementiert,

appliziert und validiert wird die lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung anschließend in Abschnitt 6. Eine Reflexion hinsichtlich der Erfüllung der Anforderungen und der Beantwortung der Forschungsfrage beschreibt Abschnitt 7. In Abschnitt 8 wird die Arbeit zusammengefasst und ein Ausblick auf weiterführende Fragestellungen gegeben.

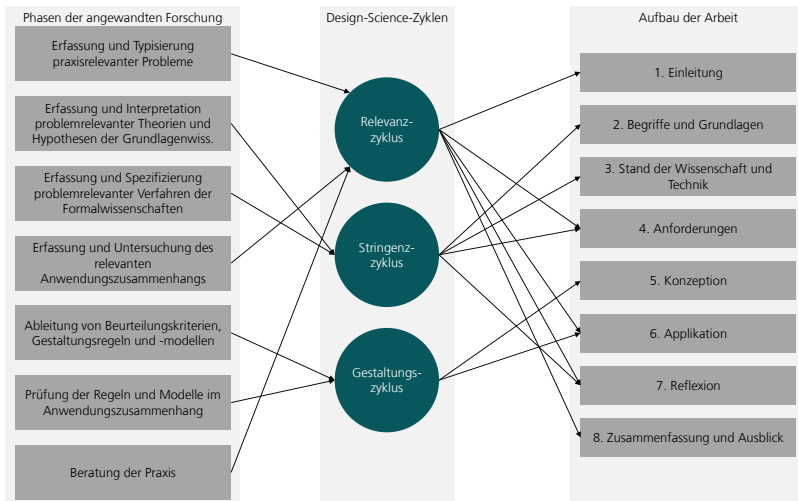


Abbildung 1.5: Aufbau der Arbeit (eigene Darstellung in Anlehnung an Hevner 2007, S. 88; Ulrich 1981, S. 20)

2 Begriffe und Grundlagen

Wann immer ein bestimmtes Phänomen der Realität sprachlich beschrieben werden soll, wird dazu ein Begriff benötigt (Schanz 1988, S. 17). Begriffe beschleunigen und vereinheitlichen das Verständnis zwischen Fachvertretern, indem sie neben der Funktion der Sprachverkürzung auch als Sprachregelung dienen. In der Wissenschaft ist eine präzise Definition von Begriffen deshalb eine unerlässliche Voraussetzung für das Verständnis und die Bildung von Theorien (Chmielewicz 1994, S. 51; Schanz 1988, S. 19–20).

Im vorliegenden Abschnitt werden folglich Begriffe definiert und in einen Kontext gesetzt, die für das weitere Verständnis der Arbeit notwendig sind. Dabei wird auch das Forschungsgebiet genauer eingegrenzt. Die Begriffsdefinition umfasst die Begriffe der komplexen Werkstattfertigung und des Wertschöpfungsnetzwerks (vgl. Abschnitt 2.1), der Produktionsplanung und -steuerung (vgl. Abschnitt 2.2) sowie der lernenden Systeme (vgl. Abschnitt 2.3).

2.1 Komplexe Werkstattfertigung und Wertschöpfungsnetzwerk

Nach Westkämper (2006, S. 200) ist die Produktion ein Transformationsprozess, welcher aus Rohmaterialien und Halbzeugen unter Verwendung natürlicher Ressourcen und Wissen in Fabriken Fertigteile und Produkte erzeugt (vgl. Abbildung 2.1). Die Produktion ist dabei als ein Prozess der Wertschöpfung zu verstehen, der von Menschen unter Zuhilfenahme von Produktions-, Mess-, Lager- und Transportmitteln durchgeführt wird. Dieser Prozess umfasst Verfahren der Fertigung und der Montage (Westkämper & Löffler 2016, S. 2; Westkämper 2006, S. 195–196; VDI 1992, S. 167). Das ökonomische Ziel der Produktion ist dabei die Transformation von Materialien zu Komponenten mit höherem Wert, also die Wertschöpfung (Groover 2020, S. 3). Dies umfasst die „Herstellung der richtigen Produkte nach Art und Menge, zum richtigen Zeitpunkt, in einer spezifizierten Qualität und zu akzeptablen Kosten“ (Westkämper 2006, S. 200). Nachfolgend wird die Charakteristik der komplexen Werkstattfertigung näher ausgeführt (vgl. Abschnitt 2.1.1). Anschließend wird diese in das

Wertschöpfungsnetzwerk eingebettet (vgl. Abschnitt 2.1.2) sowie dessen Ereignisse als Komplexitätstreiber aufgezeigt (vgl. Abschnitt 2.1.3). Es folgt die Darstellung der IT-Unterstützung für das Management von Wertschöpfungsnetzwerken (vgl. Abschnitt 2.1.4).

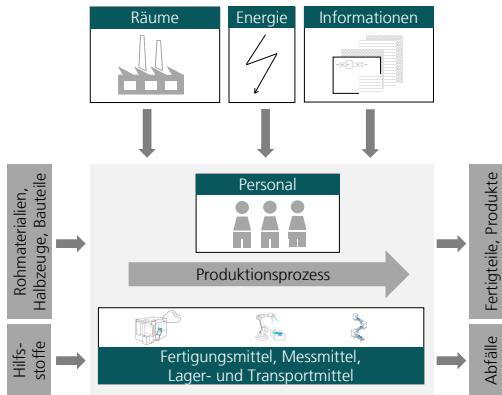


Abbildung 2.1: Aufbau und Elemente der Produktion (eigene Darstellung in Anlehnung an Westkämper 2006, S. 200)

2.1.1 Die Charakteristik der komplexen Werkstattfertigung

Die Fertigung umfasst „Verfahren und Einrichtungen zur Herstellung von Einzelteilen nach Maßgabe der Konstruktion“ (Westkämper 2006, S. 203). Ziel der Fertigung ist es, den Fertigstellungsgrad des Werkstücks zu erhöhen. Hierfür werden wertschöpfende Prozesse zur Änderung der Geometrie, der Eigenschaften oder des Aussehens angewandt (Groover 2020, S. 10; DIN 8580, S. 4–5). Manche Branchen stellen dabei besondere Anforderungen bezüglich der Umgebungsbedingungen wie Temperatur, Raumklima oder Reinheit der Luft, die das Fertigungskonzept zu berücksichtigen sind (Westkämper 2006, S. 208). Die Fertigung verursacht im Unternehmen deshalb oft die größten Kosten, die auch größer als die der Montage sind (Eversheim 1989, S. 5). In der Halbleiterfertigung entfallen beispielsweise bis zu 75 Prozent der Durchlaufzeit und ein großer Kostenanteil auf die Fertigung (Mönch et al. 2013, S. 14). Aufgrund der hohen Kosten und der langen Durchlaufzeiten sind in der Fertigung vielfältige Effizienzsteigerungen möglich. Die Untersuchung im Rahmen der vorliegenden Arbeit fokussiert sich deshalb auf die Fertigung.

Grundsätzlich lassen sich Fertigungen durch zwei Beschreibungsformen charakterisieren. Diese werden als Fertigungsart und Fertigungsprinzip bezeichnet. Die Fertigungsart leitet sich dabei aus den zu fertigen Mengen ab, wohingegen das Fertigungsprinzip die räumliche und organisatorische Struktur der Fertigung beschreibt. Das Fertigungsprinzip wird deshalb in der Literatur auch als Organisationstyp bezeichnet (Westkämper 2006, S. 198; Wiendahl & Wiendahl 2019, S. 43).

Die *Fertigungsart* wird über die Wiederholhäufigkeit eines Produktes, also die Anzahl der Lose pro Jahr, ermittelt (Eversheim 1989, S. 11). Weitere Kriterien sind die Art der Aufträge (Lagerfertigung, Kundenauftragsfertigung) und die Größe der Lose. Hierüber lässt sich die Fertigungsart in die Kategorien Einmalfertigung, Wiederholfertigung, Variantenfertigung, Serienfertigung und Massenfertigung einteilen (Westkämper 2006, S. 199; Wiendahl & Wiendahl 2019, S. 52). Zwar sind der Literatur Grenzwerte für die Wiederholhäufigkeiten der einzelnen Kategorien zu entnehmen, diese unterscheiden sich jedoch und haben sich über die Zeit auch verändert. Entsprechend sind diese Grenzwerte nur als Richtwerte anzusehen (Eversheim 1989, S. 11; Schuh et al. 2012c, S. 131). Eine ausführliche Darstellung der Fertigungsarten ist der Literatur zu entnehmen (Lödding 2016, S. 126; Westkämper 2006, S. 199; Schuh et al. 2012c, S. 130–131). Bisher wenig berücksichtigt in der Literatur zu den Fertigungsarten ist das Konzept der Mass Personalization, also die Erstellung personalisierter Produkte zu Kosten nahe der Massenproduktion (Bauernhansl 2017b, S. 107).

Das *Fertigungsprinzip* wird durch eine Vielzahl von Einflüssen des Unternehmens geprägt. Dazu zählen der Markt, die Kunden, die Technologien, das Personal und die Wirtschaftlichkeit (Wiendahl & Wiendahl 2019, S. 42). Grundsätzlich lassen sich fünf verschiedene Fertigungsprinzipien unterscheiden, das Werkbankprinzip, das Baustellenprinzip, das Werkstattprinzip, das Inselprinzip und das Fließprinzip (vgl. Abbildung 2.2). Ausführlich beschrieben werden alle Fertigungsprinzipien in der Literatur (Wiendahl & Wiendahl 2019, S. 44–48; Lödding 2016, S. 123–125; Westkämper 2006, S. 198–199). In der Industrie sind die Fertigungsprinzipien der Werkstattfertigung und der Fließfertigung am häufigsten anzutreffen (Wiendahl & Wiendahl 2019, S. 44).

In der Werkstattfertigung, auch funktionale Fertigung oder Fertigung nach dem Verrichtungsprinzip genannt, werden Bereiche technologisch orientiert nach den Bearbeitungsverfahren angeordnet. Werkstücke werden einzeln oder in Losen zwischen den Arbeitsplätzen

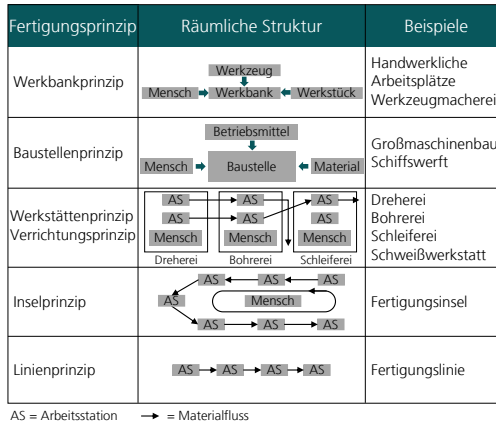


Abbildung 2.2: Unterscheidung der Fertigungsprinzipien (eigene Darstellung in Anlehnung an Wiendahl & Wiendahl 2019, S. 44)

transportiert, weshalb die Bearbeitungsreihenfolge flexibel an neue Werkstücke angepasst werden kann. Nachteilig ist die tendenziell lange Durchlaufzeit, die durch Transporte und Warten vor Arbeitsgängen begründet ist (Groover 2020, S. 17–18; Wiendahl & Wiendahl 2019, S. 44; Westkämper 2006, S. 199). Die Werkstattfertigung ist in der Industrie weit verbreitet (Lödding 2020, S. 14). Ein Subtyp der Werkstattfertigung ist die flexible Werkstattfertigung, die anstatt aus einer Anzahl von Maschinen aus einer Anzahl von sogenannten Workcentern besteht, die wiederum parallel angeordnete identische Maschinen umfassen. Jeder Auftrag hat seine eigene Route durch die Fertigung, wobei die Workcenter mehrmals durchlaufen werden können (Pinedo 2016, S. 15). Ein weiterer Subtyp der Werkstattfertigung und gleichzeitig eine in der Literatur bislang wenig explizit adressierte Sonderform ist das Fertigungsprinzip der komplexen Werkstattfertigung¹ (Mason et al. 2002, S. 248).

Im Grundsatz können Fertigungsarten und Fertigungsprinzipien frei miteinander kombiniert werden. Typisch sind in der Industrie jedoch beispielsweise die Einmal- oder geringe Wiederholfertigung nach dem Werkbankprinzip im Werkzeugbau oder die Herstellung von Standardteilen der Automobilindustrie in der Massenfertigung nach dem Fließprinzip (Wiendahl & Wiendahl 2019, S. 52; Westkämper 2006, S. 199). Während die Werkstattfertigung in der Produktionsliteratur meist der Einmal-, Wiederhol- und Variantenfertigung

¹engl. Complex Job Shop

zugeordnet wird (Wiendahl & Wiendahl 2019, S. 52; Westkämper 2006, S. 198), finden sich für die komplexe Werkstattfertigung auch wichtige Anwendungen in der industriellen Serien- und Massenfertigung. Erfolgreiche Anwendungen zeigen das Potenzial der komplexen Werkstattfertigung in der Serien- und Massenfertigung (vgl. Abschnitt 6.1). Typische Kombinationen von Fertigungsarten und Fertigungsprinzipien ergänzt, um die komplexe Werkstattfertigung, sind Abbildung 2.3 zu entnehmen.

		Fertigungsprinzipien						
		Werkbankprinzip						
		Bauteilenprinzip						
		Werkstättenprinzip						
		Komplexes Werkstättenprinzip						
		Fertigungsinsel						
		Fließprinzip, lose verketteter, nicht getakter						
		Taktprinzip, starr verketteter, getakter						
Fertigungsarten	Einmal-fertigung	X	X	X				
	Wiederhol-fertigung	X	X	X		X		
	Varianten-fertigung	X		X	X	X		
	Serien-fertigung				X	X	X	
	Massen-fertigung				X		X	X

Abbildung 2.3: Kombinationen von Fertigungsarten und Fertigungsprinzipien (eigene Darstellung in Anlehnung an Wiendahl & Wiendahl 2019, S. 52; Westkämper 2006, S. 198; Waschneck et al. 2017, S. 3)

„I think the next century (21st century) will be the century of complexity.“
Stephen Hawking (1942-2018)²

Für die Charakterisierung der komplexen Werkstattfertigung ist zuerst Komplexität zu definieren. Wenngleich der Begriff der Komplexität in der Literatur häufig verwendet wird, existiert keine universelle, präzise und weitgehend akzeptierte Definition (ElMaraghy et al. 2012, S. 794). Zur Definition der Komplexität hilft die Abgrenzung zwischen Kompliziertheit und Komplexität, da diese insbesondere in der Alltagssprache häufig vermischt werden.

Nach Ulrich & Probst (1995, S. 61) sind Systeme charakterisiert durch die Anzahl und Verschiedenheit der Elemente und Beziehungen, also der Vielfalt und Vielzahl, sowie der

² zitiert aus Mack & Khare 2016, S. 3.

(2012, S. 794) nehmen diese Unterscheidung nicht vor und gehen bei komplexen Systemen immanent von einer Vielzahl an Elementen und Beziehungen aus.

Komplexität besagt damit zweierlei, nämlich, dass nicht mehr alle Elemente eines Systems jederzeit miteinander verknüpft werden können sowie dass das System unbestimmt und unvorhersagbar ist (Schuh & Riesener 2017, S. 11; Luhmann 2015, S. 46). Das unbestimmte und unvorhersagbare Systemverhalten folgt aus der Dynamik, also den veränderlichen Wirkungsabläufen, und der Unsicherheit. Ein komplexes System ist deshalb nicht nur in seiner Zusammensetzung kompliziert, sondern verändert auch seinen Zustand kontinuierlich und nicht vorhersagbar. In einer gegebenen Zeitspanne können deshalb eine große Zahl verschiedener Systemzustände eingenommen werden (Ulrich & Probst 1995, S. 58). Die Unsicherheit in komplexen Systemen entsteht daraus, dass sich die Interaktionen zwischen den Elementen jederzeit verändern können (Ulrich & Probst 1995, S. 60). Im Rahmen dieser Arbeit soll Komplexität hinsichtlich eines Systems deshalb als äußerst komplexes System mit einer immanenten Dynamik und Unsicherheit verstanden werden.

Die komplexe Werkstattfertigung unterscheidet sich mit ihrer Charakteristik demnach deutlich von der Werkstattfertigung und der flexiblen Werkstattfertigung (Pinedo 2016, S. 15; Waschneck et al. 2016, S. 2; Mönch et al. 2013, S. 25; Mönch et al. 2011, S. 586–587; Mönch et al. 2003, S. 163; Mason et al. 2002, S. 249; Uzsoy et al. 1992, S. 49–50):

Hohe Varianz Die von den Kunden nachgefragten Produkte und deren Mengen variieren sehr stark und verändern sich auch über die Zeit. Es ist deshalb eine kontinuierliche Anpassung an die Kundenbedürfnisse und damit eine hohe Flexibilität bezüglich des Produktmixes erforderlich.

Unterschiedliche Losgrößen Die Losgrößen können sich stark unterscheiden, insbesondere, wenn Prototypen (typischerweise kleine Losgrößen) und Produktivlose (typischerweise größere, konstante Losgrößen) in der gleichen Umgebung gefertigt werden.

Inhomogene parallele Maschinen Auch die flexible Werkstattfertigung beinhaltet in den Workcentern bereits parallele Maschinen. Eignen sich diese jedoch trotz theoretischer Gleichartigkeit nicht gleichermaßen für bestimmte Produkte, werden die Maschinen als inhomogen bezeichnet und es wird versucht, je Produkt eine bestimmte Maschine auszuwählen.

Ungleiche Prozesszeiten Die Prozesszeiten unterscheiden sich zwischen den unterschiedlichen Arbeitsplätzen von wenigen Minuten bis zu einigen Stunden. Zusätzlich weichen auch die Prozesszeiten der unterschiedlichen Produkte an gleichen Arbeitsplätzen und Maschinen sehr stark voneinander ab.

Wiedereintretende Prozessflüsse Werkstücke fließen in wiedereintretenden Flüssen durch die Fertigung. Das bedeutet, dass Werkstücke mehrfach an einem Arbeitsplatz bearbeitet werden, beispielsweise, um verschiedene Schichten zu bearbeiten und aufzubringen. Unter Berücksichtigung der Inhomogenität kann dies auch bei jedem Durchlauf dieselbe Maschine bedingen.

Zeitabhängigkeiten Zwischen verschiedenen Prozessen können Zeitabhängigkeiten bestehen, die eine maximale Wartedauer bis zur Bearbeitung am nächsten Arbeitsplatz vorschreiben.

Sequenzabhängige Rüstzeiten Die Rüstzeiten der Arbeitsplätze sind sequenzabhängig und ändern sich damit bei einer anderen Auftragsreihenfolge. Typischerweise betrifft dies physikalische oder chemische Arbeitsgänge, bei denen Flüssigkeiten oder Gase ausgetauscht werden müssen.

Unterschiedliche Prozesstypen Je nach Prozesstyp ist die Art der Fertigung sehr unterschiedlich. So können einerseits einzelne Lose in sequentieller Reihenfolge bearbeitet werden, andererseits sind auch Prozesse vorhanden, welche Lose zu Chargen zusammenfassen.

Häufige Störungen Der Fertigungsablauf wird häufig von unvorhergesehenen Störungen unterbrochen, beispielsweise durch unvorhergesehene Maschinenausfälle.

Die Komplexität zeigt sich in dieser Charakteristik der komplexen Werkstattfertigung. So ist die Vielfalt und Vielzahl durch die Anzahl und Ausprägungen der möglichen Prozessrouten durch die Fertigung gegeben. Eine jederzeitige Verknüpfung aller Elemente ist durch ungleiche Prozesszeiten an gleichen Anlagen und Zeitabhängigkeiten zwischen Prozessen nicht möglich. Die Dynamik bildet sich über die hohe Varianz der Kundennachfragen und den damit kontinuierlich wechselnden Produktmix ab. Aus den häufigen Störungen folgt die Unsicherheit in der komplexen Werkstattfertigung.

2.1.2 Die komplexe Werkstattfertigung als Teil des Wertschöpfungsnetzwerks

Eine Lieferkette³ besteht aus miteinander verbundenen und aufeinander angewiesene Einheiten, die mit dem Ziel kooperieren, Kundenanforderungen effizient zu erfüllen. Dazu zählen produzierende Unternehmen und deren Standorte ebenso wie Lieferanten, Logistiker, Händler und zu beliefernde Kunden (Chopra 2018, S. 1; Stadtler 2015, S. 4; Blackstone 2013, S. 171; Christopher 2011, S. 4; Hopp 2008, S. 5). Der Begriff der Lieferkette impliziert dabei, dass je Stufe nur ein Beteiligter existiert. Tatsächlich können es jedoch je Stufe mehrere Beteiligte sein, beispielsweise mehrere Lieferanten und Kunden eines Produzenten. In diesem Zusammenhang, gepaart mit der Tatsache, dass das Ziel jeder Lieferkette die Wertgenerierung ist, ist der Begriff Wertschöpfungsnetzwerk der passendere Begriff (Schiffer et al. 2020, S. 13; Chopra 2018, S. 2–3; Christopher 2011, S. 3). Wertschöpfungsnetzwerke können unternehmensinterne Einheiten, beispielsweise verschiedene Werke, als auch unternehmensexterne Einheiten, also rechtlich selbständige Unternehmen, enthalten (Stadtler 2015, S. 4).

Aufgrund der veränderten Bedingungen in ihrem Umfeld (vgl. Abschnitt 1.1) organisieren sich produzierende Unternehmen zunehmend in Wertschöpfungsnetzwerken, was die Produktion zu einem Teil des Wertschöpfungsnetzwerks macht (Schenk et al. 2012, S. 426; Bruns & Dunkel 2010, S. 5). Unternehmen erwarten sich dadurch Kosteneinsparungen, steigende Profitabilität, zusätzliche Marktchancen, höhere Kundenzufriedenheit und agile Prozesse (Chopra 2018, S. 3; Schenk et al. 2012, S. 426–427; Christopher 2011, S. 3).

Solche Wertschöpfungsnetzwerke sind gekennzeichnet durch eine kooperative Leistungserstellung. Die Partner zielen damit darauf ab, auf das Wissen der Partner sowie auf deren Ressourcen, Produkt- und Produktionstechnologien zugreifen zu können (Schuh et al. 2011b, S. 476). Dabei spielt es keine Rolle, ob es sich um unabhängige, rechtlich separate Unternehmen in einem inter-organisatorischen Wertschöpfungsnetzwerk oder um ein Unternehmen mit global verteilten Standorten, also ein intra-organisatorisches Wertschöpfungsnetzwerk, handelt. Intra-organisatorische Wertschöpfungsnetzwerke können wiederum Teil von inter-organisatorischen Wertschöpfungsnetzwerken sein (Stadtler

³engl. Supply Chain

2015, S. 4). So findet oftmals ein Großteil der Wertschöpfung in produzierenden Unternehmen an global verteilten Standorten in Produktionsnetzwerken statt, die teilweise sehr komplex sein können. Die Steuerung des Wertschöpfungsnetzwerks hingegen wird jedoch häufig zentral vom Stammsitz des Unternehmens aus vorgenommen (Friedli & Schuh 2012, S. 187). Die Wertschöpfungskette⁴ nach Porter (1985, S. 37) weitet sich deshalb über Unternehmensgrenzen hinweg aus. Wert wird nicht mehr durch ein einzelnes Unternehmen generiert, sondern durch alle verbundenen Partner im Wertschöpfungsnetzwerk. In Zukunft stehen deshalb nicht mehr einzelne Unternehmen im Wettbewerb, sondern Wertschöpfungsnetzwerke (Stadtler 2015, S. 4; Christopher 2011, S. 11).

Das Ziel der Planung von Wertschöpfungsnetzwerken ist die effiziente Strukturierung der Waren-, Informations- und Geldflüsse zwischen den Partnern zur Erfüllung der Kundenanforderungen (Chopra 2018, S. 2; Blackstone 2013, S. 171). Diese Flüsse sind oftmals bidirektional und werden entweder von einem Partner oder einem Intermediär verwaltet (Chopra 2018, S. 2). In einem Wertschöpfungsnetzwerk müssen oft bis zu tausendfach individuelle Entscheidungen pro Minute getroffen werden (Fleischmann et al. 2015, S. 71). Diese haben einen starken Einfluss auf die Profitabilität und den Erfolg des Wertschöpfungsnetzwerks (Chopra 2018, S. 7). Die Planungs- und Entscheidungsaufgaben lassen sich dabei nach dem Zeithorizont und dem betroffenen Prozess des Wertschöpfungsnetzwerks gliedern (vgl. Abbildung 2.5).

Die Charakteristik eines Wertschöpfungsnetzwerks mit den assoziierten Entscheidungen zur Koordination von Waren-, Informations- und Geldflüssen und deren inhärenter Dynamik ist dabei ähnlich komplex wie die Charakteristik einer Fertigung (vgl. Abschnitt 2.1.1; Serdarasan 2013, S. 533; Hopp 2008, S. 8–9). Die Komplexität in Wertschöpfungsnetzwerken ist deshalb Gegenstand zahlreicher wissenschaftlicher Untersuchungen (Macchion et al. 2020; Bode & Wagner 2015; Serdarasan 2013; Wilding 1998). Auch in Wertschöpfungsnetzwerken kann zwischen statischer Komplexität und dynamischer Komplexität unterschieden werden (Macchion et al. 2020, S. 2679; Bode & Wagner 2015, S. 216; Serdarasan 2013, S. 533). Die statische Komplexität ergibt sich aus der Anzahl der Partner eines Wertschöpfungsnetzwerks und deren vielzähliger Verbindungen in Form von Waren-, Informations- und Geldflüssen. Die dynamische Komplexität resultiert aus dem Verhalten eines Wert-

⁴engl. Value Chain

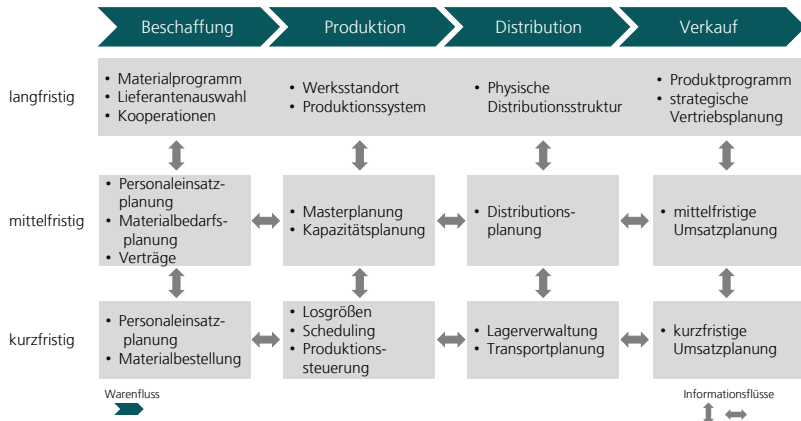


Abbildung 2.5: Planungs- und Entscheidungsaufgaben in Wertschöpfungsnetzwerken (eigene Darstellung in Anlehnung an Fleischmann et al. 2015, S. 77)

schöpfungsnetzwerks und seiner Umgebung, was insbesondere auch die Unsicherheit mit einschließt. Aus der statischen und dynamischen Komplexität folgt auch eine Komplexität der Entscheidungsfindung in Wertschöpfungsnetzwerken.

Die in der Reaktion auf die Komplexität besten Wertschöpfungsnetzwerke zeichnet dabei nicht maßgeblich deren Effizienz aus, sondern die drei Qualitäten Agilität, Ausrichtung von Interessen und Adaptivität (Lee 2004, S. 102–103; Chopra 2018, S. 26). Die Agilität beschreibt eine schnelle Reaktion auf Veränderungen im Wertschöpfungsnetzwerk, beispielsweise hinsichtlich Nachfragemengen und Lieferterminen (Lee 2004, S. 103). Damit kann auch einer Nachfrageverstärkung, dem sogenannten Bullwhip-Effekt, entgegenge wirkt werden (Stadtler 2015, S. 20). Eine Anpassung an sich verändernde Märkte und Gegebenheiten über die Zeit beschreibt die Adaptivität (Lee 2004, S. 103). Dies kann sowohl kurz- als auch langfristig erfolgen und steht damit in Wechselwirkung mit der statischen und dynamischen Komplexität. Wertschöpfungsnetzwerke, die durch ihre Agilität und Adaptivität in der Lage sind, sich anzupassen und, nachdem sie durch äußere Einflüsse verändert wurden, in ihren ursprünglichen Leistungszustand zurückkehren können, werden als resilient bezeichnet - eine wichtige Eigenschaft für Wertschöpfungsnetzwerke produzierender Unternehmen (Schiffer et al. 2020, S. 35; Ivanov & Dolgui 2020, S. 2906; Blackstone 2013, S. 172; Ponomarov & Holcomb 2009, S. 128). Unter Ausrichtung von Interessen

wird verstanden, dass die Kompetenzen der Partner in einem Wertschöpfungsnetzwerk so kombiniert werden, dass jeder Partner durch eine Maximierung der eigenen Interessen gleichzeitig eine Maximierung der Leistung des Wertschöpfungsnetzwerks anstrebt (Lee 2004, S. 102–103). Damit wird vermieden, dass durch parallele Interaktionen lokale Optima an einzelnen Standorten anstatt eines globalen Optimums für das gesamte Wertschöpfungsnetzwerk angestrebt werden (Aelker et al. 2013, S. 82; Friedli & Schuh 2012, S. 187). Wertschöpfungsnetzwerke, die mit inhärenter Agilität, Adaptivität und Ausrichtung von Interessen agieren, bieten den beteiligten Unternehmen einen nachhaltigen Wettbewerbsvorteil (Lee 2004, S. 103).

2.1.3 Ereignisse als Komplexitätstreiber

„When I can't handle events, I let them handle themselves.“

Henry Ford (1863-1947)⁵

In der Stochastik wird ein Ereignis als Teilmenge der Ergebnismenge eines Zufallsexperiments mit einer endlichen Anzahl von möglichen Resultaten beschrieben. Die Chance, dass ein Ereignis eintritt, kann durch seine Wahrscheinlichkeit beschrieben werden (Jacod & Protter 2004, S. 3–4). Allgemeiner definiert ist ein Ereignis nach VDI 3633 (S. 11) eine Begebenheit, die eine Zustandsänderung bewirkt. Ein Ereignis kann deshalb auch als Zustandsänderung in einem System bezeichnet werden, die keine Realzeit verbraucht (VDI 3633, S. 11; Hedtstück 2017, S. 12; DIN SPEC 91329, S. 6; Bruns & Dunkel 2010, S. 48; Fishman 2001, S. 39). Damit kann ein Ereignis eine Aktivität, einen Vorgang oder eine Entscheidung beinhalten (Bruns & Dunkel 2010, S. 48). In der IT ist ein Ereignis ein Objekt, das eine Aufzeichnung von einer Aktivität, einem Vorgang oder einer Entscheidung in einem System darstellt (Luckham 2010, S. 88). Ein Ereignis ist deshalb sowohl ein Vorkommnis in einem System als auch dessen Repräsentation in einem IT-System (Roberts 2018, S. 31; Etzion & Niblett 2011, S. 4). Im Kontext der Repräsentation wird deshalb auch von Instanzen gesprochen, womit Ereignisse mit einem Zeitstempel als Ausprägung einer Klasse von gleichartigen Ereignissen, sogenannten Ereignistypen, bezeichnet wird

⁵zitiert aus Etzion & Niblett 2011, S. 61.

(Hedstüch 2017, S. 13). Ereignisse können durch Zeit, Kausalität und Aggregation mit anderen Ereignissen verbunden sein (Luckham 2010, S. 88).

Jede Aktivität, jeder Vorgang und jede Entscheidung in der realen Welt oder in einem Softwaresystem mit Auswirkungen auf Prozesse in einem Unternehmen stellt ein Ereignis dar (Bruns & Dunkel 2010, S. 13). Unternehmen sind deshalb seit jeher ereignisgetrieben und entsprechend vielfältig sind die Beispiele für Ereignisse (Luckham 2010, S. 29). Als grundlegende Auslöser dynamischer Vorgänge sind Ereignisse deshalb von zentraler Bedeutung für Unternehmen (Günthner 2008, S. 377). Beispiele für Ereignisse sind Telefonanrufe, E-Mails, Finanztransaktionen sowie Anlieferungen oder Auslieferungen von Waren (Etzion & Niblett 2011, S. 6; Bruns & Dunkel 2010, S. 14). Auch eingehende, veränderte oder stornierte Kundenaufträge sind Ereignisse, ebenso wie Lieferverzögerungen bei Lieferanten (Lödding 2016, S. 3; Fleischmann et al. 2015, S. 74). Ereignisse in der Fertigung resultieren unter anderem aus Rückmeldedaten der einzelnen Arbeitsgänge, überschrittenen Grenzwerten von Sensoren sowie Maschinen- oder Personalausfällen (Lödding 2016, S. 3; Fleischmann et al. 2015, S. 74; Luckham 2010, S. 29; Bruns & Dunkel 2010, S. 14). Ereignisse sind zwar oftmals negativ konnotiert, können aber ebenso positiv sein (Etzion & Niblett 2011, S. 6; Heusler et al. 2006, S. 21). In der Literatur werden negative Ereignisse häufig als Störungen, also nicht geplante Unterbrechungen eines Arbeitsablaufes, bezeichnet (VDI 1992, S. 128). Mitarbeiter und Führungskräfte in der Fertigung kümmern sich mit meist großem zeitlichen manuellen Aufwand darum, den Fertigungsablauf trotz dieser Ereignisse sicherzustellen (Lödding 2016, S. 3).

Für die weitere Betrachtung von Ereignissen ist es sinnvoll, diese zu klassifizieren, wofür in der Literatur unterschiedliche Ansätze existieren (vgl. Abschnitt A.1). Gezielt auf produzierende Unternehmen in Wertschöpfungsnetzwerken ausgerichtet, können Ereignisse nach ihrem Ursprung, ihrer Ursache, dem Lebenszyklusabschnitt des Wertschöpfungsnetzwerks, der Vorhersagbarkeit ihres Eintretens und ihres Inhalts, ihrem tatsächlichen Inhalt sowie dem notwendigen Entscheidungshorizont klassifiziert werden (Bauer et al. 2019b, S. 89–90).

Aus dem volatilen, unsicheren, komplexen und mehrdeutigen Umfeld produzierender Unternehmen (vgl. Abschnitt 1.1) resultiert eine Vielzahl von Ereignissen, welche die dynamische Komplexität in Wertschöpfungsnetzwerken erhöhen (Macchion et al. 2020, S. 2679; Bode & Wagner 2015, S. 216; Serdarasan 2013, S. 533). Ereignisse sind deshalb Treiber

der dynamischen Komplexität in Wertschöpfungsnetzwerken und resultieren aus Marktunsicherheiten, Abweichungen zwischen Prognosen und tatsächlichem Verhalten sowie Nachfrageverstärkungen und parallelen Interaktionen (Serdarasan 2013, S. 534). Daraus folgende Prioritätsänderungen von Kundenaufträgen treten täglich auf, oftmals sogar mehrfach am Tag (Pfund et al. 2006, S. 221).

Ereignisse können signifikante finanzielle Auswirkungen auf produzierende Unternehmen und deren Wertschöpfungsnetzwerke haben (Christopher 2011, S. 188; Vakharia & Yenipazarli 2008, S. 246). Ebenso wie Ereignisse positiv oder negativ sein können, können sich auch deren Auswirkungen positiv oder negativ auf Umsatz, Gewinn und Firmenwert bzw. Aktienkurs auswirken. Ereignisse haben jedoch in Form von Mengen-, Termin- oder Prioritätsänderungen insbesondere auch Auswirkungen auf die Produktion (Pfund et al. 2006, S. 222). Folglich sind Prozesse für produzierende Unternehmen notwendig, um auf Ereignisse zu reagieren (Bauer et al. 2021b, S. 388; Bauer et al. 2019b, S. 90). Aufgrund der Komplexität der Entscheidungsfindung sind bei eintretenden Ereignissen meist Algorithmen notwendig, um Handlungen abzuleiten und den weiteren Verlauf zu bestimmen (Schiffer et al. 2020, S. 36–37; Serdarasan 2013, S. 533).

Das Grundprinzip ereignisgetriebener Systeme besteht aus den Schritten Erkennen, Verarbeiten und Reagieren. Zunächst müssen Ereignisse in relevanten Betrachtungsbereichen erkannt und für die weitere Verarbeitung kommuniziert werden. Anschließend werden die Ereignisse analysiert, insbesondere hinsichtlich ihrer Beziehungen und Abhängigkeiten sowie daraus folgender Muster. Darauf basierend kann auf die Ereignisse reagiert werden. Mögliche Reaktionen reichen dabei von einfachen Benachrichtigungen bis hin zum automatisierten Auslösen von Geschäftsprozessen (Bruns & Dunkel 2010, S. 49–50). Das sogenannte Complex Event Processing (CEP) ist eine Sammlung von Technologien und Lösungsansätzen, um Prozesse zu überwachen, Ereignisse zu erkennen und zu analysieren sowie, basierend darauf, den Ablauf dieser Prozesse oder davon abhängiger Prozesse zu beeinflussen (Hedtstück 2017, S. 12; Luckham 2010, S. XV). Aus einer Wissensbasis heraus werden Ereignisse ausgewertet und meist mittels einer sogenannten Rule Engine Schlussfolgerungen angewendet, um eine Entscheidung über die Reaktion auf die Ereignisse herbeizuführen (Hedtstück 2017, S. 4).

2.1.4 IT-Unterstützung für das Management von Wertschöpfungsnetzwerken

Das Supply Chain Management (SCM) hat die Aufgabe, Einheiten in einem Wertschöpfungsnetzwerk zu integrieren und die assoziierten Waren-, Informations- und Geldflüsse (vgl. Abschnitt 2.1.2) unternehmensübergreifend zu optimieren (Stadtler 2015, S. 5). Damit wird eine Steigerung der Wettbewerbsfähigkeit des Wertschöpfungsnetzwerks, der Kundenorientierung, der effektiven Synchronisation von Bedarf und Versorgung, der bedarfsgerechten Produktion sowie dem Abbau von Beständen angestrebt (Stadtler 2015, S. 5; Blackstone 2013, S. 172; Kuhn & Hellgrath 2002, S. 11). SCM umfasst die Gestaltung, Planung, Ausführung, Steuerung und Überwachung aller wertschöpfungsnetzwerkbezogenen Aktivitäten (Blackstone 2013, S. 172). Deshalb ist SCM ein integraler Bestandteil eines Wertschöpfungsnetzwerks und sollte eng mit dessen Strategie verknüpft sein (Serdarasan 2013, S. 533).

Während sich klassische betriebswirtschaftliche Optimierungsansätze zumeist auf intra-organisatorische Maßnahmen konzentrieren, wird im Rahmen des SCM versucht, Waren-, Informations- und Geldflüsse inter-organisatorisch zu planen, zu steuern und zu optimieren (Schönsleben 2016, S. 385; Stadtler 2015, S. 5; Heusler et al. 2006, S. 19). Damit grenzt das SCM konzeptionell an die PPS einzelner Standorte eines Wertschöpfungsnetzwerks an. Praktisch sind SCM und PPS oftmals jedoch nur wenig vernetzt (Lödding 2020, S. 40). Prognosen stellen deshalb die Grundlage für Planungen an den Standorten von Wertschöpfungsnetzwerken dar (Chopra 2018, S. 172). Prognosen sind jedoch nie exakt, sodass die daraus resultierende Abweichung zwischen Prognosen und tatsächlichem Bedarf bei Entscheidungen über Waren-, Informations- und Geldflüsse zu berücksichtigen ist (Chopra 2018, S. 173). Die Genauigkeit von Prognosen nimmt darüber hinaus mit zunehmendem Zeithorizont ab, langfristige Prognosen sind deshalb ungenauer als kurzfristige Prognosen (Chopra 2018, S. 173; Schuh et al. 2013, S. 482). Zur Erstellung von Prognosen in Wertschöpfungsnetzwerken werden qualitative, zeitreihenbasierte, kausale und simulative Methoden genutzt (Chopra 2018, S. 175–188).

Die Qualität und Effizienz von Unternehmen und damit auch deren Wettbewerbsfähigkeit hängt signifikant vom effektiven und effizienten Management von Informationsflüssen

in Wertschöpfungsnetzwerken ab (Bruns & Dunkel 2010, S. 5). Der Austausch von Daten zwischen allen Partnern eines Wertschöpfungsnetzwerks ist hierfür von zentraler Bedeutung (Schiffer et al. 2020, S. 36). Daten müssen in der notwendigen Zeit- und Objektgranularität sowie mit dem richtigen Inhalt am richtigen Ort vorliegen (Reinhart & Gyger 2008, S. 302; Fleisch et al. 2005, S. 16–17). IT-Systemen kommt dabei die Aufgabe zu, Daten und Informationsflüsse zu ermöglichen, zu analysieren und die Entscheidungsfindung zu unterstützen (Chopra 2018, S. VII-2). Der Einsatz von IT-Systemen ist darin begründet, dass die meisten Entscheidungsprobleme aufgrund ihrer Komplexität eine zu große Rechenleistung erfordern, als dass diese manuell vom Menschen gelöst werden können (Schönsleben 2016, S. 393).

Wie in Abbildung 2.6 dargestellt, gliedern sich die Aufgaben der IT-Systeme des SCM, für deren detaillierte Beschreibung auf die Literatur verwiesen wird, in Lieferantenbeziehungsmanagement, Internes Supply Chain Management und Kundenbeziehungsmanagement, unterstützt durch das Transaktionsmanagement (Chopra 2018, S. VII-4). Die Schnittstellen zu den IT-Systemen der PPS bestehen im internen SCM (vgl. Abschnitt 2.2.4).

Lieferanten- beziehungsmanagement	Internes Supply Chain Management	Kunden- beziehungsmanagement
Entwicklungszusammenarbeit	Strategische Planung	Marketing
Sourcing	Bedarfsplanung	Verkauf
Verhandlungsmanagement	Bereitstellungsplanung	Auftragsmanagement
Einkauf	Fulfillment	Kundenservice
Planungszusammenarbeit	Service im Feld	
Transaktionsmanagement		

Abbildung 2.6: IT-Systeme des SCEM (eigene Darstellung in Anlehnung an Chopra 2018, S. VII-8)

Während die IT-Systeme des SCM die Synchronisation der Partner eines Wertschöpfungsnetzwerks auf Planungsebene vornehmen, führt das sogenannte Supply Chain Event Management (SCEM) eine ereignisbasierte Resynchronisierung durch (Otto 2003, S. 5). SCEM kann damit die Lücke zwischen der mittel- bis langfristig ausgerichteten Planung und der kurzfristigen operativen Ausführung in Wertschöpfungsnetzwerken schließen (Heusler

et al. 2006, S. 19). Dabei hat SCEM die Aufgabe, Ereignisse in Wertschöpfungsnetzwerken (vgl. Abschnitt 2.1.3) zu erfassen, zu überwachen und zu bewerten, um basierend darauf Aktionen auszulösen (Blackstone 2013, S. 171–172; Otto 2003, S. 1; Nissen 2002, S. 477). Aktionen können Benachrichtigungen über das Auftreten eines Ereignisses umfassen, ebenso jedoch komplexe Aktionen zur Reaktion auf das Ereignis (Blackstone 2013, S. 171). Das Ziel von SCEM ist es, Abweichungen in Form von Ereignissen in den Waren-, Informations- und Geldflüssen des Wertschöpfungsnetzwerks zu identifizieren und deren negative Konsequenzen zu minimieren, bevor diese die Effizienz und Kundenzufriedenheit im Wertschöpfungsnetzwerk beeinträchtigen (Heusler et al. 2006, S. 19; Otto 2003, S. 3). In der Literatur wird teilweise auch die Vermeidung von Ereignissen im Wertschöpfungsnetzwerk dem SCEM zugerechnet, dies kann jedoch auch eine Planungsaufgabe des SCM darstellen (Baader & Montanus 2008, S. 4; Heusler et al. 2006, S. 19). Nach Knickle & Kemmeter (2002, S. 16) umfassen die Kernaufgaben des SCEM das Überwachen, Melden, Simulieren, Steuern und Messen. Während sich das Überwachen und Melden auf die Erkennung und Benachrichtigung über Ereignisse beziehen, dienen das Simulieren und Steuern der Entscheidungsfindung und deren Umsetzung. Parallel dazu hat das Messen die Aufgabe, durch eine kontinuierliche Zielüberprüfung die Erreichbarkeit des Soll-Zustands im Wertschöpfungsnetzwerk zu gewährleisten. SCEM-Systeme benötigen hierfür einerseits eine Integration mit den beschriebenen SCM-Systemen, andererseits eine Integration mit den IT-Systemen der PPS (vgl. Abschnitt 2.2.4).

2.2 Produktionsplanung und -steuerung

Die Produktionsplanung und -steuerung (PPS) hat die Aufgabe, „das laufende Produktionsprogramm in regelmäßigen Abständen nach Art und Menge für mehrere Planungsperioden im Voraus zu planen und unter Beachtung gegebener oder zu planender Kapazitäten, trotz unvermeidlicher Störungen wie Personalausfall, Lieferverzögerungen oder Ausschuss, möglichst wirtschaftlich zu realisieren“ (Wiendahl & Wiendahl 2019, S. 245). Der Planungsaspekt umfasst dabei das Festlegen von Zielen sowie die Vorbereitung von Produktionsaufgaben und deren Ablaufbestimmung, um diese Ziele zu erreichen. Der Steuerungsaspekt hingegen

bezieht sich auf das Veranlassen, Überwachen und Sicherstellen der Durchführung des geplanten Produktionsablaufs (VDI 1992, S. 167).

Vor dem Hintergrund der Einbettung von global verteilten Produktionsstandorten in ein Wertschöpfungsnetzwerk interagieren die Planung und Steuerung des Wertschöpfungsnetzwerks mit der lokalen PPS eines Produktionsstandorts (vgl. Abschnitt 2.1.2; Schönsleben 2016, S. 385; Schuh et al. 2012b, S. 32). Planungsaufgaben auf Ebene des Wertschöpfungsnetzwerks haben deshalb oftmals ein entsprechendes Pendant in der lokalen PPS. Die Aufgaben der lokalen PPS bauen dabei auf den Ergebnissen der Planung im Wertschöpfungsnetzwerk auf, die als Eingangsgrößen genutzt werden (Schuh et al. 2012b, S. 32–37). Die Verknüpfung zwischen Wertschöpfungsnetzwerk und PPS erfolgt deshalb derzeit über Planungsobjekte wie Produktionsmengen oder Produktionstermine (Schenk et al. 2012, S. 429).

Nachfolgend wird die Rolle der PPS (vgl. Abschnitt 2.2.1) und ihre Ziele (vgl. Abschnitt 2.2.1) näher ausgeführt. Anschließend werden die Besonderheiten der PPS in der komplexen Werkstattfertigung aufgezeigt (vgl. Abschnitt 2.2.3). Es folgt die Darstellung der IT-Unterstützung in der PPS (vgl. Abschnitt 2.2.4).

2.2.1 Die Rolle der Produktionsplanung und -steuerung

In der Produktion wirken Menschen und Produktionsmittel unter Anwendung von Konzepten, Methoden und Werkzeugen zusammen, um den Produktionsablauf effektiv und effizient zu gestalten (vgl. Abschnitt 2.1.1). Dieses Zusammenwirken wird auch als Produktionssystem bezeichnet. Ein wesentlicher Baustein solch eines Produktionssystems ist die PPS, die heute nach wie vor eine Kernkompetenz eines jeden produzierenden Unternehmens bildet (Schuh et al. 2012a, S. 11). Im Hinblick auf die Herausforderungen im Unternehmensumfeld (vgl. Abschnitt 1.1) kann davon ausgegangen werden, dass die Bedeutung einer akkuraten und zuverlässigen PPS zukünftig weiter zunehmen wird (Lödding 2016, S. 3; Reinhart & Gyger 2008, S. 302).

Die Aufgaben der PPS setzen sich aus aufeinander aufbauenden Teilaufgaben zusammen (Wiendahl & Wiendahl 2019, S. 279; Hackstein 1989, S. 16-17). Im deutschen Sprachraum ist dabei das Aachener PPS-Modell nach Schuh et al. (2012a) sehr verbreitet, welches auf

Luczak & Eversheim (1999) zurückgeht. Bei der Beschreibung der Aufgaben unterscheidet das in Abbildung 2.7 dargestellte erweiterte Aachener PPS-Modell die drei Kategorien der überbetrieblichen Netzwerkaufgaben, der Kernaufgaben und der Querschnittsaufgaben (Schuh et al. 2012a, S. 21). Die überbetrieblichen Netzwerkaufgaben bilden dabei die Integration zwischen der Planung im Wertschöpfungsnetzwerk und der Planung an einzelnen Produktionsstandorten ab (vgl. Abschnitt 2.1.2). Die Kernaufgaben fokussieren die PPS für einen einzelnen Produktionsstandort. Dabei werden sämtliche Aufgaben abgedeckt, die mit dem eigentlichen Produkterstellungsprozess in Zusammenhang stehen. Querschnittsaufgaben wiederum zielen auf eine Unterstützung, Integration und damit Optimierung der Netzwerk- und Kernaufgaben ab. Über alle Kategorien erstreckt sich die Aufgabe der Datenverwaltung, da diese Voraussetzung für die gesamte PPS ist (Schuh et al. 2012a, S. 20–21).

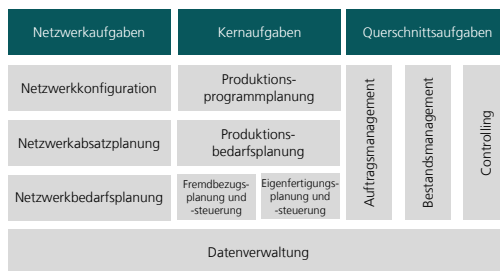


Abbildung 2.7: Aufgaben nach dem Aachener PPS-Modell (eigene Darstellung in Anlehnung an Schuh et al. 2012a, S. 21)

Unter den Kernaufgaben findet sich die für den operativen Betrieb der Fertigung zentrale Aufgabe der Eigenfertigungsplanung und -steuerung. Diese detailliert die Planvorgaben der vorgelagerten Aufgaben im Rahmen des zur Verfügung stehenden Dispositionsspielraums aus und steuert deren Umsetzung (Schuh et al. 2012b, S. 51–52). Hierfür sind die Teilaufgaben der Losgrößenrechnung, Feinterminierung, Ressourcenfeinplanung, Ressourcenbelegungsplanung, Reihenfolgeplanung, Verfügbarkeitsprüfung und Auftragsfreigabe notwendig (Schuh et al. 2012b, S. 53–57). Ergänzt werden diese Teilaufgaben um die Auftragsüberwachung (VDI 1992, S. 167; Hackstein 1989, S. 16–17). Diese beinhaltet die Erfassung und Verwaltung von Zustandsänderungen der Aufträge und Ressourcen während der Fertigung. So wird ein stets aktueller Überblick über den Auftragsstand sowie die

Belastungssituation der Ressourcen angestrebt. Dies ist die Basis für Korrekturmaßnahmen zur Sicherstellung der Durchführung des Produktionsablaufs.

In der PPS wird von den beiden Teilgebieten der Planung und Steuerung ausgegangen, die aus Aufgabensicht darauf aufbauen, welche Tätigkeiten einen planenden Charakter haben und welche einen steuernden. Die Auftragsfreigabe wird dabei häufig als Übergang zwischen Planung und Steuerung gesehen (Lödding 2016, S. 8; Wiendahl & Wiendahl 2019, S. 284). Der Übergang zwischen Planung und Steuerung ist jedoch nicht immer eindeutig festlegbar, da, abhängig vom Fertigungsprinzip und den organisatorischen Rahmenbedingungen, einzelne Planungsaufgaben auch Steuerungsaufgaben sein können und vice versa (Zäpfel 2001, S. 60). Grundsätzlich besteht die Unterscheidung zwischen Planung und Steuerung jedoch dort, wo Planvorgaben in Durchsetzungsaktivitäten übergehen (Corsten & Gössinger 2016, S. 586; Zäpfel 2001, S. 60).

Aus Ablaufsicht baut die Unterscheidung zwischen Planung und Steuerung auf den zeitlichen Horizont auf, wann Gestaltungsentscheidungen getroffen werden und wann diese umgesetzt werden. Bei der Produktionsplanung werden die Produktionsprozesse auf der dispositiven Ebene gestaltet, bevor sie ausgeführt werden, was einen mittel- bis langfristigen Zeithorizont umfasst. Die Produktionsplanung arbeitet meist mit abstrahierten Modellen der Produktion, die Produktionsabläufe typischerweise mit mittelwertbasierten Kennzahlen beschreiben. In der Produktionsplanung kann es für Detailspekte der Produktion vor Produktionsbeginn unbekannt sein, welche Ausprägungen diese in Abhängigkeit der Planungsentscheidungen annehmen. Die Produktionssteuerung hingegen zeichnet sich dadurch aus, dass die Produktionsprozesse auf der dispositiven Ebene während ihrer Ausführung gestaltet werden. Es werden abgegrenzte Produktionsbereiche in einem kurzfristigen Zeithorizont betrachtet. Im Fokus steht eine mikroskopische Sichtweise auf einzelne Aufträge, Arbeitsgänge und somit eine Vielzahl von Einzelereignissen (Corsten & Gössinger 2016, S. 586–587; Westkämper et al. 2000, S. 204).

Neuere Ansätze der PPS, welche die zunehmende Komplexität der Fertigung berücksichtigen, gehen davon aus, dass Planung und Steuerung zukünftig noch weiter verschmelzen werden und diese Unterscheidung dadurch weniger wichtig wird. Einerseits werden zukünftig durch die Steuerung Entscheidungen in Echtzeit getroffen, andererseits werden

deren Auswirkungen vorhergesagt und zur Optimierung der Planung genutzt (Siegert et al. 2018, S. 181).

Unabhängig von der Abgrenzung zur Planung ist die Aufgabe der Produktionssteuerung, in der Fertigung konkret der Fertigungssteuerung, die Realisierung des geplanten Fertigungsprogramms trotz unvermeidbarer und unvorhergesehener Ereignisse und Störungen. Hierfür sind die Fertigungsprozesse zu überwachen und bei Abweichungen zwischen Plan und tatsächlichem Zustand Korrekturmaßnahmen vorzunehmen (Wiendahl & Wiendahl 2019, S. 245; VDI 1992, S. 138). Lödding (2016, S. 8–10) beschreibt dieses Vorgehen in seinem Modell der Fertigungssteuerung. Durchgesetzt und beeinflusst wird das Fertigungsprogramm in diesem Modell über die Auftragsfreigabe, die Kapazitätssteuerung und die Reihenfolgebildung. Demgegenüber stehen die Planvorgaben der Auftragserzeugung, welche nach dem beschriebenen Aachener PPS-Modell im Rahmen der Eigenfertigungsplanung ermittelt werden. Das Modell der Fertigungssteuerung stellt damit eine Verbindung zwischen den Aufgaben der Fertigungssteuerung und den Zielgrößen der PPS (vgl. Abschnitt 2.2.2) her (Lödding 2016, S. 7). Eine weitergehende Übersicht von Verfahren zur Umsetzung der Aufgaben in der Fertigungssteuerung ist der Literatur zu entnehmen (Lödding 2016; Schuh & Schmidt 2014; Kienzle 2012).

2.2.2 Ziele und Messgrößen der Produktionsplanung und -steuerung

Die Leistung und damit die Ziele der PPS werden durch kurze Durchlaufzeiten und die Einhaltung von Lieferterminen ausgedrückt. Gleichzeitig wird durch geringe Bestände in Lager und Produktion sowie eine hohe Kapazitätsauslastung eine Minimierung der Kosten angestrebt (Wiendahl & Wiendahl 2019, S. 247). Zusätzlich zielt die PPS auf eine hohe Flexibilität der Produktion (Schuh et al. 2012b, S. 29). Zur Erreichung dieser logistischen und wirtschaftlichen Zielsetzungen ist es notwendig, dass auch unvermeidbare Störungen im Ablauf beherrscht werden (Wiendahl & Wiendahl 2019, S. 279). Dabei beeinflusst die PPS mit ihren Entscheidungen das Produktionssystem und wird gleichzeitig durch dieses wiederum selbst beeinflusst (Uzsoy et al. 1994, S. 51). Die Zielerreichung der PPS in einer Fertigung kann über die nachfolgenden logistischen Zielgrößen Liefertermineinhaltung, Durchlaufzeit und Bestand gemessen werden (Lödding 2016, S. 8). Gleichzeitig können Soll-Werte

dieser Größen auch Steuergrößen für eine Fertigung darstellen. Weitere, untergeordnete Kennzahlen sind Mönch et al. (2013, S. 55–61) zu entnehmen.

Zur Berechnung der Liefertermineinhaltung ist im ersten Schritt die Bestimmung der Lieferterminabweichung LTA notwendig, welche sich aus der Differenz zwischen dem Ist-Liefertermin TL_{Ist} und dem Plan-Liefertermin TL_{Plan} ergibt (vgl. Gleichung 2.1). Nach dieser Definition beschreibt eine positive Lieferterminabweichung eine verspätete Lieferung, eine negative Lieferterminabweichung eine verfrühte Lieferung (Lödding 2016, S. 27).

$$LTA = TL_{Ist} - TL_{Plan} \quad (2.1)$$

Daraus lässt sich anhand Gleichung 2.2 die Liefertermineinhaltung LTE , der prozentuale Anteil pünktlich oder verfrüht gelieferter Aufträge n_A , berechnen (Lödding 2016, S. 27–28).

$$LTE = \frac{n_A \text{ mit } LTA \leq 0}{n_A} \cdot 100 \quad (2.2)$$

Die Liefertermineinhaltung ist eine Schlüsselkenngröße für die Kundenzufriedenheit, da sie einen direkten Einfluss auf das Vertrauensverhältnis zum Kunden hat. Bei einer schlechten Liefertermineinhaltung entstehen deshalb auch Folgekosten für eine gegebenenfalls dadurch ebenfalls schlechtere Liefertermineinhaltung des Kunden sowie durch Sicherheitsbestände, welche dadurch notwendig werden (Lödding 2016, S. 29; Mönch et al. 2011, S. 583). Gegenüber anderen Fertigungsprinzipien ist die Liefertermineinhaltung in der Werkstattfertigung tendenziell schlechter (Niehues 2017, S. 231).

Die Durchlaufzeit eines Auftrags ZDA berechnet sich aus der Differenz zwischen dem Bearbeitungsbeginn TAB und Bearbeitungsende TAE eines Auftrags (vgl. Gleichung 2.3). Dieser Zeitraum reicht von der Auftragsfreigabe bis zum Versand an den Kunden (Lödding 2016, S. 32; Mönch et al. 2013, S. 12). Neben der Durchlaufzeit für den Auftrag kann die Durchlaufzeit auch für einzelne Arbeitsvorgänge als ZDL angegeben werden. Diese untergliedert sich dann noch weiter in die Komponenten der Übergangszeit ZUE und der Durchführungszeit ZDF (vgl. Abbildung 2.8). Tendenziell hat die Werkstattfertigung gegenüber anderen Fertigungsprinzipien, insbesondere der Fließfertigung, die höchste

Durchlaufzeit (Niehues 2017, S. 232).

$$ZDA = TAE - TAB \quad (2.3)$$

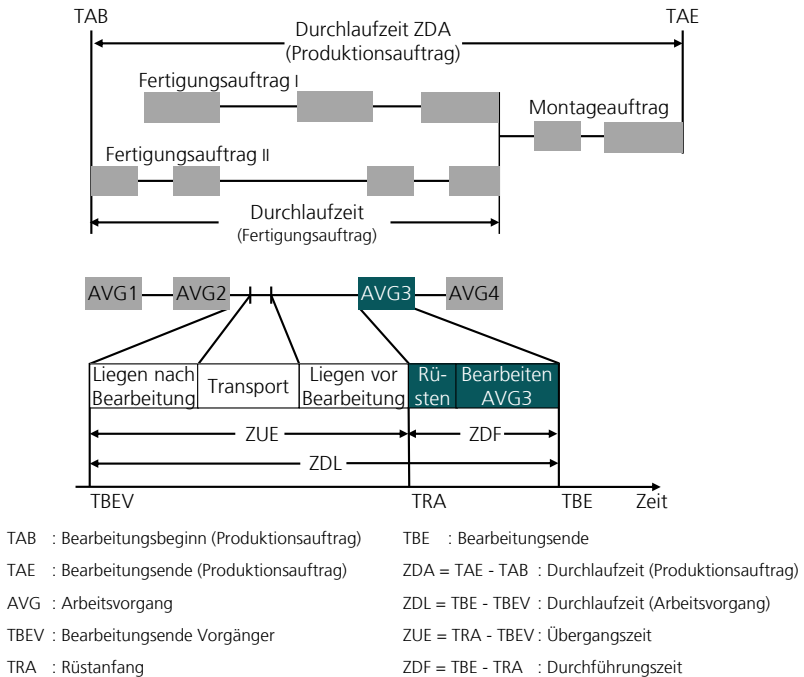


Abbildung 2.8: Komponenten der Durchlaufzeit (eigene Darstellung in Anlehnung an Löffding 2016, S. 32; Nyhuis & Wiendahl 2012, S. 22)

Als relatives Maß für die Durchlaufzeit bietet sich der Flussfaktor FF^6 an, welcher sich analog zu Gleichung 2.4 aus der Durchlaufzeit und der Summe der Durchführungszeiten ZDF , auch als Rohprozesszeit bezeichnet, errechnet (Nyhuis & Wiendahl 2012, S. 23). Die Durchlaufzeit kann dabei auf einen Auftrag (ZDA) oder einen Arbeitsgang (ZDL) sowie als Mittelwert auf die Fabrik bezogen werden (Waschneck 2020, S. 33; Mönch et al. 2013, S. 92; Stubbe 2010, S. 27). Durch seine normalisierende Wirkung erlaubt der

⁶engl. Flow Factor oder X-Factor

Flussfaktor auch einen Leistungsvergleich über Arbeitspläne und Fabriken hinweg (Stubbe 2010, S. 27).

$$FF = \frac{ZDA}{ZDF} \quad (2.4)$$

Der Bestand⁷ einer Fertigung drückt sich über die Differenz von Zugängen und Abgängen in einem Bezugszeitraum aus (Lödding 2016, S. 9). Berechnet wird der Bestand B_m nach Little's Gesetz (Little 1961, S. 383)⁸ über die Multiplikation von mittlerer Durchlaufzeit ZDA_m und mittlerer Leistung L_m ⁹ (vgl. Gleichung 2.3; Lödding 2016, S. 141).

$$B_m = ZDA_m \cdot L_m \quad (2.5)$$

Bei gleichbleibender Leistung steigt deshalb mit zunehmender Durchlaufzeit auch der Bestand in der Fertigung an (Little 1961, S. 383). Oder anders formuliert, der Bestand einer Fertigung beeinflusst sowohl die Auslastung und damit die Leistung als auch die Durchlaufzeiten der Aufträge. Mit steigendem Bestand nehmen deshalb beide Werte zu (Lödding 2016, S. 37). Dieser Zielkonflikt zwischen niedrigen Beständen, kurzen Durchlaufzeiten und einer hohen Auslastung wird auch als Dilemma der Ablaufplanung bezeichnet (Lödding 2016, S. 37). Zusätzlich ist zum Bestand noch anzumerken, dass dadurch Kapital im Umlaufvermögen eines Unternehmens gebunden und der finanzielle Spielraum für Investitionen so verringert wird (Lödding 2016, S. 36). Dargestellt werden können die Zusammenhänge von Bestand, Leistung und Durchlaufzeit im Durchlaufdiagramm eines Produktionssystems (vgl. Abbildung 2.9).

⁷engl. Work in Progress

⁸engl. Little's Law

⁹In der deutschsprachigen Produktionsliteratur wird der Durchsatz häufig durch die Leistung ausgedrückt, welche sich aus dem Abgang dividiert durch die Länge eines Bezugszeitraums ergibt (Lödding 2016, S. 39).

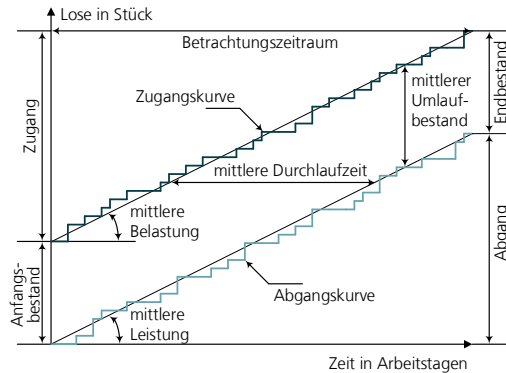


Abbildung 2.9: Durchlaufdiagramm eines Produktionssystems (eigene Darstellung in Anlehnung an Wiendahl & Wiendahl 2019, S. 254; Nyhuis & Wiendahl 2012, S. 22; Wiendahl 2012, S. 113)

2.2.3 Produktionsplanung und -steuerung in der komplexen Werkstattfertigung

Die komplexe Werkstattfertigung gilt als eine der herausforderndsten Produktionsumgebungen, bietet teils extreme Bedingungen und stellt enorme Herausforderungen an die PPS (Chien et al. 2008, S. 2097). Eine Vielzahl von Problemen in der PPS der komplexen Werkstattfertigung ist folglich nicht exakt lösbar (Eiselt & Sandblom 2004, S. 350). Neben mathematisch exakten Methoden werden deshalb insbesondere Heuristiken genutzt, um zukünftige Zustände und Produktionspläne vorherzusagen (Schuh et al. 2013, S. 478).

Aufgrund der langen Durchlaufzeiten sind Prognosen mit kurz-, mittel- und langfristigem Zeithorizont deshalb unbedingt notwendig (Mönch et al. 2013, S. 265). Die Genauigkeit nimmt jedoch mit zunehmendem Zeithorizont ab, bei drei Tagen in der Zukunft liegt diese nach Schuh et al. (2013, S. 482) nur noch bei etwa 25 Prozent. Spezifisch für die komplexe Werkstattfertigung in der Halbleiterindustrie geben Hintz et al. (2011, S. 42–43) an, dass die Genauigkeit der Prognosen bei den neun größten Halbleiterherstellern weltweit um 30 bis 90 Prozent variiert. Es ist deshalb von großer Bedeutung, dass die PPS in der komplexen Werkstattfertigung auch bei Ereignissen und Planabweichungen funktioniert (Wiendahl & Wiendahl 2019, S. 279). Dies umfasst neben internen Ereignissen (vgl. Ab-

schnitt 2.1, z. B. unvorhersehbare Maschinenausfälle) auch externe Ereignisse aus dem Wertschöpfungsnetzwerk (vgl. Abschnitt 2.1.3, z. B. Abweichungen von Prognosen).

Die Ziele der PPS in der komplexen Werkstattfertigung sind die Minimierung der Durchlaufzeit und eine Maximierung der Liefertermineinhaltung (Mönch et al. 2011, S. 583). Das zielt auch darauf ab, Nachfolgaufträge zu generieren und damit eine Kundenabwanderung zum Wettbewerb zu vermeiden (Mason et al. 2002, S. 248). Die Leistungsziele gleichen damit der allgemeinen PPS (vgl. Abschnitt 2.2.2). Zusätzlich ist aber die hohe Kapazitätsauslastung von besonderer Wichtigkeit, da branchenspezifisch durch die komplexe Werkstattfertigung ein noch höherer Kostenanteil verursacht wird als dies durchschnittlich in der Fertigung der Fall ist (Mönch et al. 2011, S. 586; Eversheim 1989, S. 5). Dem Flussfaktor kommt in der komplexen Werkstattfertigung eine besondere Bedeutung zu, da dieser einen Planwert, eine Steuergröße und eine Messgröße darstellt (Waschneck 2020, S. 30–31; Mönch et al. 2013, S. 92; Nyhuis & Wiendahl 2012, S. 23). Die Planungs- und Steuerungsstrategie hat dabei einen großen Einfluss auf die Zielerreichung (Reinhart & Gyger 2008, S. 302).

Abbildung 2.10 zeigt eine hierarchische Darstellung der Aufgaben in der PPS der komplexen Werkstattfertigung. Die *Planung* kann mit verschiedenen Zeithorizonten ausgeführt werden und zielt auf die Antizipation des zukünftigen Bedarfs ab. Der Planungshorizont, der je nach Branche sehr verschieden sein kann, wird dabei in Zeitabschnitte unterteilt, auf die sich die Planungsentscheidungen beziehen. Die Planung selbst wird rollierend ausgeführt, gegenüber der letzten Planung wird also die Planung um eine Periode in die Zukunft fortgeschrieben. Neben dieser zeitgetriebenen Planung ist auch eine ereignisgetriebene Planung möglich, bei der Planänderungen über vordefinierte Ereignisse ausgelöst werden. Des Weiteren sind hybride Formen der Planung möglich, welche sowohl zeitgetriebene als auch ereignisgetriebene Elemente enthalten (Mönch et al. 2013, S. 27). Ergebnis der Planung ist, unabhängig von ihrer Form, eine Zuordnung von Mengen zu Zeitabschnitten und Werken, welche in der Regel einige Monate in die Zukunft reicht (Mönch et al. 2013, S. 26–7; Schuh et al. 2012b, S. 39). Die *Auftragsfreigabe* startet die Bearbeitung der Lose in der Fertigung. Wie in der allgemeinen PPS wird an dieser Stelle auch in der komplexen Werkstattfertigung der Übergang von der Planung zur Steuerung gesehen (Mönch et al. 2013, S. 27). Das sogenannte *Scheduling* ist der Prozess zur Zuweisung von knappen

Ressourcen der Fertigung zu Aufträgen. Aufgrund der Komplexität ist dies derzeit nur für eine Schicht oder einen Tag möglich, nicht jedoch für längere Zeithorizonte im Voraus (vgl. Abschnitt 2.2.4). Wird eine Ressource frei, entscheidet das sogenannte *Dispatching* über den nächsten Auftrag, welcher auf dieser Ressource bearbeitet wird. Gegenüber dem Scheduling ist das Dispatching deshalb eine Just-in-Time-Aufgabe (Mönch et al. 2013, S. 27; Waschneck et al. 2016, S. 2–3).

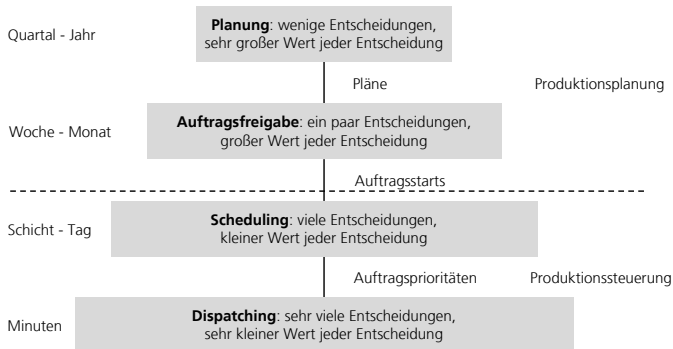


Abbildung 2.10: Hierarchie der PPS-Aufgaben in der komplexen Werkstattfertigung (eigene Darstellung in Anlehnung an Mönch et al. 2013, S. 28)

Die Aufgaben der PPS in der komplexen Werkstattfertigung lassen sich, wie in Tabelle 2.1 dargestellt, auch den Aufgaben der allgemeinen PPS (vgl. Abschnitt 2.2.1) zuordnen. Scheduling und Dispatching sind zwar entsprechenden Aufgaben zugeordnet, in der PPS der komplexen Werkstattfertigung werden diese beiden Aufgaben jedoch wesentlich stärker als Steuerungsaufgaben verstanden als dies in der allgemeinen PPS der Fall ist. Dies ist auch daran zu erkennen, dass Scheduling und Dispatching in der komplexen Werkstattfertigung zeitlich erst nach der Auftragsfreigabe angesiedelt sind. Nicht enthalten in der Tabelle ist die Losgrößenrechnung, da in der komplexen Werkstattfertigung, abgesehen von Prototypen, üblicherweise mit einer konstanten Losgröße gearbeitet wird (vgl. Abschnitt 2.1.1).

Tabelle 2.1: Zuordnung der Aufgaben zwischen komplexer Werkstattfertigung und allgemeiner PPS

Komplexe Werkstattfertigung PPS Mönch et al. (2013)	Allgemeine PPS Schuh et al. (2012a)
Planung	Kombination aus Produktionsprogrammplanung und Produktionsbedarfsplanung
Auftragsfreigabe	Kombination aus Verfügbarkeitsprüfung und Auftragsfreigabe
Scheduling	Ressourcenbelegungsplanung (teilweise)
Dispatching	Reihenfolgeplanung (teilweise)

2.2.4 IT-Unterstützung in der Produktionsplanung und -steuerung

Methoden und Algorithmen der PPS bedingen IT-Systeme, da die meisten Entscheidungsprobleme aufgrund ihrer Komplexität eine zu große Rechenleistung erfordern, als dass diese manuell vom Menschen gelöst werden können (Brambring 2017, S. 40; Schönsleben 2016, S. 393; Mönch et al. 2013, S. 247). Abbildung 2.11 zeigt übliche IT-Systeme bzw. deren Konzepte in der PPS der komplexen Werkstattfertigung, welche in Abschnitt A.3 detailliert dargestellt werden.

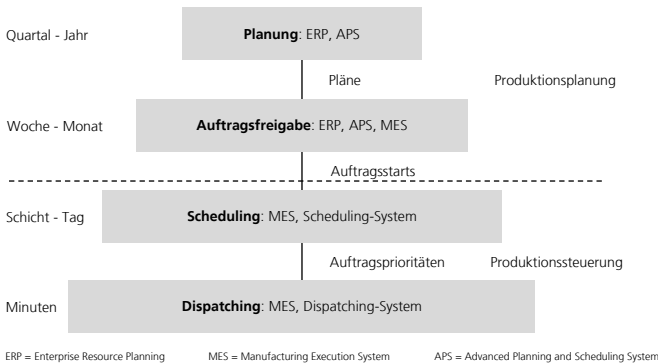


Abbildung 2.11: IT-Systeme der PPS in der komplexen Werkstattfertigung (eigene Darstellung in Anlehnung an Mönch et al. 2013, S. 28, 258)

Ein effektives Datenmanagement ist, analog zu IT-Systemen für das Management von Wertschöpfungsnetzwerken (vgl. Abschnitt 2.1.4), auch für funktionierende IT-Systeme der PPS eine Grundvoraussetzung. Dieses Datenmanagement umfasst die Erfassung, Analyse und Aufbereitung aller für die Planung und Steuerung der Produktion relevanten Daten (Schuh et al. 2012a, S. 20–21; Zäpfel 2001, S. 62). Konkret sind dies Stamm- und Bewegungsdaten (vgl. Abschnitt A.2). Für die Erfassung von Bewegungsdaten werden die folgenden beiden Arten unterschieden, welche IT-seitig in der Regel als Funktionen des Manufacturing Execution System (MES) abgebildet werden (Kletti 2015, S. 21). Die Betriebsdatenerfassung (BDE) umfasst alle Maßnahmen, die zur Rückmeldung von Betriebsdaten aus der Fertigung und deren Bereitstellung in verarbeitungsfähiger Form dienen. Damit wird die Ist-Situation aus der Fertigung an den Ort der Verarbeitung, die Produktionssteuerung, rückgekoppelt (Hansmann 2014, S. 375; VDI 1992, S. 125). Als Pendant zur BDE gilt die Maschinendatenerfassung (MDE), die auf eine automatische Erfassung von Daten über Produktionsanlagen abzielt. Damit werden Zustandsdaten von Ressourcen sowie deren verarbeitete Mengen über Zähler, Waagen und vergleichbare Einrichtungen erfasst (Kletti 2015, S. 20–21).

Durch eine lückenhafte oder zeitversetzte BDE und MDE, bedingt durch eine manuelle Datenerfassung, entsteht heute oftmals eine mangelhafte Integration der realen und digitalen Welt (Lödding 2020, S. 15–16; Fleisch et al. 2005, S. 3–4). Fehlerhafte Rückmeldedaten und nicht gepflegte Stammdaten sind dabei die Hauptursachen für falsche Annahmen in der Produktionsplanung. Darüber hinaus werden Daten aus den Wertschöpfungsnetzwerken kaum genutzt, um die Aufgaben der PPS zu unterstützen (Lödding 2020, S. 40; Nyhuis et al. 2016, S. 15–20). Derzeitige Ansätze zur Optimierung der Fertigung zielen deshalb darauf ab, dass nicht mit einem zeitversetzten Datenabbild der Realität geplant wird, was bisher ein großes Problem darstellt, sondern mit Echtzeiteinformationen gearbeitet wird. Erreicht wird dies über eine De-Hierarchisierung der PPS hin zu einem serviceorientierten Netz. In diesem Netz wird eine vertikale und horizontale Integration entlang des Wertschöpfungsnetzwerks erreicht, indem Softwareservices und physikalische Objekte über standardisierte Schnittstellen kommunizieren. Zusätzlich zu den Daten aus der Fertigung werden so auch Daten aus dem Wertschöpfungsnetzwerk ohne Zeitversatz in die PPS integriert (Bauernhansl 2017a, S. 22–26).

Eine Referenzarchitektur zur Einordnung der IT- und Datenerfassungssysteme in eine ereignisbasierte PPS zeigt Abbildung 2.12. Die Referenzarchitektur beinhaltet die verschiedenen interagierenden Systeme sowie die unterschiedlichen Ebenen der Automatisierungspyramide. Die Verknüpfung zwischen den IT- und Datenerfassungssystemen ist durch Ereignisse (vgl. Abschnitt 2.1.3) realisiert (DIN SPEC 91329, S. 7).

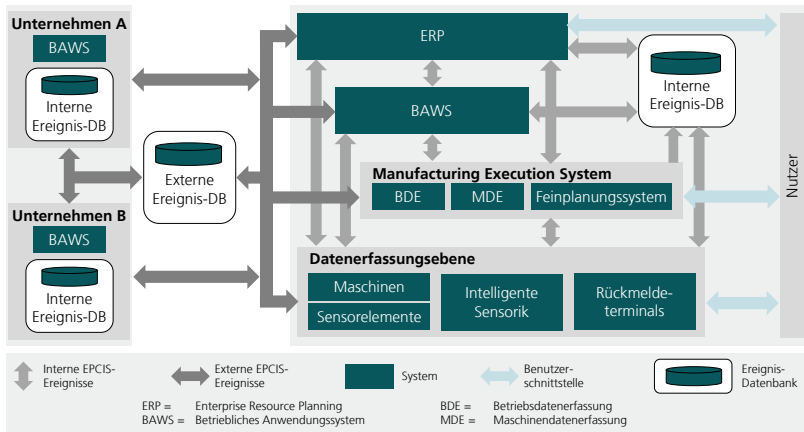


Abbildung 2.12: Referenzarchitektur für eine ereignisbasierte PPS (eigene Darstellung in Anlehnung an DIN SPEC 91329, S. 7)

2.3 Lernende Systeme

Lernen schafft das Potenzial zu handeln. Es legt also die Grundlage für Fähigkeiten und damit für Leistungsfähigkeit (Gross 2015, S. 175). Ebenso wie dies für Menschen gilt, ist es auch für technische Systeme zutreffend. Die Lernfähigkeit von Systemen ist deshalb eine Schlüsseleigenschaft für die Produktion der Zukunft (Westkämper & Löffler 2016, S. 189).

Nachfolgend wird die Charakteristik lernender Systeme näher ausgeführt (vgl. Abschnitt 2.3.1). Anschließend werden Ansätze für die Lernfähigkeit von technischen Systemen (vgl. Abschnitt 2.3.2) und Regelkreise als Handlungsmöglichkeit lernender Systeme (vgl. Abschnitt 2.3.3) erläutert. Es folgt die Darstellung von Regelkreisen in der PPS (vgl. Abschnitt 2.3.4).

2.3.1 Die Charakteristik lernender Systeme

In der Psychologie bezeichnet Lernen eine meist dauerhafte Änderung des Verhaltens als Ergebnis von Übung oder Erfahrung (Lachman 1997, S. 477). Damit ermöglicht Lernen auch eine Anpassung an veränderte Rahmenbedingungen (Simon 1996, S. 100). Lernen kann jedoch auch definiert werden als eine meist dauerhafte Änderung des Verhaltenspotenzials (Anderson 1995). Damit kann das Lernen (Verhaltenspotenzial) von der Leistung (tatsächliches Verhalten) abgegrenzt werden, wobei das Lernen eine wichtige Grundlage für die Leistung ist (Gross 2015, S. 175). Lernende sind dabei aktive Agenten ihres eigenen Lernens, die Informationen auswählen und ihre eigene Bedeutung aus diesen ausgewählten Informationen konstruieren (Anderson & Krathwohl 2001, S. 38). Ein solcher Agent, human oder künstlich, kann dadurch lernen, dass er seine Leistung für zukünftige Aufgaben basierend auf Beobachtungen von Informationen verbessert (Russell & Norvig 2021, S. 651).

Lernen in der Fertigung kann nach Westkämper & Löffler (2016, S. 186) durch die Stimulus-Reaktionstheorie und die kognitive Lerntheorie erklärt werden (vgl. Abbildung 2.13). Die zur Gruppe des konditionierten Lernens gehörende Stimulus-Reaktionstheorie besagt, dass ein lernfähiges Individuum durch Änderung seiner direkten und indirekten Handlungsweisen auf Einflussgrößen reagiert. Wirkungen von Handlungsweisen werden dabei durch Vergleich mit Zielen und Erwartungen unter Berücksichtigung von Wissen aus der Vergangenheit analysiert. Diese Form des Lernens ist typisch bei unbekannten Objekten und Situationen (Westkämper & Löffler 2016, S. 185–186). Die kognitive Lerntheorie beschreibt die Umstrukturierung von Erfahrungen und Erlebnissen des bisher vorhandenen kognitiven Schemas aufgrund von abstrahierten Beobachtungen und daraus gebildeten Modellen. Diese Modelle lassen sich auf vergleichbare und geänderte Aufgaben anwenden (Westkämper & Löffler 2016, S. 186). Beide Lerntheorien unterscheiden sich maßgeblich dadurch, dass der Mensch nach der Stimulus-Reaktionstheorie Handlungen und Handlungsfolgen lernt, während er nach der kognitiven Lerntheorie die Ergebnisse von Handlungen und Handlungsfolgen zu antizipieren lernt (Westkämper & Löffler 2016, S. 186).

Die Lernfähigkeit technischer Systeme ist eine spezifische Form des kognitiven Lernens, die Wissen aus der Beobachtung des Zustands des technischen Systems generiert. Das Lernen geschieht dabei in einem Regelkreis mittels eines iterativen Prozesses mit einem Zugewinn an Wissen und Erfahrung (Westkämper & Löffler 2016, S. 188–189). Die ei-

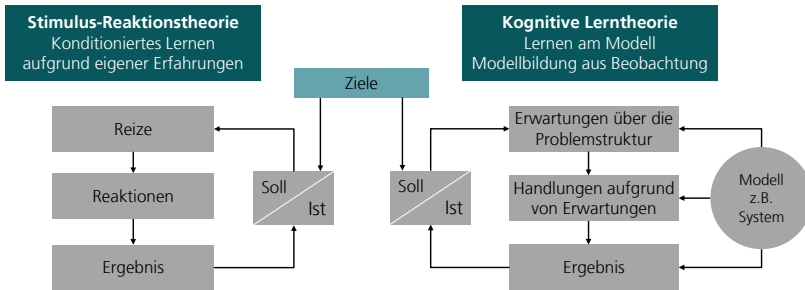


Abbildung 2.13: Theorien des Lernens in der Fertigung (eigene Darstellung in Anlehnung an Westkämper & Löffler 2016, S. 186)

gentliche Lernfähigkeit und die Handlungsfähigkeit in Form von Regelkreisen sind deshalb Schlüsseleigenschaften autonomer technischer Systeme und werden in Abschnitt 2.3.2 und Abschnitt 2.3.3 näher ausgeführt (Styr et al. 2020, S. 354; Groche et al. 2018, S. 129).

2.3.2 Maschinelles Lernen zum Aufbau lernender Systeme

Seit Jahrtausenden versucht die Menschheit Intelligenz, also die Denkweise von Menschen, zu verstehen. Die Entwicklungen der Künstlichen Intelligenz (KI)¹⁰ gehen weiter und versuchen, Intelligenz nicht nur zu verstehen, sondern intelligente Artefakte aufzubauen (Russell & Norvig 2021, S. 1). Die Arbeiten zu KI begannen in den 1950er Jahren, insbesondere Alan Turing war dafür wegweisend (Russell & Norvig 2021, S. 17–18). Er entwickelte den Turing-Test, um zu überprüfen, ob eine Maschine Intelligenz aufweist (Turing 1950, S. 433–460). Ein früher Ansatz von KI basiert auf der Beschreibung von Wissen in formalen Sprachen. Ein Computer kann dann basierend auf logischen Inferenzregeln automatisch Schlüsse ziehen (Goodfellow et al. 2016, S. 2). Das Scheitern von fest programmierten Wissensbasen und Regeln bei der Anwendung auf komplexe Problemstellungen lässt jedoch vermuten, dass eine KI die Fähigkeit des Lernens benötigt (Russell & Norvig 2021, S. 651; Goodfellow et al. 2016, S. 2).

Die Fähigkeit, Wissen aus Daten zu erlernen, indem Algorithmen aus den Daten ein (komplexes) Modell ableiten, wird als Maschinelles Lernen (ML) bezeichnet und ist eine

¹⁰engl. Artificial Intelligence. Eine Übersicht zur Historie von KI ist der Literatur zu entnehmen (Russell & Norvig 2021; Cardon et al. 2018).

Schlüsseltechnologie für KI in realen Problemstellungen (Döbel et al. 2018, S. 8; Goodfellow et al. 2016, S. 2). Lernen bedeutet dabei auch, dass das Modell seine Leistung für zukünftige Aufgaben basierend auf Daten über die Umwelt verbessert (Russell & Norvig 2021, S. 651). Das automatisierte Lernen von Wissen aus Daten ist dabei der wichtigste Unterschied gegenüber früheren Verfahren mit fest programmierten Wissensbasen und Regeln. Von Heuristiken und Verfahren wie der linearen Optimierung unterscheidet sich ML durch die Fähigkeit zur Generalisierung auf unbekannte Daten (Le Roux et al. 2012, S. 404). Notwendig ist ML, da Entwickler nicht alle potenziell möglichen Situationen und deren Änderungen über die Zeit antizipieren können. Des Weiteren ist es bei komplexen Problemstellungen für Menschen nicht immer möglich zu wissen, wie die korrekte Lösung aussehen muss (Russell & Norvig 2021, S. 651). In komplexen Werkstattfertigungen und deren Wertschöpfungsnetzwerken ist dies maßgeblich durch deren statische und dynamische Komplexität begründet (vgl. Abschnitt 2.1.1; Abschnitt 2.1.2).

Ansätze des ML können nach der Form des Lernfeedbacks in unüberwachtes, überwachtes und bestärkendes Lernen gegliedert werden (vgl. Tabelle 2.2). *Unüberwachtes Lernen* bedeutet, dass den Eingangsdaten¹¹ keine Soll-Werte, sogenannte Labels, über Trainingsdaten zugewiesen werden. Es existiert also kein explizites Feedback aus den Daten. Durch das unüberwachte Lernen sollen Regelmäßigkeiten in den Eingangsdaten identifiziert werden. Die häufigste Form des unüberwachten Lernens ist die Clusteranalyse, also die Erkennung zusammenhängender Gruppen in den Eingangsdaten. Eine weitere Anwendung ist die Analyse der Datenverteilung im Raum der Eingangsdaten (Russell & Norvig 2021, S. 653; Alpaydin 2020, S. 11; Richter 2019, S. 289; Goodfellow et al. 2016, S. 105; Bishop 2006, S. 3).

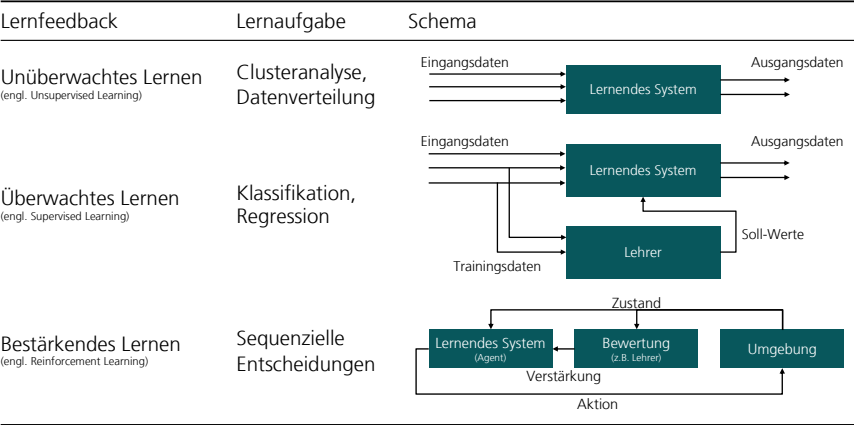
Das *überwachte Lernen* lernt anhand von Beispielpaaren, denen Soll-Werte über Labels zugewiesen sind, mit Eingangs- und Ausgangsdaten¹². Über die Soll-Werte wird hier ein explizites Feedback durch einen Lehrer¹³ vorgegeben. Überwachtes Lernen zielt darauf ab, nach der Trainingsphase auch für unbekannte Eingangsdaten entsprechende Ausgangsdaten zu präzisieren. Angewendet wird das überwachte Lernen zur Regression, der Prognose von kontinuierlichen Zahlenwerten und zur Klassifikation, der Zuordnung diskreter Katego-

¹¹engl. Input Data, häufig auch als Features bezeichnet

¹²engl. Output Data

¹³Lehrer ist an dieser Stelle abstrakt als Vorgabe für Soll-Werte zu verstehen und ist in der Regel keine Person.

Tabelle 2.2: Gliederung verschiedener Ansätze des ML (eigene Darstellung in Anlehnung an Russell & Norvig 2021, S. 653; Alpaydin 2020, S. 11–13; Döbel et al. 2018, S. 10; Bishop 2006, S. 3)



rien zu den Eingangsdaten (Russell & Norvig 2021, S. 653; Alpaydin 2020, S. 11; Richter 2019, S. 2–3; Goodfellow et al. 2016, S. 105–106; Bishop 2006, S. 3). Das *bestärkende Lernen* funktioniert über eine Serie von Verstärkungen, die entweder eine Belohnung oder eine Bestrafung auslösen können. Dabei interagiert das lernende System, der sogenannte Agent, mit seiner Umwelt und beeinflusst mit seinen Aktionen meist nicht nur die direkte Belohnung oder Bestrafung, sondern auch die Belohnungen und Bestrafungen der nachfolgenden Zeitschritte. Ziel des bestärkenden Lernens ist es, passende Aktionen zur Maximierung der Belohnung zu finden, ohne dass die Aktionen durch einen Lehrer vorgegeben werden (Russell & Norvig 2021, S. 653; Alpaydin 2020, S. 13; Graesser & Keng 2019, S. 1; Szepesvári 2019, S. 3; Sutton & Barto 2018, S. 1; François-Lavet et al. 2018, S. 224; van Otterlo & Wiering 2012, S. 5; Bishop 2006, S. 3). Charakteristisch für bestärkendes Lernen ist das Lernen durch Versuch und Irrtum sowie die oftmals verzögerte Belohnung für Aktionen des Agenten (Graesser & Keng 2019, S. 3; Richter 2019, S. 256; Sutton & Barto 2018, S. 2). Bestärkendes Lernen eignet sich deshalb für sequenzielle Entscheidungen, für welche keine mit Labels versehenen Trainingsdaten verfügbar sind und damit aus der Erfahrung eines Agenten gelernt werden muss (Sutton & Barto 2018, S. 2; François-Lavet et al. 2018, S. 233).

Beim bestärkenden Lernen, fortan analog zur Literatur mit dem englischen Akronym Reinforcement Learning (RL) abgekürzt, lernt ein Agent also eine Strategie¹⁴, um eine kumulative Belohnung zu maximieren. Diese Strategie wird in Interaktion zwischen Agent und Umgebung aus einer Reihe von Zuständen und durchgeführten Aktionen sowie deren entsprechenden Verstärkungen gelernt, die entweder Belohnungen (positiv) oder Bestrafungen (negativ) sein können. Im Falle von Bestrafungen bedeutet Maximierung, dass der Agent bestrebt ist, die kumulative Bestrafung zu minimieren. Zur Vereinfachung wird im Folgenden nur noch von Belohnungen gesprochen. Dem Agenten wird dabei nicht vorgegeben, welche Aktion bei welchem Zustand ausgeführt werden soll, sondern er muss durch Versuch und Irrtum herausfinden, welche Aktionen die größte kumulative Belohnung ergeben (Graesser & Keng 2019, S. 2; Sewak 2019, S. 2; Sutton & Barto 2018, S. 6). Ein solches Lernproblem lässt sich mit einem Markov-Entscheidungsprozess¹⁵ abbilden (Szepesvári 2019, S. 4; Sutton & Barto 2018, S. 47). Der Markov-Entscheidungsprozess beschreibt Zustände S_t in einem Zustandsraum S , Aktionen A_t in einem Aktionsraum A und Belohnungen R_t für einen Agenten, der zu Zeitpunkten t mit seiner Umgebung interagiert (vgl. Abbildung 2.14). Zustands- und Aktionsraum können diskret oder kontinuierlich angelegt sein. Zu jedem Zeitpunkt t erhält der Agent eine Repräsentation des Zustands der Umgebung $S_t \in S$ und wählt basierend auf seiner Strategie $\pi(S_t)$ eine Aktion $A_t \in A$. Einen Zeitschritt später, zum Zeitpunkt $t + 1$, verursacht durch seine Aktion, erhält der Agent eine Belohnung, $R_{t+1} \in R$ und findet sich in einem neuen Zustand S_{t+1} wieder (Sutton & Barto 2018, S. 48). Mit der Markov-Eigenschaft, die besagt, dass der neue Zustand S_{t+1} nur vom vorherigen Zustand S_t und der Aktion A_t abhängt, ergeben sich die Belohnungsfunktion $R(S_t, A_t, S_{t+1})$ und die Übergangsfunktion $T(S_t, A_t, S_{t+1})$ mit der Übergangswahrscheinlichkeit $P(S_{t+1}|S_t, A_t)$ (Graesser & Keng 2019, S. 5–7; Richter 2019, S. 256–257; François-Lavet et al. 2018, S. 236; Sutton & Barto 2018, S. 49).

Das Ziel des Agenten beim RL ist nicht die Optimierung der unmittelbaren Belohnung R_t , sondern die der langfristigen, kumulierten Belohnung, auch als Rückgabe¹⁶ G_t bezeichnet. Im einfachsten Fall ist dies die Summe der Einzelbelohnungen der Schritte zwischen den Zeitpunkten $t = t_0$ bis $t = T$ mit T als finalem Zeitpunkt. Dieser Zeitraum wird auch als Episode

¹⁴engl. Policy

¹⁵engl. Markov Decision Process

¹⁶engl. Return

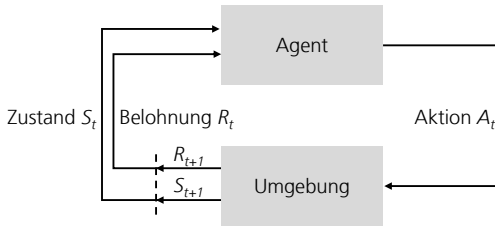


Abbildung 2.14: Interaktion eines Agenten mit seiner Umgebung (eigene Darstellung in Anlehnung an Sutton & Barto 2018, S. 48)

bezeichnet (Sutton & Barto 2018, S. 54). Die Abfolge der Zustände, Aktionen und Belohnungen kann als Zustands-Aktions-Belohnungs-Trajektorie $\tau = (S_0, A_0, R_0), \dots, (S_T, A_T, R_T)$ beschrieben werden (Graesser & Keng 2019, S. 8). In vielen Fällen ist es sinnvoll, einen Diskontierungsfaktor $\gamma \in [0, 1]$ für die Rückgabe G_t einzusetzen, um Belohnungen unterschiedlich zu gewichten (vgl. Gleichung 2.6). Mit $\gamma = 0$ agiert der Agent kurzsichtig, da er sich nur mit der Maximierung der unmittelbaren Belohnung beschäftigt. Wenn sich γ an 1 nähert, berücksichtigt die Rückgabe die zukünftigen Belohnungen stärker, der Agent agiert weitsichtiger (Sutton & Barto 2018, S. 55; Richter 2019, S. 261).

$$G_t = \sum_{k=0}^{T-t-1} \gamma^k R_{t+k+1} \text{ mit } 0 \leq \gamma \leq 1 \quad (2.6)$$

Zur Lösung des Markov-Entscheidungsprozess muss der Agent eine optimale Strategie π^* finden (Sewak 2019, S. 21; Arulkumaran et al. 2017, S. 28). Diese muss er jedoch nicht direkt lernen, vielmehr beinhaltet das Lernen eines Agenten, dass dieser eine oder mehrere der nachfolgenden Möglichkeiten lernt: eine Strategie, eine Wertfunktion oder ein Modell seiner Umgebung. Die ersten beiden Möglichkeiten werden dabei als modellfrei bezeichnet, die letzte als modellbasiert¹⁷ (Graesser & Keng 2019, S. 9–10; François-Lavet et al. 2018, S. 238). Lernt der Agent direkt eine Strategie¹⁸, zielt diese optimale Strategie π^* darauf ab, Aktionen auszuwählen, welche die Rückgabe maximieren. Der Vorteil daran ist, dass der Lösungsansatz sehr generisch ist und auf unterschiedliche Problemstellungen angewendet werden kann. Dabei konvergiert er auch stets. Die Stichprobeneffizienz, also die Effizienz des Lernens auf Basis eines abgegrenzten Datensatzes, ist jedoch vergleichsweise gering

¹⁷engl. model-free bzw. model-based

¹⁸engl. policy-based

(Graesser & Keng 2019, S. 12; Sutton & Barto 2018, S. 321; Arulkumaran et al. 2017, S. 30). Lernt der Agent eine Wertfunktion¹⁹, nutzt er diese, um Zustände oder Zustands-Aktions-Paare zu bewerten und daraus eine Strategie zur Auswahl von Aktionen abzuleiten. Neben der Zustands-Wert-Funktion $V^\pi(S_t)$, meist vereinfacht als Wertfunktion bezeichnet, ist insbesondere die Qualitätsfunktion $Q^\pi(S_t, A_t)$ verbreitet, da diese direkt die Zustands-Aktions-Paare beinhaltet und deshalb leichter in eine Strategie zu überführen ist. Mit der Advantage-Funktion $A^\pi(S_t, A_t) = Q^\pi(S_t, A_t) - V^\pi(S_t)$ existiert auch ein Ansatz zur Kombination beider Funktionen. Analog zur optimalen Strategie existieren auch optimale Wertfunktionen $V^*(S_t)$, $Q^*(S_t, A_t)$ und $A^*(S_t, A_t)$. Das Lernen einer Wertfunktion ist deutlich stichprobeneffizienter als dies beim direkten Lernen einer Strategie der Fall ist. Daten aus der Umgebung werden dabei deutlich besser genutzt und die Varianz ist niedriger, Konvergenz ist jedoch nicht garantiert (Graesser & Keng 2019, S. 13; François-Lavet et al. 2018, S. 238; Sutton & Barto 2018, S. 58; Arulkumaran et al. 2017, S. 29). Modellbasierte Agenten nutzen entweder ein Modell ihrer Umgebung oder lernen dieses in Form der Übergangsfunktion des Markov-Entscheidungsproblems. Mit solch einem Modell ist es möglich, die Zustands-Aktions-Trajektorie für einige Zeitschritte in die Zukunft zu prädictieren. Modellbasierte Agenten benötigen damit einen deutlich geringeren Stichprobenumfang, da das Wissen über die Umgebung im Modell enthalten ist. Die meisten Umgebungen sind jedoch stochastisch und deshalb ist deren Übergangsfunktion nicht bekannt. Das dennoch mögliche Lernen von Modellen ist eine junge Disziplin und noch nicht sehr weit entwickelt (Graesser & Keng 2019, S. 13–14). Im Gegensatz zum modellbasierten Lernen haben Agenten im Fall des Lernens einer Strategie oder einer Wertfunktion keinen Zugriff auf die Übergangsfunktion und die Belohnungsfunktion. Die einzige Möglichkeit für einen Agenten, Informationen über diese Funktionen zu erhalten, sind die Zustände, Aktionen und Belohnungen, die er tatsächlich in seiner Umgebung erfährt, also die Tupel (S_t, A_t, R_t) (Graesser & Keng 2019, S. 7).

Für jede der Lernmöglichkeiten und teilweise auch deren Kombination existieren diverse Algorithmen (vgl. Abbildung 2.15), für deren Detaildarstellung auf die Literatur verwiesen wird (Graesser & Keng 2019; Szepesvári 2019; François-Lavet et al. 2018). In der Literatur häufig verwendete Algorithmen sind Deep Q-Network (DQN) mitsamt seinen Weiterent-

¹⁹engl. value-based

wicklungen, welche in Rainbow zusammengefasst sind, Asynchronous Advanced Actor Critic (A3C) und Proximal Policy Optimization (PPO). Für verteilte Architekturen wurden Algorithmen wie Distributed Prioritized Experience Replay (Ape-X) für DQN, Importance Weighted Actor-Learner Architectures (IMPALA) für A3C und Asynchronous Proximal Policy Optimization (APPO) für PPO entwickelt. Die Algorithmen wurden basierend auf ihren wissenschaftlichen Veröffentlichungen in diverse Open-Source-Frameworks für RL integriert, u.a. SLM Lab (Graesser & Keng 2019), RLlib (Liang et al. 2018) und Coach (Caspi et al. 2017). Da reale Problemstellungen oftmals hochdimensionale Zustandsräume aufweisen, haben sich in der Praxis insbesondere Algorithmen etabliert, die auf tiefen Künstlichen Neuronalen Netzen (KNN) basieren und als Deep Reinforcement Learning bezeichnet werden. KNN sind darauf ausgelegt, hochdimensionale Eingangsdaten zu verarbeiten, ohne bei zusätzlichen Dimensionen einen exponentiell wachsenden Datensatz zu benötigen. Des Weiteren können KNN sehr gut inkrementell trainiert werden, wie dies beim RL der Fall ist (François-Lavet et al. 2018, S. 238–239).

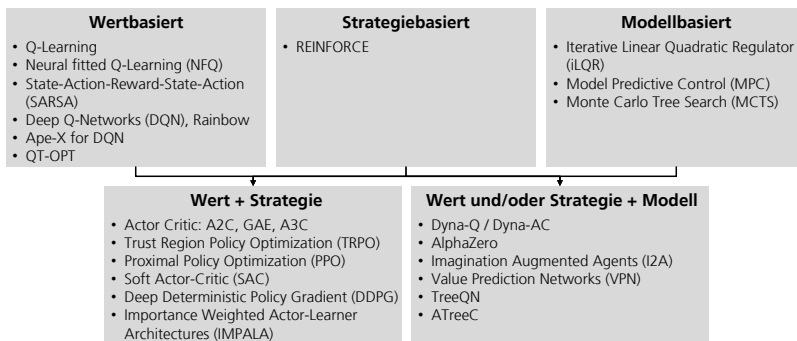


Abbildung 2.15: Algorithmen-Familien für bestärkendes Lernen (eigene Darstellung in Anlehnung an Graesser & Keng 2019, S. 12; François-Lavet et al. 2018, S. 242–271)

Algorithmen können bezüglich ihrer Trainingsphase in On-policy und Off-policy gegliedert werden. Während On-policy-Algorithmen ihre Strategie direkt bewerten und verbessern, nutzen Off-policy-Algorithmen zwei verschiedene Strategien. Eine Strategie für die Generierung von Daten und getrennt davon eine zweite, lernende Strategie. Off-policy-Algorithmen können somit meist schneller lernen, da sie aus jeder Form von Erfahrung aus der Interaktion mit ihrer Umgebung lernen können (François-Lavet et al. 2018, S. 241; Sutton & Barto 2018,

S. 100). Das Training der Algorithmen kann darüber hinaus in offline und online gegliedert werden. Beim Offline-Training werden die Erfahrungen des Agenten a priori gesammelt und dann gebündelt für das Lernen verwendet, wohingegen diese beim Online-Training in einer sequenziellen Reihenfolge verfügbar und verwendet werden. Der Kerngedanke der Algorithmen ist bei beiden Ansätzen der gleiche. Der Hauptunterschied besteht darin, dass nur beim Online-Training der Agent zu jedem Zeitschritt beeinflussen kann, wie er Erfahrung sammelt (François-Lavet et al. 2018, S. 233–234).

Beim Training eines Agenten muss zudem ein Kompromiss zwischen Exploration und Exploitation gefunden werden, was auch als Dilemma des RL bezeichnet wird. Bei der Exploration geht es um die Beschaffung von Informationen über die Umgebung (Übergangsfunktion und Belohnungsfunktion), während es bei der Exploitation um die Maximierung der erwarteten Rückgabe angesichts des aktuellen Wissens des Agenten geht. Der Agent muss einen Kompromiss finden und dabei entscheiden, ob er mehr über seine Umgebung lernen (Exploration) oder basierend auf seiner aktuellen Strategie die vielversprechendsten Aktionen auswählen will (Exploitation) (François-Lavet et al. 2018, S. 284–285; Sutton & Barto 2018, S. 3; Arulkumaran et al. 2017, S. 35). Verfahren zur Kompromissfindung zwischen Exploration und Exploitation sind der Literatur zu entnehmen (François-Lavet et al. 2018, S. 286–288).

2.3.3 Regelkreise als Handlungsmöglichkeit lernender Systeme

Regelung²⁰, dargestellt in Abbildung 2.16, bezeichnet einen „Vorgang, bei dem fortlaufend eine variable Größe, die Regelgröße x , erfasst (gemessen), mit einer anderen variablen Größe, der Führungsgröße w , verglichen und im Sinne einer Angleichung an die Führungsgröße beeinflusst wird“ (DIN IEC 60050-351, S. 136, ergänzt um Formelzeichen). Messen ist hier allgemein gefasst und bedeutet die Abbildung der Regelgröße auf eine andere Größe, die zur weiteren Verarbeitung besser geeignet ist als die Regelgröße selbst (Föllinger et al. 2016, S. 2–3). Ziel der Regelung ist es, innerhalb der Regelstrecke auch bei einem Auftreten von Störungen, abgebildet über die Störgröße z , durch Anpassung der Stellgröße y die Regelgröße schnellstmöglich der Führungsgröße nachzuführen (Föllinger et al. 2016,

²⁰engl. Closed-Loop Control. In der Literatur wird für die Regelung häufig auf DIN 19226 verwiesen, welche jedoch durch DIN IEC 60050-351 ersetzt wurde.

S. 4; DIN IEC 60050-351, S. 206). Die Störgröße wirkt darauf hin, die Regelgröße vom gewünschten Verhalten abzubringen. Darüber hinaus ist ihr zeitliches Verhalten nicht genau vorhersagbar, weil es unerwarteten Änderungen unterliegt (Föllinger et al. 2016, S. 1). Für das Grundprinzip der Regelung spielt es dabei keine Rolle, ob eine oder mehrere Regelgrößen zu beeinflussen sind. Im zweiten Fall wird dies als Mehrgrößenregelung bezeichnet (Föllinger et al. 2016, S. 15). Anzumerken ist, dass die Begriffe Regelglied, Regler und Regeleinrichtung im gängigen Sprachgebrauch oftmals synonym verwendet werden (Föllinger et al. 2016, S. 3). Wird von einem Regler gesprochen, ist damit meist – ebenso wie im weiteren Verlauf dieser Arbeit – die Regeleinrichtung gemeint.

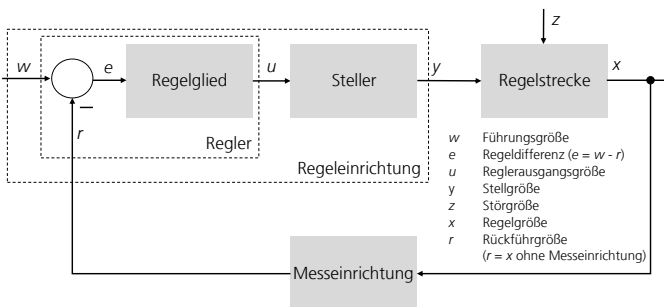


Abbildung 2.16: Regelkreis mit Störgröße (eigene Darstellung in Anlehnung an Föllinger et al. 2016, S. 3; DIN EN 60027-6, S. 15)

Kennzeichnend für die Regelung ist der geschlossene Wirkungsablauf mit fortlaufender Selbstbeeinflussung der Regelgröße im Wirkungsweg des Regelkreises (DIN IEC 60050-351, S. 136). Darin besteht auch der Unterschied zur Steuerung²¹. Der Wirkungsweg kann bei der Steuerung zwar ebenfalls geschlossen sein, der Wirkungsablauf ist jedoch offen, die Ausgangsgrößen wirken also nicht fortlaufend wieder über dieselben Eingangsgrößen auf sich selbst. Auch bei der Steuerung geht es also um eine Verminderung oder Vermeidung von Abweichungen zwischen Soll- und Ist-Werten, die jedoch nicht kontinuierlich nachgeregelt wird. Im Gegensatz zur Steuerung kann die Regelung Störungen erkennen und nachregeln (DIN IEC 60050-351, S. 137–138).

Bei der Regelung werden die zeitkontinuierliche Regelung und die Abtastregelung unterschieden. Für eine zeitkontinuierliche Regelung wird die Regelgröße ununterbrochen erfasst

²¹engl. Open-Loop Control

und mit der Führungsgröße verglichen, wohingegen dies bei der Abtastregelung diskontinuierlich und damit nur zu festgelegten Zeitpunkten geschieht (DIN IEC 60050-351, S. 145). Eine adaptive Regelung, zeitkontinuierlich oder diskontinuierlich realisiert, kann abhängig von Veränderungen in Betriebsbedingungen oder Betriebszuständen die Struktur oder Parameter der Regelung selbsttätig anpassen. Dies ist insbesondere dann notwendig, wenn die Parameter der Regelstrecke bedingt durch die Betriebsbedingungen oder Betriebszustände signifikant schwanken (Föllinger et al. 2016, S. 15–16; DIN IEC 60050-351, S. 161). Um Modelle an veränderte Gegebenheiten anzupassen, ist eine adaptive Regelung notwendig. Der Regler muss also lernfähig sein und damit den eigenen Handlungsraum kontinuierlich anpassen. Gleichzeitig sind Regelkreise eine reale Handlungsmöglichkeit für technische Systeme und damit ein wichtiger Bestandteil autonomer Systeme (vgl. Abschnitt 2.3.1). Wird eine Regelungsaufgabe verschachtelt aus mehreren Regelkreisen aufgebaut, wird dies als Kaskadenregelung bezeichnet (Föllinger et al. 2016, S. 218).

2.3.4 Regelkreise in der Produktionsplanung und -steuerung

Allgemein auf ein Produktionssystem bezogen formuliert Dyckhoff (2006, S. 7) einen Regelkreis bestehend aus Managementsystem und Leistungssystem. Die Planung und Steuerung stellt dabei das Managementsystem dar, die Wertschöpfung das Leistungssystem. Die Übertragung regelungstechnischer Prinzipien auf die Interaktion zwischen PPS und Fertigung erfährt bereits seit längerer Zeit eine hohe Aufmerksamkeit. Der Grundgedanke dabei ist, mittels automatischer und kontinuierlicher Informationsrückkopplung aus der Fertigung einen geschlossenen Wirkungsweg aufzubauen, um der PPS, trotz Störungen, durch veränderte Stellgrößen eine Nachführung der Regelgröße an deren Plan-Werte in Form der Führungsgröße zu ermöglichen (Schuh et al. 2011a, S. 67–68; Zäpfel 2001, S. 224–225). Zwar werden damit Analogien zwischen Regelkreisen und der Produktion abgeleitet, dies muss jedoch nicht zwingend eine regelungstechnische Auslegung beinhalten – die Betrachtung der Produktion als Regelkreis folgt dem kybernetischen Managementansatz (Warnecke 1989, S. 111).

Erste Ideen zur Nutzung von regelungstechnischen Ansätzen für Produktionssysteme gehen auf Simon (1952) zurück. Darauf aufbauend lässt sich, wie in Abbildung 2.17 dargestellt,

die Interaktion zwischen PPS und Fertigung als Kaskadenregelung (vgl. Abschnitt 2.3.3) beschreiben, welche eine innere Schleife mit der Produktionssteuerung als Regler und eine äußere Schleife mit der Produktionsplanung als Regler beinhaltet (Arping 1977, S. 13–14; Wiendahl et al. 1993a, S. 256; Pritschow & Wiendahl 1995, S. 421). Für die Produktionsplanung als Regler stehen dabei Durchschnittswerte der Zielgrößen (vgl. Abschnitt 2.2.2) im Fokus, für die Produktionssteuerung einzelne Aufträge und Fertigungsprozesse (Pritschow & Wiendahl 1995, S. 421–422). Nach Wiendahl (1997, S. 258–259) sind aus Sicht der Regelungstechnik die nachfolgenden Kriterien zu erfüllen, um von einer Fertigungsregelung zu sprechen. Zuerst müssen eindeutige, messbare Regelgrößen definiert werden. Des Weiteren ist der Regler so auszulegen, dass er das Systemverhalten nachvollziehbar beeinflusst, beispielsweise hinsichtlich Stabilität, Führungsverhalten und Störverhalten. Darüber hinaus muss die Regelstrecke eine Rückführung enthalten, was meist über BDE und MDE gelöst wird. An Bedeutung verloren haben die Kriterien des Vorhandenseins eines regelungstechnischen Modells der Regelstrecke und der Wirkzusammenhänge zwischen Zielgrößen und Stellgrößen. Durch den Einsatz von ML können diese Zusammenhänge zunehmend aus Daten gelernt und approximiert werden (vgl. Abschnitt 2.3.2).

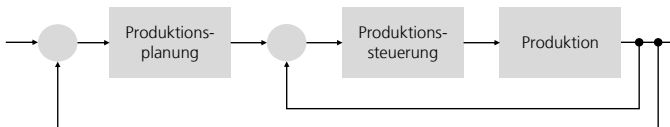


Abbildung 2.17: Grundstruktur der PPS als Regelkreis (eigene Darstellung in Anlehnung an Pritschow & Wiendahl 1995, S. 421)

Einen guten Überblick und eine Klassifikation existierender Ansätze der Fertigungsregelung gibt Niehues (2017). Sie lassen sich anhand ihres Reglertyps in Bestandsregelung, Kapazitätsregelung, Belastungsregelung, wissensbasierte Regelung und Regelung durch Neu-/Umplanung einteilen (vgl. Abbildung 2.18). Die Reglertypen unterscheiden sich dabei in ihren Führungsgrößen, Regelgrößen und Stellgrößen (vgl. Abschnitt A.4).

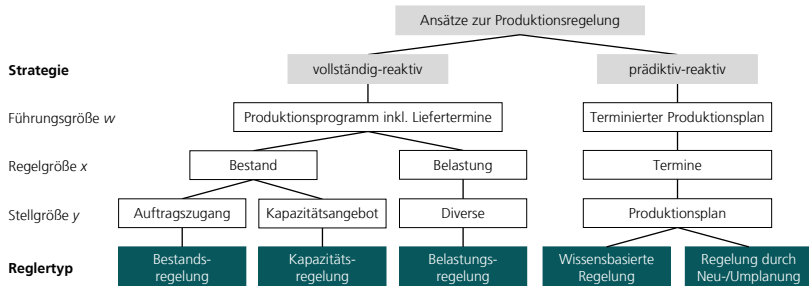


Abbildung 2.18: Klassifikation von Ansätzen der Fertigungsregelung (eigene Darstellung in Anlehnung an Niehues 2017, S. 21)

2.4 Zwischenfazit und Handlungsbedarf

Die Kombination aus global verteiltem Wertschöpfungsnetzwerk mit darin eingebetteten Produktionsstandorten, organisiert als komplexe Werkstattfertigung, stellt ein komplexes System dar. Dieses System ist charakterisiert durch die Vielzahl und Vielfalt seiner Elemente und ihrer Beziehungen sowie die Dynamik und Unsicherheit des Systemverhaltens, also eine statische und dynamische Komplexität (vgl. Abschnitt 2.1; Abschnitt 2.1.2). Hieraus folgen vielfältige Ereignisse, die das Management des Wertschöpfungsnetzwerks vor große Herausforderungen stellt. Logistische Zielgrößen, welche durch diese Ereignisse beeinträchtigt werden, stehen dabei zunehmend im Fokus unternehmerischer Exzellenz aus Kundenperspektive. Des Weiteren können negative Ereignisse auch zu Produktivitätsverlust und zum Aufbau von Sicherheitsbeständen als Puffer führen, weshalb sie signifikante finanzielle Auswirkungen auf die Unternehmen des Wertschöpfungsnetzwerks haben (vgl. Abschnitt 2.1.3). Zwar wird die Reaktion auf Ereignisse aus dem Wertschöpfungsnetzwerk bereits durch IT-Systeme unterstützt, diese sind bislang jedoch nur wenig mit der PPS verknüpft (vgl. Abschnitt 2.1.4).

Dem gegenüber steht die größtenteils statisch aufgebaute PPS, welche die Schnittstelle zwischen Wertschöpfungsnetzwerk und Produktionssystem eines Produktionsstandorts darstellt. Es ist die Aufgabe der PPS, das Produktionsprogramm zu planen und auch bei unvermeidbaren Ereignissen zu realisieren. Ereignisse aus dem Wertschöpfungsnetzwerk werden dabei derzeit jedoch nur zeitversetzt über Planungsobjekte berücksichtigt (vgl. Abschnitt 2.2).

Innerhalb der PPS kann zwischen der Produktionsplanung und der Produktionssteuerung unterschieden werden, die häufig am Punkt der Auftragsfreigabe ineinander übergehen. Die Produktionssteuerung ist dabei für die Realisierung des Produktionsprogramms verantwortlich, hat jedoch keine direkte Verbindung zum Wertschöpfungsnetzwerk. Sofern im Wertschöpfungsnetzwerk Ereignisse auftreten, werden diese erst nach sequentiell ablaufenden Planungsschritten an die Produktionssteuerung propagiert (vgl. Abschnitt 2.2.1; Abschnitt 2.2.3). Je nach Ausgestaltung der IT-Systeme in der PPS sind diese Schritte nicht zwangsläufig alle vernetzt und automatisiert, sondern können auch Medienbrüche oder manuelle Tätigkeiten beinhalten (vgl. Abschnitt 2.2.4).

Auch die der PPS bzw. deren IT-Systemen zugrunde liegenden Modelle sind oft statisch aufgebaut und werden nach initialem Aufbau nicht mehr aktualisiert. Vielmehr findet nur eine Versorgung mit Bewegungsdaten statt, die in unterschiedlicher Aktualität und Akkuratess vorliegen. Das Modell passt sich nicht an veränderte Situationen in der Realität an, der Entscheidungsspielraum bleibt unverändert. Dadurch entsteht eine zunehmende Lücke zwischen realem System und dessen digitalem Modell (vgl. Abschnitt 2.2.4). Bedingt durch diese Lücke werden deshalb gegebenenfalls Entscheidungen in der PPS getroffen, die für die Situation des realen Systems nicht ideal sind. Auch steigt durch die Einschränkungen des Modells die Anzahl manueller Eingriffe, die jedoch ein seltsames oder unerwartetes Verhalten des Produktionssystems auslösen können. Bedingt ist dies durch die Systemkomplexität der PPS, welche es dem Menschen nicht erlaubt, alle Einflussfaktoren und Auswirkungen von Entscheidungen in der Produktionssteuerung zu überblicken und zu bewerten (vgl. Abschnitt 2.2.1; Abschnitt 2.2.4).

Lernende Systeme bieten interessante Ansatzpunkte für die Gestaltung adaptiver Modelle (vgl. Abschnitt 2.3). Die Grundlage für eine Lernfähigkeit technischer Systeme stellt das ML dar, für sequenzielle Entscheidungsprobleme, wie sie insbesondere in der Produktionssteuerung vorliegen, konkret das RL (vgl. Abschnitt 2.3.2). Regelkreise nehmen dabei eine Doppelrolle ein. Einerseits ermöglichen Regelkreise durch die Informationsrückkopplung das Lernen technischer Systeme, andererseits stellen sie auch die Handlungsmöglichkeit lernender Systeme im realen System dar (vgl. Abschnitt 2.3.1; Abschnitt 2.3.3). Das Potenzial von Regelkreisen in der PPS wurde bereits erkannt und Aspekte davon in diversen Ansätzen realisiert. Diese Ansätze fokussieren sich jedoch auf die Interaktion zwischen PPS,

konkreter meist der Produktionssteuerung, und dem Produktionssystem. Eine Schnittstelle zum Wertschöpfungsnetzwerk ist weiterhin nur über die Produktionsplanung vorgesehen. Des Weiteren sind die Ansätze meist nur für bekannte Parameter ausgelegt und nicht oder nur eingeschränkt als lernende Systeme. Auch der Automatisierungsgrad ist meist gering. Nur wenige Konzepte berücksichtigen mehrdimensionale Parameter als Stellgrößen (vgl. Abschnitt 2.3.4).

Basierend auf den erarbeiteten Grundlagen kann die erste Teilforschungsfrage nach den relevanten Bestandteilen der komplexen Werkstattfertigung und ihres Wertschöpfungsnetzwerks, um die Produktionssteuerung basierend auf Ereignissen aus dem Wertschöpfungsnetzwerk zu optimieren, beantwortet werden. Aus diesen relevanten Bestandteilen und ihren Herausforderungen leitet sich auch der folgende Handlungsbedarf ab:

- Es muss eine direkte, ereignisbasierte Verknüpfung zwischen dem Wertschöpfungsnetzwerk und der Produktionssteuerung vorgesehen werden, um die Ermittlung und Ausführung von Korrekturmaßnahmen in der Produktionssteuerung zu ermöglichen.
- Beim Konzeptdesign sind manuelle Eingriffe zur Reaktion auf Ereignisse zu vermeiden, da diese zu unerwartetem Verhalten des Produktionssystems führen können.
- Modelle in der Produktionssteuerung müssen lernfähig gestaltet sein, um den Entscheidungsspielraum an sich stetig verändernde Situationen anzupassen.
- Eine hohe Datenqualität und ein integriertes Datenmanagement über alle Systemkomponenten hinweg sind sicherzustellen.

Im Rahmen dieser Arbeit stehen die ereignisbasierte Verknüpfung zwischen Wertschöpfungsnetzwerk und Produktionssteuerung, die Reduktion manueller Eingriffe und die Lernfähigkeit der Modelle in der Produktionssteuerung im Fokus. Für das Handlungsfeld der Datenqualität und des Datenmanagements einschließlich der Vernetzung in der Fabrik sei auf die Vielzahl von Forschungsarbeiten unter den Stichworten Industrie 4.0, Digitale Transformation und Smart Factory verwiesen (Toro et al. 2021; Rieglmayer 2020; Neugebauer 2018; Brambring 2017; Vogel-Heuser et al. 2017a; Vogel-Heuser et al. 2017b; Reinhart 2017).

3 Stand der Wissenschaft und Technik

Vor dem Hintergrund des sich wandelnden, dynamischen Unternehmensumfelds kommt der Erfüllung von Kundenwünschen und damit der Liefertermineinhaltung eines produzierenden Unternehmens eine steigende Bedeutung zu (vgl. Abschnitt 1.1). Dies stellt die Verknüpfung zwischen Wertschöpfungsnetzwerk und PPS sowie die Produktionssteuerung vor Herausforderungen (vgl. Abschnitt 2.4). Das zeigt sich auch bei der Analyse des nationalen und internationalen Stands der Wissenschaft und Technik.

Für die Analyse des Stands der Wissenschaft und Technik wird zuerst das methodische Vorgehen dargestellt (vgl. Abschnitt 3.1), anschließend werden existierende Ansätze kurz erläutert (vgl. Abschnitt 3.2; Abschnitt 3.3; Abschnitt 3.4). Dies stellt die Grundlage für die Bewertung der Ansätze in Abschnitt 4 dar.

3.1 Methodisches Vorgehen

Ausgehend von der Forschungsfrage der Gestaltung einer lernenden, auf Ereignissen aus dem Wertschöpfungsnetzwerk basierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung (vgl. Abschnitt 1.3) und den themenbezogenen Grundlagen (vgl. Abschnitt 2) ergeben sich drei Themengebiete, die hinsichtlich des aktuellen Stands der Wissenschaft und Technik zu untersuchen sind: die Reaktion auf Ereignisse aus dem Wertschöpfungsnetzwerk (vgl. Abschnitt 3.2), die Ereignisverarbeitung in der Produktionssteuerung (vgl. Abschnitt 3.3) und die Nutzung von bestärkendem Lernen in der Produktionssteuerung der Werkstattfertigung (vgl. Abschnitt 3.4).

Für jedes dieser Themenfelder wurden strukturierte Suchstrings erarbeitet und damit in den Datenbanken arXiv, Scopus und TEMA sowie den Verzeichnissen der Deutschen Nationalbibliothek und der US-amerikanischen Library of Congress recherchiert. Für die Filterung der identifizierten Ansätze wurden in einem ersten Schritt Duplikate eliminiert, anschließend anhand der thematischen Relevanz selektiert. Dargestellt in dieser Arbeit sind nur jene Ansätze, welche die höchste thematische Relevanz aufweisen. Ansätze werden

chronologisch absteigend, bei mehreren Ansätzen innerhalb eines Jahres alphabetisch anhand des Erstautors bzw. der Erstautorin sortiert.

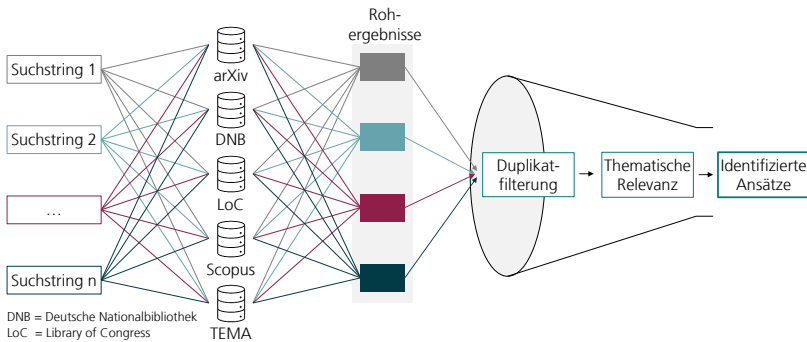


Abbildung 3.1: Methodisches Vorgehen zur Analyse des Stands der Wissenschaft und Technik

3.2 Reaktion auf Ereignisse aus Wertschöpfungsnetzwerken

Das SCM hat die Aufgabe, alle Einheiten und Flüsse in einem Wertschöpfungsnetzwerk zu koordinieren und umfasst damit die Gestaltung, Planung, Ausführung, Steuerung und Überwachung aller Wertschöpfungsnetzwerkbezogenen Aktivitäten. Mit dem SCEM kann SCM auch eine Komponente zur Erfassung, Überwachung, Bewertung und Reaktion auf Ereignisse aus dem Wertschöpfungsnetzwerk beinhalten. Das Interne Supply Chain Management stellt die Schnittstelle zur PPS eines produzierenden Unternehmens dar (vgl. Abschnitt 2.1.4). Nachfolgend werden deshalb zuerst allgemeine Ansätze zur Reaktion auf Ereignisse aus dem Wertschöpfungsnetzwerk dargestellt (vgl. Abschnitt 3.2.1) und im zweiten Schritt Ansätze, welche bereits eine explizite Integration mit der PPS vorsehen (vgl. Abschnitt 3.2.2).

Wird der Begriff Algorithmus breiter gefasst und schließt auch regelbasierte Systeme mit ein, folgen die Ansätze der Annahme, dass nach Eintreten eines Ereignisses Algorithmen notwendig sind, um Maßnahmen abzuleiten und damit den weiteren Verlauf zu beeinflussen (Schiffer et al. 2020, S. 36–37). Die Funktionalität und Automatisierung unterscheidet sich dabei, wie in Abbildung 3.2 dargestellt, jedoch von Ansatz zu Ansatz.

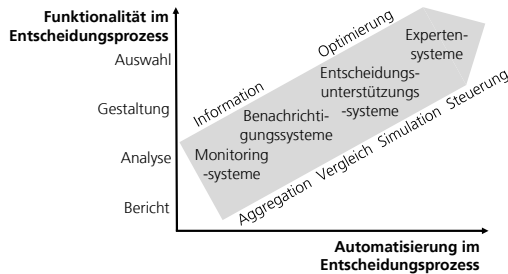


Abbildung 3.2: Charakterisierung des Entscheidungsprozesses (eigene Darstellung in Anlehnung an Heinecke 2013, S. 22)

3.2.1 Fokus auf die Ereignisverarbeitung

Ansätze mit Fokus auf die Ereignisverarbeitung umfassen die Erkennung, Analyse und Bewertung von Ereignissen sowie darauf basierend die Ableitung von Maßnahmen (mit Ausnahme des Ansatzes von Zimmermann et al. (2006)), ohne jedoch eine explizite Verknüpfung mit der Produktion vorzusehen.

Fertier et al. (2021) nutzen Ereignisse, um damit das Modell eines Wertschöpfungsnetzwerks in der Pharmaindustrie zu aktualisieren. Die regelbasierte Maßnahmenauswahl kann dann ebenfalls auf dieses Modell zugreifen, um den aktuellen Zustand des Wertschöpfungsnetzwerks mit einzubeziehen. Ausgewählte Maßnahmen werden in Form eines Entscheidungsunterstützungssystems zur Verfügung gestellt (Fertier et al. 2021, S. 3–5).

GHX LLC (2020) hat einen Ansatz entwickelt, welcher Ereignisketten verarbeitet, relevante Ereignisse identifiziert und regelbasiert Maßnahmen zur Reaktion auf diese Ereignisse ableitet. Damit wird darauf abgezielt, die Stabilität von Wertschöpfungsnetzwerken in der Pharma- und Medizinindustrie zu steigern (GHX LLC 2020).

Palma et al. (2019a) nutzen Fuzzy-Logik zur Bewertung der Kritikalität von Ereignissen in einem Wertschöpfungsnetzwerk der Automobilindustrie und eine daran anschließende systemdynamikbasierte Simulation zur Evaluation der Auswirkungen der Ereignisse. Die Parameter der Simulation müssen je Ereignis manuell angepasst werden. Die Reaktion auf

Ereignisse ist in Form von Aktionsplänen für involvierte Parteien vorgesehen (Palma et al. 2019a, S. 460–464; Palma et al. 2019b, S. 2700–2702).

Vlahakis et al. (2018) zeigen eine IT-Architektur zur Ereignisverarbeitung, in welcher in einem ersten Schritt ein sogenannter Event Listener relevante Ereignisse aus IT-Systemen extrahiert. Die nachfolgende CEP Engine filtert, erkennt und klassifiziert die Ereignisse als Eingangsgröße für den Correlation Agent, welcher Korrelationen zwischen Ereignissen und bekannten Situationen ableitet. Entscheidungen werden dann regelbasiert getroffen und können strategisch (z.B. die Auswahl eines neuen Lieferanten) oder operativ (z.B. die Routenanpassung eines Auftrags) sein. Der Erfolg der Maßnahmen wird über deren Auswirkungen auf Kennzahlen analysiert und für zukünftige Entscheidungen gespeichert (Vlahakis et al. 2018, S. 93–95; Vlachakis & Apostolou 2014, S. 442)

Giannakis & Louis (2016) beschreiben ein Framework, bestehend aus verschiedenen Agenten u.a. für die Produktionsplanung, das Risikomanagement, etc. Der sogenannte Big Data Agent analysiert Datenströme im Wertschöpfungsnetzwerk und entscheidet anhand von fallbasiertem Schließen, abhängig von der Situation, welche über ein Ereignis und die Rahmenbedingungen beschrieben wird, über Maßnahmen. Maßnahmen können dabei sowohl Benachrichtigungen menschlicher Entscheider als auch automatisierte Umsetzungen umfassen (Giannakis & Louis 2016, S. 713–715).

Schuh et al. (2015) fokussieren ihren Ansatz auf Transportvorgänge in einem Wertschöpfungsnetzwerk. Störungen für Transportvorgänge werden klassifiziert und darauf aufbauend Muster mit Aktionen zur Reaktion auf die jeweilige Störung definiert. Ausgewählt werden passende Aktionsmuster dann anhand ihres Einflusses auf die Kostenstruktur des Wertschöpfungsnetzwerks. Schließlich werden sie in Form eines Entscheidungsunterstützungssystems vorgeschlagen (Schuh et al. 2015, S. 147–150).

Behdani (2013) verbindet in seinem Framework das Management von Risiken (proaktiv) und Ereignissen (reaktiv). Der Ereignismanagementprozess besteht aus der Erkennung von Ereignissen, welche auch eine Charakterisierung und simulationsbasierte Auswirkungsanalyse enthält, und der darauf aufbauenden Einleitung von Maßnahmen. Diese bestehen aus

vordefinierten Reaktionsplänen. Die tatsächlichen Auswirkungen der Maßnahmen werden überwacht und in zukünftige Entscheidungen einbezogen (Behdani 2013, S. 34–40).

Bearzotti et al. (2012) beschreiben einen Multiagentenansatz zur Koordination von Aufgaben innerhalb eines Unternehmens und mit Partnern im Wertschöpfungsnetzwerk. Jede Aufgabe wird dabei von einem Agenten repräsentiert. Dazu gehört auch die Erkennung von Ereignissen und die Ableitung von Maßnahmen. Können keine Maßnahmen für das Ereignis bestimmt werden, erfolgt eine Fehlermeldung und Benachrichtigung eines menschlichen Entscheiders (Bearzotti et al. 2012, S. 471–474; Bearzotti et al. 2008, S. 526–527).

Kurbel et al. (2009) zeigen einen Multiagentenansatz mit diversen Agenten innerhalb eines Unternehmens. Ereignisse werden mittels Soll-Ist-Abgleich durch Monitoring-Agenten erkannt, anschließend werden regelbasiert Maßnahmen abgeleitet. Diese Maßnahmen können entweder automatisiert umgesetzt oder in Form eines Entscheidungsunterstützungssystems zur Verfügung gestellt werden (Kurbel et al. 2009, S. 4–5; Jankowska et al. 2007, S. 248–254).

Adhitya et al. (2007a) überwachen in ihrem Ansatz Kennzahlen und leiten bei Soll-Ist-Abweichungen modellbasiert mögliche Maßnahmen ab. Die Auswirkungen der Maßnahmen auf die Kennzahlen werden daraufhin simuliert und mit einem Optimierer die umzusetzende Maßnahme ausgewählt. Umgesetzt werden die Maßnahmen am Beispiel der Ölindustrie mittels einer veränderten Ablaufplanung (Adhitya et al. 2007a, S. 497–508). Einen sehr ähnlichen Ansatz, der heuristikbasiert anstatt modellbasiert arbeitet, zeigen Adhitya et al. (2007b).

Zimmermann et al. (2006) nutzen einen agentenbasierten Ansatz mit Fokus auf die Erkennung und Analyse von Ereignissen. Mittels Fuzzy-Logik werden anschließend relevante Akteure ausgewählt und benachrichtigt, eine Maßnahmenableitung findet nicht statt (Zimmermann et al. 2006, S. 2–4; Bodendorf 2006, S. 680–681).

3.2.2 Fokus auf die Integration der Reaktion in die Produktionsplanung und -steuerung

Ansätze mit Fokus auf die Integration der Reaktion in die PPS umfassen ebenfalls die Ereignisverarbeitung. Sie gehen jedoch weiter und integrieren diese explizit mit der PPS. Maßnahmen beziehen sich deshalb meist auf Planungs- und Steuerungsentscheidungen innerhalb der PPS.

Eirinakis et al. (2021) bauen auf die Arbeit von Vlahakis et al. (2018) auf und zeigen ein System zur Identifikation aktueller und Prädiktion zukünftiger Ereignisse, der Bewertung von Ereignissen und der situationsbezogenen Umsetzung von Maßnahmen. Identifiziert werden Ereignisse über eine regelbasierte CEP Engine, zur Vorhersage von Ereignissen werden ML-Methoden genutzt. Ein modellbasierter Optimierer wählt anschließend Maßnahmen aus, bewertet diese simulativ und setzt sie anschließend in einem neuen Ablaufplan für die Produktion um (Eirinakis et al. 2021, S. 366-373).

Genc et al. (2014) zeigen einen Ansatz zur frühzeitigen Erkennung und Benachrichtigung bei kritischen Ereignissen. Hierfür werden Ereignisse aus dem Wertschöpfungsnetzwerk überwacht und ein Soll-Ist-Abgleich durchgeführt, um kritische Ereignisse zu erkennen. Basierend darauf werden nicht näher spezifizierte Maßnahmen in der PPS ausgewählt und umgesetzt (Genc et al. 2014, S. 41–42).

Guarnaschelli et al. (2013) bauen auf Bearzotti et al. (2012) auf und erweitern den Multiagentenansatz hinsichtlich der Umsetzung von Maßnahmen in einer angepassten Ablaufplanung für die Produktion. Wird ein Ereignis – aus dem Wertschöpfungsnetzwerk oder innerhalb des eigenen Unternehmens – erkannt, wird geprüft, ob dieses eine Planabweichung notwendig macht. Ist dies der Fall, wird der Ablaufplan der Produktion optimiert und gleichzeitig die Partner des Wertschöpfungsnetzwerks über die Auswirkungen in Form eines Ereignisses benachrichtigt (Guarnaschelli et al. 2013, S. 224–226).

Heinecke (2013) beschreibt ein System für die Automobilindustrie zur Verknüpfung von Ereignissen aus dem Wertschöpfungsnetzwerk mit dem MES. Ereignisse werden regel-

basiert über eine CEP Engine erkannt und deren Auswirkungen anhand von Kennzahlen mittels Systemdynamik analysiert. Es werden Maßnahmen zur Anpassung des Ablaufplans der Produktion daraus abgeleitet und anhand von Kennzahlen analysiert (Heinecke 2013, S. 72–98, S. 115–117). Der Ansatz ist eng mit den nachfolgend beschriebenen Arbeiten von Lepratti et al. (2012) und Lamparter et al. (2011) verknüpft.

Lepratti et al. (2012) zeigen ein Assistenzsystem, welches Ereignisse aus dem Wertschöpfungsnetzwerk und der Produktion identifiziert und nach ihrer Kritikalität bewertet. Wird ein Ereignis als kritisch bewertet, stößt das Assistenzsystem eine Änderung des Ablaufplans für die Produktion an. Aufträge werden dabei anhand ihres Schlupfs neu sortiert (Lepratti et al. 2012, S. 38–40).

Lamparter et al. (2011) haben eine Architektur entwickelt, um Ereignisse aus dem Wertschöpfungsnetzwerk mit der Ablaufplanung der Produktion zu verknüpfen. Eingehende Ereignisse werden analysiert, um kritische Situationen anhand von Mustern zu identifizieren. Darauf aufbauend wird regelbasiert entschieden, ob menschliche Entscheider benachrichtigt oder Maßnahmen zur Anpassung der Ablaufplanung automatisiert umgesetzt werden (Lamparter et al. 2011, S. 2724).

Álvarez & Díaz (2011) beschreiben Störungen als Ereignisse, welche einen internen oder externen Ursprung haben können. Bei externem Ursprung wird zusätzlich zwischen Kunden- und Lieferantenseite des Wertschöpfungsnetzwerks unterschieden. Wird eine Störung als Ereignis identifiziert, werden regelbasiert betroffene Agenten – innerhalb eines Unternehmens sowie bei Partnern des Wertschöpfungsnetzwerks – benachrichtigt. Im Falle notwendiger Umplanungen, welche aus dem Ereignis resultieren, wird der Ablaufplan der Produktion mittels eines genetischen Algorithmus optimiert (Álvarez & Díaz 2011, S. 682–686).

3.3 Ereignisverarbeitung in der Produktionssteuerung

Während bei den in Abschnitt 3.2 dargestellten Ansätzen die Reaktion auf Ereignisse aus dem Wertschöpfungsnetzwerk im Fokus steht, zielen die nachfolgend dargestellten Ansätze vor allem auf die Verarbeitung von Ereignissen und die Entscheidungsfindung innerhalb der Produktionssteuerung ab. Dabei werden meist interne Störereignisse, wie beispielsweise Maschinenausfälle, berücksichtigt, weshalb eine kurzfristige Reaktion über die Produktionssteuerung erfolgen muss. Im Rahmen des Aachener PPS-Modells ordnet sich diese in die Eigenfertigungsplanung und -steuerung ein (vgl. Abschnitt 2.2.1). In der Werkstattfertigung werden die Maßnahmen zur Reaktion auf Ereignisse im Rahmen des Scheduling und Dispatchings umgesetzt (vgl. Abschnitt 2.2.3). Gegliedert werden können die analysierten Ansätze in eine Architektursicht mit Fokus auf die Ereignisverarbeitung (vgl. Abschnitt 3.3.1) und eine Ablaufsicht mit Fokus auf dem Scheduling und Dispatching, wobei Ereignisse die Auslöser dafür sind (vgl. Abschnitt 3.3.2).

3.3.1 Fokus auf die Architektur zur Ereignisverarbeitung

Ansätze mit Fokus auf die Architektur zur Ereignisverarbeitung zielen insbesondere darauf ab, die Kommunikation von Ereignissen und die Verarbeitungsschritte in der Produktionssteuerung zu beschreiben. Die Anpassung des Ablaufplans als Maßnahme zur Reaktion auf Ereignisse steht dabei weniger im Vordergrund.

Pielmeier (2020) zeigt ein System zur ereignisorientierten PPS bestehend aus Ereignismodellierung, Ereignisverarbeitung und -analyse sowie einer Methode zur ereignisorientierten Produktionssteuerung. Während die Ereignismodellierung als informationstechnische Grundlage dient, liegt der Fokus der Arbeit auf der Data-Mining-basierten Ereignisverarbeitung und -analyse zur Regelableitung aus Ereignissen, um die Produktionssteuerung laufend zu verbessern. Die Methode zur ereignisorientierten Produktionssteuerung klassifiziert Ursachen und Auswirkungen der Ereignisse, definiert und priorisiert Maßnahmen in der Produktionssteuerung und steuert über Kennzahlen und deren Gewichtung (Pielmeier 2020, S. 67–69; Pielmeier et al. 2019, S. 374–376; Pielmeier et al. 2017, S. 273–274).

Yao et al. (2018) zielen mit ihrem Ansatz vor allem darauf ab, mittels Radio-Frequency Identification Ereignisse zeit-, orts- und objektbezogen zu erkennen. Verarbeitet werden diese anschließend über eine regelbasierte CEP Engine, um die Auftragsreihenfolge anzupassen. Dabei steht vor allem die Priorisierung von Eilaufträgen im Fokus (Yao et al. 2018, S. 231–235).

Kasakow et al. (2016) erweitern die Stücklisten von Aufträgen, um den Auftragsdurchlauf durch eine Werkstattfertigung autonom basierend auf Ereignissen zu steuern. Die Stückliste dient dabei als Vorranggraph und wird mittels First In First Out (FIFO) abgearbeitet. Trotz des Ziels einer autonomen Steuerung werden jedoch nur Statusereignisse (z.B. das Eintreffen eines Kundenauftrags) und keine Störereignisse (z.B. der Ausfall einer Maschine) berücksichtigt (Kasakow et al. 2016, S. 435–437).

Kropp (2016) identifiziert relevante Steuergrößen für die Produktionssteuerung und modelliert deren Zusammenhänge mittels Systemdynamik. Die Liefertermineinhaltung steht dabei im Fokus. Davon ausgehend werden Ereignisse abgeleitet, welche die Steuergrößen beeinflussen. Dies umfasst Änderungen der Steuergrößen selbst sowie Ereignisse aus Maschinen- und Sensordaten. Die Reaktion auf Ereignisse wird in Form von Regeln festgelegt, steht jedoch weniger im Fokus des Ansatzes (Kropp 2016, S. 131–141; Kropp & Schuh 2014, S. 3–5).

Metz (2014) beschreiben eine IT-Architektur zur kontinuierlichen Überwachung von Produktionsprozessen und der Verarbeitung identifizierter Ereignisse mit einer regelbasierten CEP Engine. Abhängig vom Ereignis kann steuernd in die Produktion eingegriffen werden, um die Abweichung zu den auf der Unternehmensebene gesetzten Zielen zu minimieren (Metz 2014, S. 89–90).

Zhang et al. (2009) haben eine ereignisbasierte Plattform entwickelt, welche modular erweiterbar ist, um weitere Produktionsressourcen zu integrieren. Den Mittelpunkt der Plattform bildet der CEP-basierte sogenannte Event Analyzer, der anhand definierter Muster Ereignisse erkennt, filtert, aggregiert und Maßnahmen auswählt (Zhang et al. 2009, S. 15–21).

Walzer et al. (2008) erweitern ein MES um eine Ereignisverarbeitung. Diese umfasst Low-Level-Ereignisse (z.B. eine Variablenänderung einer Maschinensteuerung) und High-Level-Ereignisse (z.B. ein Produktionsauftragsstart). CEP dient zur Überwachung und Verarbeitung der Ereignisse. Im Ergebnis werden anhand vordefinierter Regeln Maßnahmen abgeleitet, welche Benachrichtigungen von menschlichen Entscheidern und automatisierte Ablaufplananpassungen umfassen können (Walzer et al. 2008, S. 384–389).

3.3.2 Fokus auf die Anpassung des Ablaufplans

Das regelbasierte Dispatching ist derzeit in der Praxis weit verbreitet (vgl. Abschnitt 2.2.4), während das Scheduling insbesondere in der Wissenschaft ein sehr aktives Feld ist (Lohmer & Lasch 2021; Zhang et al. 2019; Waschneck et al. 2016; Mönch et al. 2011). Ansätze mit Fokus auf die Anpassung des Ablaufplans nutzen Ereignisse als Auslöser, um daraufhin mittels Scheduling oder Dispatching den Ablaufplan der Produktion anzupassen. Da bereits ein Ablaufplan existiert und dieser, ausgelöst durch ein Ereignis, angepasst wird, bezeichnet die Literatur diese Ablaufplananpassung auch als sogenanntes event-driven rescheduling.

Wang et al. (2020) zeigen einen Ansatz zur Kombination menschlicher Entscheidungsintelligenz und eines algorithmischen Reschedulingsystems im Falle von Störungen. Dabei wird ein Verfahren der Clusteranalyse zur Parameterreduzierung und eine Partikelschwarmoptimierung zur Parameteranpassung genutzt. Deren Ergebnisse sind die Eingangsgröße für eine Fuzzy-Logik zur situationsabhängigen Auswahl von Prioritätsregeln. Der menschliche Entscheider gibt dabei vor, welche Produktionsparameter zu optimieren sind (Wang et al. 2020, S. 789–791).

Tian et al. (2019) beschreiben ein System, welches sowohl einen initialen Ablaufplan generiert, als auch dessen Anpassung im Falle von Maschinenausfällen. Zur Erkennung von Ereignissen werden Produktionsdaten kontinuierlich überwacht und mittels Schwellwertüberwachung ausgewertet. Bei der Überschreitung eines Schwellwerts wird dann die Anpassung des Ablaufplans angestoßen, welche auf Petrinetzen und einem Ameisenalgorithmus aufbaut (Tian et al. 2019, S. 3–9).

Pfitzer et al. (2018) zeigen anhand einer Blechfertigung ein System zur ereignisbasierten Anpassung des Ablaufplans. In diesem Fall konkret dadurch, dass Aufträge priorisiert werden, welche eine geringe Bearbeitungszeit an der Engpassmaschine aufweisen. Damit wird sichergestellt, dass der Puffer vor der Engpassmaschine nicht überlastet wird (Pfitzer et al. 2018, S. 940–942).

Zhu et al. (2018) gliedern Ereignisse in Basisereignisse (vorhersagbare Ereignisse, z.B. das Bearbeitungsende eines Auftrags) und Störereignisse (nicht vorhersagbare Ereignisse, z.B. ein Maschinenausfall). Jede Maschine, Transportsystem, etc. wird dann mit einem Agenten versehen, welche in Summe ein Multiagentensystem bilden. Bei Eintreten eines Basisereignisses wird dezentral durch den jeweiligen Agenten die nächste Aktion ausgewählt und umgesetzt. Bei Störereignissen wird an allen Agenten die Anpassung des Ablaufplans angestoßen (Zhu et al. 2018, S. 122).

Niehues (2017) nutzt Abweichungen bei Start- und Endterminen als Ereignisse zum Auslösen einer Anpassung des Ablaufplans. Ereignisse werden dabei definierten Maßnahmenkaskaden zugeordnet, wobei eine komplette Neubildung des Ablaufplans immer die letzte Maßnahme darstellt. Für die Anpassung des Ablaufplans wird eine Kostenfunktion als Optimierungsgröße genutzt (Niehues et al. 2018, S. 67–69; Niehues 2017, S. 70–74).

Pfund & Fowler (2017) vergleichen anhand eines Anwendungsfalls aus der Leiterplattenherstellung eine periodische Anpassung des Ablaufplans mit einer ereignisbasierten Anpassung. Die ereignisbasierte Ausführung erzielte dabei signifikant bessere Ergebnisse. Voraussetzung hierfür ist jedoch, dass der Algorithmus sehr schnell ausgeführt werden kann, um kurzfristig auf Ereignisse zu reagieren (Pfund & Fowler 2017, S. 3303).

Kalinowski et al. (2014) zeigen ein Vorgehen, um anhand von Ereignissen Methoden zur Anpassung des Ablaufplans auszuwählen, Lösungen zu generieren und zu evaluieren, bevor eine Methode ausgewählt und umgesetzt wird. Ereignisse werden dabei vorab analysiert und der weitere Prozess nur dann angestoßen, wenn eine Anpassung des Ablaufplans notwendig ist – das Vorgehen dieser Analyse wird jedoch nicht eingehend beschrieben (Kalinowski et al. 2014, S. 840–843).

3.4 Nutzung von bestärkendem Lernen in der Produktionssteuerung der Werkstattfertigung

RL beschreibt die Interaktion eines Agenten mit seiner Umwelt, um eine Strategie zu erlernen, welche die Maximierung seiner Belohnung verspricht. Ziel ist es, passende Aktionen zur Maximierung der Belohnung zu finden, ohne dass die Aktionen durch einen Lehrer vorgegeben werden. RL eignet sich deshalb für sequenzielle Entscheidungen, für welche keine gelabelten Daten verfügbar sind und damit aus der Erfahrung eines Agenten gelernt werden muss (vgl. Abschnitt 2.3.2). In der jüngeren Vergangenheit konnte RL beachtliche und auch publikumswirksame Erfolge in Brettspielen (z.B. Schach, Shogi und Go), Videospielen (z.B. Atari, StarCraft) und der Genforschung erzielen (Senior et al. 2020; Schrittwieser et al. 2020; Vinyals et al. 2019; Silver et al. 2018). Große Potenziale werden deshalb auch in der Übertragung der Methoden und Technologien auf Problemstellungen aus der Produktion gesehen (Huber et al. 2021, S. 25). Ansätze wurden beispielsweise entwickelt, um über den Einsatz recycelter Komponenten zu entscheiden (Shah et al. 2010), für die Planung von Wartungs- und Transportvorgängen (Lamprecht et al. 2021; Arviv et al. 2016; Wang et al. 2016), zur Steuerung industrieller Energienetze (Panten 2019) sowie zur energieorientierten PPS (Rösch 2021).

Nachfolgend werden Ansätze vorgestellt, welche das Scheduling und Dispatching in der Werkstattfertigung um RL erweitern¹. Gegenüber den in Abschnitt 3.3 erläuterten Ansätzen steht dabei insbesondere die Integration einer Lernfähigkeit im Fokus.

May et al. (2021) beschreiben ein Multiagentensystem, in welchem jeder Auftrag, jede Maschine und jede Transporteinheit durch einen RL-Agenten repräsentiert wird. Entscheidungen werden über ein wirtschaftliches Gebotssystem ermittelt. Auftragsagenten bieten hierbei für jede Aktion einen Preis und haben das Ziel, die Summe zu minimieren. Maschinen- und Transportagenten wählen unter den Geboten den jeweils nächsten Auftrag aus. Die Performanz des Ansatzes wird im Vergleich mit Prioritätsregeln gemessen, auch in der Kombination von RL und Prioritätsregeln (May et al. 2021, S. 5–7).

¹Die Ansätze von May et al. (2021) und Hofmann et al. (2020) beziehen sich auf eine Matrixfertigung. Da diese jedoch Ähnlichkeiten zur Werkstattfertigung aufweist, werden die Ansätze einbezogen.

Oren et al. (2021) zeigen eine Kombination aus Online-Suche und Offline-Training. Beim Offline-Training wird ein DQN-Agent trainiert. Die Online-Suche, welche auf Monte Carlo Tree Search basiert, dient dann zur Auswahl der robustesten Lösung. Verglichen wird die Performanz mit Benchmark-Datensätzen, für welche die mathematisch exakte Lösung bekannt ist (Oren et al. 2021, S. 99–102).

Zhao et al. (2021) nutzen DQN und gestalten basierend darauf einen Agenten zur Auswahl von Prioritätsregeln für die Entscheidung über den nächsten zu fertigenden Auftrag. Abgezielt wird dabei auf eine Verbesserung der Liefertermineinhaltung. Die Performanz wird mit bekannten Prioritätsregeln verglichen (Zhao et al. 2021, S. 123000–123005).

Zhou et al. (2021a) bauen auf vernetzte Produktions-IT-Architekturen auf und integrieren echtzeitnahe Daten in ein Multiagentensystem, in welchem jede Maschine durch einen RL-Agenten repräsentiert wird, der über den nächsten Auftrag entscheidet. Durch die echtzeitnahen Daten soll effizient auf Störungen, wie beispielsweise Maschinenausfälle, reagiert werden. Das Multiagentensystem zielt durch eine zusammengesetzte Belohnungsfunktion auf die gleichzeitige Minimierung von Durchlaufzeit und Ausbalancieren der Auslastung ab. Verglichen wird die Performanz mit bekannten Verfahren wie genetischen Algorithmen (Zhou et al. 2021a, S. 2–9; Zhou et al. 2021b, S. 756–760).

Altenmüller et al. (2020) zeigen einen RL-Agent, welcher auf DQN basiert und der insbesondere darauf ausgelegt ist, Zeitabhängigkeiten zwischen Prozessen in einer Halbleiterfertigung bei Entscheidungen über den nächsten Auftrag zu berücksichtigen. Erreicht wird dies über eine Modellierung in der Belohnungsfunktion des Agenten. Verglichen wird die Performanz mit Prioritätsregeln (Altenmüller et al. 2020, S. 321–324).

Chen et al. (2020) kombinieren in ihrem Ansatz genetische Algorithmen mit RL. Der genetische Algorithmus wird dabei zur Optimierung der Auftragsreihenfolge genutzt, RL zur Parameteranpassung für den genetischen Algorithmus. Mit State-Action-Reward-State-Action (SARSA) und Q-Learning werden dabei je nach Trainingszustand des RL-Agent unterschiedliche Algorithmen genutzt, um zuerst die Lerngeschwindigkeit zu steigern und

anschließend die Präzision der Ergebnisse zu erhöhen. Die Performanz hinsichtlich der Durchlaufzeitoptimierung wird mit genetischen Algorithmen verglichen (Chen et al. 2020, S. 4–7).

Dittrich & Fohlmeister (2020) präsentieren ein System aus einem zentralen DQN-Agent und dezentralen Agenten jeweils für Aufträge und Maschinen. Die Maschinenagenten erheben dabei Produktionsdaten, der zentrale DQN-Agent lernt daraus und die Auftragsagenten treffen basierend auf dem erlernten DQN dann Entscheidungen über die nächste Aktion (Dittrich & Fohlmeister 2020, S. 390–391).

Hofmann et al. (2020) erläutern einen Multiagentenansatz, welcher sich aus je einem RL-Agenten pro Auftrag zusammensetzt. Als Algorithmus kommt für die Agenten DQN zum Einsatz. Ziel des Ansatzes ist die Durchlaufzeitoptimierung trotz unvorhergesehenen Maschinenausfällen (Hofmann et al. 2020, S. 26–28).

Kardos et al. (2020) beschreiben ebenfalls einen Multiagentenansatz mit einem RL-Agenten je Auftrag zur Durchlaufzeitminimierung. Neben der nächsten Maschine kann der Agent jedoch auch situationsabhängig eine Prioritätsregel zur weiteren Entscheidung auswählen. Maschinenausfälle und andere Ereignisse werden dabei nicht berücksichtigt. Die Performanz des Ansatzes wird im Vergleich mit Prioritätsregeln ermittelt (Kardos et al. 2020, S. 105–108).

Kuhnle (2020) zeigt einen Multiagentenansatz für die Halbleiterfertigung mit Agenten für Maschinen und Transporteinheiten, welche auf dem Algorithmus Trust Region Policy Optimization (TRPO) basieren. Ziel des Ansatzes ist die Verbindung von Auftrags- und Ressourcenzielgrößen in Form der Minimierung der Durchlaufzeit und der gleichzeitigen Maximierung der Maschinenauslastung. Grundlage ist eine umfangreiche Datenerfassung und Abbildung im Zustandsraum für eine feingranulare Modellierung der Belohnungsfunktion. Die Performanz des Ansatzes wird mit Prioritätsregeln verglichen (Kuhnle 2020, S. 64–93; Kuhnle et al. 2021, S. 859–864; Kuhnle et al. 2019, S. 393–395; Stricker et al. 2018, S. 512–513).

Liu et al. (2020) kombinieren einen Agenten je Maschine in einem Multiagentenansatz zur Durchlaufzeitminimierung. Aufgrund einfacher und in der Literatur standardisierter Problemstellungen in der Anwendung wird die Performanz des Ansatzes nicht nur mit Prioritätsregeln und einem genetischen Algorithmus verglichen, sondern auch mit der mathematisch exakten Lösung (Liu et al. 2020, S. 71755–71758).

Luo (2020) haben einen DQN-basierten Agenten entwickelt, um situationsabhängig eine passende Prioritätsregel zur Verbesserung der Liefertermineinhaltung auszuwählen. Die Prioritätsregel wählt dann den nächsten Prozessschritt aus und weist ihn einer geeigneten Maschine zu. Aufgerufen wird der Agent immer dann, wenn eine Bearbeitung an einer Maschine abgeschlossen wurde oder ein neuer Auftrag für die Fertigung freigegeben wird (Luo 2020, S. 5–11).

Park et al. (2020) beschreiben einen Multiagentenansatz, in welchem dezentrale Agenten an Maschinen auf die Ergebnisse eines zentralen lernenden DQN-Agenten zurückgreifen. Ziel des Systems ist die Minimierung der Durchlaufzeit. Trainiert wird der DQN-Agent mit Datensätzen nur weniger Maschinen, um die Trainingsdauer zu senken. Anschließend wird der Datensatz bei der Validierung hochskaliert. Die Performanz wird mit genetischen Algorithmen verglichen (Park et al. 2020, S. 1422–1427).

Rinciog et al. (2020) nutzen in ihrem Ansatz den ursprünglich für das Spiel Go entwickelten Aufbau AlphaGo Zero für die Optimierung der Auftragsreihenfolge in einer Blechfertigung. Es wird darauf abgezielt, einerseits die Liefertermineinhaltung zu steigern und andererseits den Materialabfall zu minimieren. Letzteres wird durch eine optimierte Anordnung der Werkstücke auf einem Blech erreicht. Der Agent wird immer dann ausgelöst, wenn sich der Status einer Maschine auf wartend verändert. Die Performanz wird im Vergleich zur Prioritätsregel Earliest Due Date (EDD) und Monte Carlo Tree Search ermittelt (Rinciog et al. 2020, S. 344–348).

Waschneck (2020) kombiniert in einem Multiagentenansatz einen DQN-Agent je Maschinengruppe. Wird eine Maschine frei und benötigt einen neuen Auftrag, wird dem Ereignis der betroffene Agent zugeordnet, welcher dann über den nächsten zu bearbeitenden

Auftrag aus der Warteschlange entscheidet. Die Zielgrößen werden je nach Anwendungsfall definiert, evaluiert wird ein Anwendungsfall zur Optimierung der Maschinenauslastung und ein Anwendungsfall zur Steigerung der Liefertermineinhaltung. Trainiert werden die Agenten mit einer Fabriksimulation und können anschließend in der Produktion angewendet werden. Dort können sie mit einem sehr kleinen Anteil an Exploration weitertrainiert werden, jedoch auch jederzeit durch anhand der Fabriksimulation neu trainierte Agenten ersetzt werden (Waschneck 2020, S. 106; Waschneck et al. 2018a, S. 302–304; Waschneck et al. 2018b, S. 1265–1268).

Lee et al. (2019) zielen mit ihrem RL-Agenten darauf ab, den Durchsatz einer Halbleiterfertigung zu erhöhen sowie gleichzeitig die Liefertermineinhaltung zu steigern und die Rüstanzahl zu minimieren. Hierfür wird SARSA als Algorithmus eingesetzt. Eine Einschränkung des Ansatzes ist es, dass die Nachfrage vollständig bekannt oder ähnlich zur bekannten Vergangenheit sein muss. Gegenüber Prioritätsregeln wird die Performanz des Agenten evaluiert (Lee et al. 2019, S. 2239–2243).

Zhao et al. (2019) nutzen einen genetischen Algorithmus zur Erstellung eines initialen Ablaufplans, welcher im Falle von Störungen einer Maschine durch einen Q-Learning-Agenten angepasst wird. Der Agent passt den Ablaufplan jedoch nicht direkt an, sondern wählt situationsabhängig eine Prioritätsregel aus, um die Liefertermineinhaltung trotz Störung möglichst konstant zu halten (Zhao et al. 2019, S. 333–334).

Thomas et al. (2018) beschreiben einen Ansatz mit einem RL-Agenten, basierend auf Q-Learning, welcher den nächsten zu bearbeitenden Auftrag auswählt, und einem zweiten Agenten, welcher mittels eines Klassifikators Engpassmaschinen identifiziert, um deren Kapazität anzupassen. Die Kapazitätsanpassung funktioniert im rein simulativen Ansatz durch Hinzufügen einer weiteren Maschine vom Typ der Engpassmaschine. Der RL-Agent wird immer dann aufgerufen, wenn eine Maschine in den Zustand wartend wechselt oder ein neuer Auftrag in der Warteschlange der Maschine ankommt. Das System zielt darauf ab, den Durchsatz zu erhöhen und wird in seiner Performanz mit Prioritätsregeln verglichen (Thomas et al. 2018, S. 130–134).

Zhang et al. (2017) zeigen einen RL-Agenten, der mittels Q-Learning aus einer Warteschlange den nächsten zu bearbeitenden Auftrag auswählt. Aufgerufen wird der Agent, wenn ein neuer Auftrag an einer Maschine mit dem Status wartend ankommt oder die Bearbeitung an einer Maschine mit nicht leerer Warteschlange abgeschlossen wird. Mit dem Ansatz sollen einerseits die Durchlaufzeit und andererseits der Bestand der Fertigung minimiert werden. Die Performanz wird im Vergleich mit Prioritätsregeln analysiert (Zhang et al. 2017, S. 3899–3906).

Gabel & Riedmiller (2012) präsentieren ein Multiagentensystem, in welchem jeder Agent eine Maschine repräsentiert und sequenziell entscheidet, welcher Auftrag als nächstes bearbeitet wird. Der Fokus des sehr mathematischen Ansatzes liegt dabei auf einem Vergleich verschiedener RL-Algorithmen zur optimierten Ausgestaltung des Ablaufplans (Gabel & Riedmiller 2012, S. 44–51; Gabel 2009, S. 45–138).

3.5 Zwischenfazit

Die strukturierte Analyse des nationalen und internationalen Stands der Wissenschaft und Technik im zurückliegenden Abschnitt zeigt, dass bereits einige Ansätze mit hoher thematischer Relevanz existieren. Ausgehend von der Forschungsfrage der Gestaltung einer lernenden, auf Ereignissen aus dem Wertschöpfungsnetzwerk basierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung und den themenbezogenen Grundlagen ergeben sich drei Themengebiete.

Ansätze zur Reaktion auf Ereignisse aus dem Wertschöpfungsnetzwerk (vgl. Abschnitt 3.2) fokussieren die Erkennung, Analyse und Bewertung von Ereignissen aus dem Wertschöpfungsnetzwerk sowie darauf basierend die Ableitung von Maßnahmen. Die Integration mit der PPS und damit Maßnahmen in Form von Planungs- und Steuerungsentscheidungen berücksichtigt ein Teil der Ansätze.

Die Ansätze zur Ereignisverarbeitung in der Produktionssteuerung (vgl. Abschnitt 3.3) können weiter untergliedert werden in eine architekturorientierte und eine ablauforientierte Sicht. Architekturorientierte Ansätze zielen insbesondere darauf ab, die Kommunikation von Ereignissen und die zugehörigen Verarbeitungsschritte in der Produktionssteuerung zu

beschreiben, wohingegen ablaforientierte Ansätze den Schwerpunkt auf die Anpassung des Ablaufplans der Produktion durch Scheduling oder Dispatching legen. Dafür stellen Ereignisse einen Auslöser dar.

Ansätze zur Nutzung von bestärkendem Lernen in der Produktionssteuerung der Werkstattfertigung (vgl. Abschnitt 3.4) erweitern das Scheduling und Dispatching um eine Lernfähigkeit, werden jedoch nicht immer durch Ereignisse ausgelöst. Insbesondere in diesem Themenfeld ist eine deutliche Zunahme der Veröffentlichungen in der jüngsten Vergangenheit zu beobachten.

Die Analyse des Stands der Wissenschaft und Technik bildet die Grundlage für die Anforderungsdefinition an eine lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung und die darauf aufbauende Bewertung der im Stand der Wissenschaft und Technik identifizierten Ansätze in Abschnitt 4.

4 Anforderungen

Die Analyse der relevanten Grundlagen (vgl. Abschnitt 2) und des darauf aufbauenden Stands der Wissenschaft und Technik (vgl. Abschnitt 3) zeigt, dass bereits diverse Ansätze existieren, um auf Ereignisse aus Wertschöpfungsnetzwerken zu reagieren, um Ereignisse in der Produktionssteuerung zu verarbeiten und um bestärkendes Lernen in der Produktionssteuerung der Werkstattfertigung zu nutzen. Folglich gilt es zu untersuchen und zu beantworten, inwieweit diese Ansätze die in Abschnitt 1.3 formulierte Forschungsfrage der Gestaltung einer lernenden, auf Ereignissen aus dem Wertschöpfungsnetzwerk basierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung bereits beantworten können.

In einem ersten Schritt werden hierfür Anforderungen an eine lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung definiert (vgl. Abschnitt 4.1). Basierend darauf werden im zweiten Schritt existierende Ansätze aus dem Stand der Wissenschaft und Technik den Anforderungen gegenübergestellt und bewertet (vgl. Abschnitt 4.2).

4.1 Anforderungsdefinition

Anforderungen können in solche aus dem Anwendungszusammenhang, an das Entscheidungsmodell und an die Systemarchitektur unterschieden werden. Die Anforderungen aus dem Anwendungszusammenhang beschreiben, welche Kriterien erfüllt werden müssen, um für die vorgesehene Anwendung an der Schnittstelle zwischen Wertschöpfungsnetzwerk, PPS und komplexer Werkstattfertigung geeignet zu sein. Dies zielt insbesondere auf den Handlungsbedarf zur Verknüpfung zwischen Wertschöpfungsnetzwerk und Produktionssteuerung ab (vgl. Abschnitt 2.4). Mit den Anforderungen an das Entscheidungsmodell wird definiert, wie ein solches ausgestaltet sein muss, um der vorgesehenen Anwendung Genüge zu tun. Damit wird einerseits der Handlungsbedarf lernfähiger und adaptiver Entscheidungsmodelle, andererseits auch jener der Vermeidung manueller Eingriffe in die

Produktionssteuerung adressiert (vgl. Abschnitt 2.4). Die Anforderungen an die Systemarchitektur definieren aus IT-Sicht, wie die Teilkomponenten zusammenwirken und wie die Interaktion mit existierenden angrenzenden IT-Systemen eines produzierenden Unternehmens gestaltet ist. Dies zielt insbesondere auf den Handlungsbedarf hinsichtlich der Vermeidung manueller Eingriffe in die Produktionssteuerung ab, in Teilen auch auf jenen der hohen Datenqualität und des integren Datenmanagements (vgl. Abschnitt 2.4). Dargestellt sind die resultierenden zehn Anforderungen, aufgeteilt auf die drei Anforderungsklassen, in Tabelle 4.1. Nachfolgend werden die Anforderungen aus dem Anwendungszusammenhang (vgl. Abschnitt 4.1.1), an das Entscheidungsmodell (vgl. Abschnitt 4.1.2) und an die Systemarchitektur (vgl. Abschnitt 4.1.3) näher ausgeführt.

Tabelle 4.1: Anforderungen an eine lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung (Erweiterung von Bauer et al. 2021a, S. 7–8; Bauer et al. 2020a, S. 1554; Bauer et al. 2020b, S. 94–95)

Anwendungs- zusammenhang	A1	Integration von Wertschöpfungsnetzwerk und Produktionssteuerung
	A2	Berücksichtigung des Fertigungsprinzips der komplexen Werkstattfertigung
	A3	Granularität des relevanten Betrachtungsraums
Entscheidungs- modell	E1	Berücksichtigung externer Ereignisse in der Produktionssteuerung
	E2	Aktionsraum zur Einwirkung auf diskrete Aufträge
	E3	Adaptivität des Entscheidungsmodells
	E4	Lernfähigkeit des Entscheidungsmodells
	E5	Geschwindigkeit des Entscheidungsmodells
System- architektur	S1	Automatisierungsgrad des Systems
	S2	Integration in bestehende Produktions-IT-Landschaft

4.1.1 Anforderungen aus dem Anwendungszusammenhang

Für den Anwendungszusammenhang ist es von besonderer Relevanz, dass eine *Integration von Wertschöpfungsnetzwerk und Produktionssteuerung (A1)* erreicht wird. Einerseits bedeutet dies, dass Wertschöpfungsnetzwerk und Produktionssteuerung nicht nur über abstrakte Planungsobjekte verknüpft werden, sondern direkt über Ereignisse (vgl. Abschnitt 2.1.3; Abschnitt 2.1.4; Abschnitt 2.2). Andererseits muss die Charakteristik des

Wertschöpfungsnetzwerks dann auch bei der Ableitung von Maßnahmen zur Reaktion auf diese Ereignisse in der Produktionssteuerung berücksichtigt werden (vgl. Abschnitt 2.1.2). Neben klassischen internen Zielgrößen eines Produktionssystems steht damit insbesondere die Liefertermineinhaltung als Zielgröße gegenüber dem Wertschöpfungsnetzwerk im Fokus (vgl. Abschnitt 2.2.2). Umgesetzt werden kann solch eine Integration von Wertschöpfungsnetzwerk und Produktionssteuerung durch die Übertragung regelungstechnischer Prinzipien über die PPS hinaus auf die Schnittstelle zwischen Wertschöpfungsnetzwerk und Produktionssteuerung (vgl. Abschnitt 2.3.3; Abschnitt 2.3.4).

Aufgrund von ihrer besonderen Charakteristik ist darüber hinaus die *Berücksichtigung des Fertigungsprinzips der komplexen Werkstattfertigung (A2)* notwendig (vgl. Abschnitt 2.1.1; Abschnitt 2.2.3). Ereignisse sind zwar Treiber der dynamischen Komplexität, bei deren Mitigation muss jedoch zusätzlich auch die statische Komplexität der komplexen Werkstattfertigung berücksichtigt werden (vgl. Abschnitt 2.1.1; Abschnitt 2.1.3).

Hinsichtlich der Abbildung der *Granularität des relevanten Betrachtungsraums (A3)* ist es notwendig, dass der Betrachtungsraum weit genug abstrahiert wird, damit nicht nur Einzelmaschinen, sondern das Produktionssystem als Ganzes berücksichtigt werden kann. Dies ergibt sich aus der Verknüpfung zwischen Wertschöpfungsnetzwerk und Produktionssteuerung. Damit steht nicht die Optimierung von internen logistischen Zielgrößen von Einzelmaschinen im Fokus, sondern die Optimierung der Liefertermineinhaltung des gesamten Produktionssystems gegenüber dem Wertschöpfungsnetzwerk. Gleichzeitig muss der Betrachtungsraum ausreichend feingranular gestaltet sein, um auf diskrete Aufträge oder Maschinengruppen einwirken zu können (vgl. Abschnitt 2.2.4).

4.1.2 Anforderungen an das Entscheidungsmodell

Resultierend aus dem Anwendungszusammenhang ist für ein Entscheidungsmodell auf Ebene der Produktionssteuerung die *Berücksichtigung externer Ereignisse (E1)* notwendig. Dies umfasst sowohl die Analyse potenzieller Konsequenzen der Ereignisse inklusive möglicher Abhängigkeiten, wie auch die Ableitung entsprechender Korrekturmaßnahmen zur Anpassung des Produktionssystems (vgl. Abschnitt 2.1.3; Abschnitt 2.2.4).

Für die Umsetzung von Korrekturmaßnahmen zur Reaktion auf externe Ereignisse durch das Entscheidungsmodell muss der *Aktionsraum zur Einwirkung auf diskrete Aufträge (E2)* gestaltet sein. Die Arbeit mit Durchschnittswerten logistischer Zielgrößen ist hierfür nicht ausreichend. Es müssen Rückführgrößen einzelner Aufträge vorliegen und gleichzeitig die Einwirkung auf diskrete Aufträge über entsprechende Stellgrößen möglich sein (vgl. Abschnitt 2.3.3; Abschnitt 2.3.4).

Aufgrund der statischen und dynamischen Komplexität in der komplexen Werkstattfertigung wie auch im Wertschöpfungsnetzwerk sind statische Entscheidungsmodelle nicht ausreichend. Es muss deshalb eine *Adaptivität des Entscheidungsmodells (E3)* zur Anpassung von Parametern gegeben sein. Die Anpassung bezieht sich dabei auf bekannte Situationen mit veränderten Daten. Entweder durch eine manuelle Anpassung von Parametern zur Veränderung von Wertebereichen oder idealerweise durch datengetriebene Modelle, welche eine automatische Anpassung anhand vordefinierter Datentypen ermöglichen. Die Anpassungsfähigkeit umfasst damit auch eine Skalierbarkeit des Entscheidungsmodells (vgl. Abschnitt 2.1.1; Abschnitt 2.1.2; Abschnitt 2.3.3).

Daran schließt die *Lernfähigkeit des Entscheidungsmodells (E4)* an, geht jedoch über die Adaptivität hinaus. Lernfähigkeit ermöglicht es, aus Situationen und Daten neue Entscheidungen zur optimierten Nutzung des Aktionsraums zu lernen und damit, im Gegensatz zur mathematischen Optimierung, auf unbekannte Daten zu generalisieren. Gleichzeitig kann durch die Lernfähigkeit die Leistung des Entscheidungsmodells im Laufe der Zeit gesteigert werden, wenn insbesondere Abhängigkeiten zwischen Ereignissen, Konsequenzen und Korrekturmaßnahmen weiter exploriert und in zukünftige Entscheidungen integriert werden (vgl. Abschnitt 2.3.1; Abschnitt 2.3.2).

Zur Reaktion auf Ereignisse aus dem Wertschöpfungsnetzwerk auf der Ebene der Produktionssteuerung muss eine hohe *Geschwindigkeit des Entscheidungsmodells (E5)* erreicht werden. Die erforderliche Geschwindigkeit bezieht sich dabei nicht auf eine eventuell vor dem Einsatz des Entscheidungsmodells notwendige Trainingsdauer, sondern auf die Geschwindigkeit der Entscheidungsfindung nach Auftreten eines Ereignisses. Die hohe Entscheidungsgeschwindigkeit ist insbesondere aufgrund des dynamischen Systems der komplexen Werkstattfertigung notwendig (vgl. Abschnitt 2.1.3; Abschnitt 2.1.4; Abschnitt 2.2.4).

4.1.3 Anforderungen an die Systemarchitektur

Ein hoher *Automatisierungsgrad des Systems (S1)* trägt einerseits zur Geschwindigkeit des Systems zur lernenden ereignisbasierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung bei, andererseits auch zur Reduktion von Fehlern bei den Entscheidungen des Entscheidungsmodells. Durch eine konsequente Datenrückführung über maschinenlesbare Schnittstellen zwischen den relevanten Komponenten werden Medienbrüche und damit Übertragungsfehler vermieden sowie Daten unmittelbar in einem maschinenlesbaren Format zur Verfügung gestellt. Darüber hinaus erlauben datenbasierte Entscheidungen rationale Entscheidungen. Im Kontext der Produktion ermöglichen sie die Vermeidung von seltsamem oder unerwartetem Verhalten sowie Unter- oder Übersteuern, wie es bei manuellen Entscheidungen in der Produktionssteuerung oftmals auftreten kann (vgl. Abschnitt 2.1.4; Abschnitt 2.2.4; Abschnitt 2.3.4).

Durch das Agieren an der Schnittstelle zwischen Wertschöpfungsnetzwerk und Produktionssteuerung ist es notwendig, eine *Integration in die bestehende Produktions-IT-Landschaft (S2)* eines Unternehmens vorzusehen. Durch maschinenlesbare Schnittstellen und entsprechende Datenmodelle wird das System mit existierenden IT-Systemen zum Management von Wertschöpfungsnetzwerken und der PPS kombinierbar. Abgelöst werden diese durch ein System zur Reaktion auf Ereignisse aus dem Wertschöpfungsnetzwerk nicht, jedoch um weitere Funktionalität ergänzt. Darüber hinaus wird so die Möglichkeit geschaffen, die lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung schrittweise einzuführen (vgl. Abschnitt 2.1.4; Abschnitt 2.2.4).

4.2 Einordnung bestehender Ansätze

Auf Grundlage der definierten Anforderungen (vgl. Abschnitt 4.1) werden die bei der Analyse des Stands der Wissenschaft und Technik identifizierten Ansätze (vgl. Abschnitt 3) bewertet. Dies umfasst die Ansätze zur Reaktion auf Ereignisse aus dem Wertschöpfungsnetzwerk (vgl. Abschnitt 4.2.1), zur Ereignisverarbeitung in der Produktionssteuerung (vgl. Abschnitt 4.2.2) und zur Nutzung von bestärkendem Lernen in der Produktionssteuerung der Werkstattfertigung (vgl. Abschnitt 4.2.3).

4.2.1 Reaktion auf Ereignisse aus Wertschöpfungsnetzwerken

Bei der Bewertung der Ansätze zur Reaktion auf Ereignisse aus Wertschöpfungsnetzwerken mit Fokus auf die Ereignisverarbeitung (vgl. Tabelle 4.2) zeigt sich, dass diese zwar Ereignisse verarbeiten und mit Ausnahme von Kurbel et al. (2009) auch teilweise adaptiv sind, der Bezug zur Produktion und insbesondere zur PPS ist jedoch nicht gegeben. Auch eine Lernfähigkeit ist nur bei den Ansätzen von Giannakis & Louis (2016) und Behdani (2013) teilweise zu erkennen. Wenngleich die Geschwindigkeit des Systems nur bei drei Ansätzen explizit berücksichtigt wird, erreichen die Ansätze dennoch bis auf drei Ausnahmen mindestens eine teilweise Automatisierung, oftmals in Form eines Entscheidungsunterstützungssystems. Die Integration mit einer bestehenden Produktions-IT-Landschaft ist zumindest bei vier Ansätzen in Teilen gegeben, meist fokussiert auf das Enterprise Resource Planning (ERP)-System des Unternehmens.

Die Ansätze zur Reaktion auf Ereignisse aus Wertschöpfungsnetzwerken mit Fokus auf die Integration der Reaktion in die PPS zeigen genau an jener Stelle ihre Stärken in Form der Berücksichtigung externer Ereignisse in der Produktionssteuerung und dem Aktionsraum zur Einwirkung auf diskrete Aufträge (vgl. Tabelle 4.3). Damit wird über alle Ansätze hinweg zumindest eine teilweise Integration von Wertschöpfungsnetzwerk und Produktionssteuerung erreicht. Einzig der Ansatz nach Genc et al. (2014) unterscheidet sich meist als Ausnahme von den übrigen Ansätzen. Wenngleich die Ansätze überwiegend eine gewisse Adaptivität aufweisen, ist eine teilweise Lernfähigkeit nur bei den Ansätzen von Eirinakis et al. (2021) und Álvarez & Díaz (2011) gegeben. Der Automatisierungsgrad ist meist hoch bis sehr hoch und geht damit über reine Entscheidungsunterstützungssysteme hinaus. Mit einer Ausnahme ist eine Integration mit einem oder mehreren Produktions-IT-Systemen gegeben.

Autor	Anforderungen
A1	Integration von Wertschöpfungsnetzwerk und Produktionssteuerung
A2	Berücksichtigung des Fertigungsprinzips der komplexen Werkstattfertigung
A3	Granularität des relevanten Betrachtungsraums
E1	Berücksichtigung externer Ereignisse in der Produktionssteuerung
E2	Aktionsraum zur Einwirkung auf diskrete Aufträge
E3	Adaptivität des Entscheidungsmodells
E4	Lernfähigkeit des Entscheidungsmodells
E5	Geschwindigkeit des Entscheidungsmodells
S1	Automatisierungsgrad des Systems

Ansätze zur Reaktion auf Ereignisse aus Wertschöpfungsnetzwerken mit Fokus auf die Integration der Reaktion in die PPS										
Eirinakis et al. (2021)										
Genc et al. (2014)										
Guarnaschelli et al. (2013)										
Heinecke (2013)										
Lepratti et al. (2012)										
Lamparter et al. (2011)										
Álvarez & Díaz (2011)										
Legende: nicht erfüllt teilweise erfüllt erfüllt										

4.2.2 Ereignisverarbeitung in der Produktionssteuerung

Bei der Bewertung der Ansätze zur Ereignisverarbeitung in der Produktionssteuerung mit Fokus auf die Architektur zur Ereignisverarbeitung (vgl. Tabelle 4.4) zeigt sich, dass zumindest neuere Ansätze einen überwiegenden Teil der Anforderungen zumindest teilweise erfüllen, insbesondere der Ansatz von Pielmeier (2020). Yao et al. (2018) und Kasakow et al. (2016) berücksichtigen das Fertigungsprinzip der Werkstattfertigung. Pielmeier (2020) und Yao et al. (2018) beziehen in Teilen externe Ereignisse in die Produktionssteuerung mit ein. Jedoch sind nur drei Ansätze adaptiv gestaltet und einzig der von Pielmeier (2020) weist eine gewisse Lernfähigkeit auf. Die Geschwindigkeit des Systems wird nur bei Pielmeier (2020) und Metz (2014) explizit berücksichtigt. Mit Ausnahme von Kropp & Schuh (2014) weisen die Ansätze zumindest in Teilen eine Automatisierung und eine Integration mit weiteren Systemen der Produktions-IT-Landschaft auf. Eine Sonderrolle nimmt der Ansatz von Kropp & Schuh (2014) dadurch ein, dass der Schwerpunkt insbesondere auf der Modellierung von Ereignissen und deren Abhängigkeiten zu Steuergrößen der Produktionssteuerung liegt.

Aus der Bewertung der Ansätze zur Ereignisverarbeitung in der Produktionssteuerung mit Fokus auf die Anpassung des Ablaufplans wird ersichtlich, dass diese ihren Schwerpunkt im Aktionsraum zur Einwirkung auf diskrete Aufträge und zumindest in Teilen auch auf der Abbildung der Granularität des relevanten Betrachtungsraums haben (vgl. Tabelle 4.5). Mit Ausnahme der Ansätze von Zhu et al. (2018) und Kalinowski et al. (2014) steht dabei auch die Werkstattfertigung im Vordergrund. Dabei sind bis auf Pfitzer et al. (2018) alle Ansätze teilweise adaptiv gestaltet, Ansätze einer Lernfähigkeit zeigen jedoch nur Wang et al. (2020) und Tian et al. (2019). Die Geschwindigkeit spielt dabei nur bei drei Ansätzen eine Rolle, die Integration mit der bestehenden Produktions-IT-Landschaft nur bei Tian et al. (2019) und Niehues (2017). Einen gewissen Automatisierungsgrad weisen hingegen alle Ansätze auf.

4.2.3 Nutzung von bestärkendem Lernen in der Produktionssteuerung der Werkstattfertigung

Die Stärken der Ansätze zur Nutzung von bestärkendem Lernen in der Produktionssteuerung der Werkstattfertigung liegen auf deren Aktionsraum zur Einwirkung auf diskrete Aufträge sowie der ausgeprägten Adaptivität, Lernfähigkeit und des hohen Automatisierungsgrads (vgl. Tabelle 4.6). Darüber hinaus wurden vier Ansätze sogar explizit für die komplexe Werkstattfertigung entwickelt. Eine Integration von Wertschöpfungsnetzwerk und Produktionssteuerung sehen jedoch nur Luo (2020) und Lee et al. (2019) ansatzweise vor, externe Ereignisse werden nur bei Waschneck (2020) in Teilen berücksichtigt. Die Geschwindigkeit spielt zumindest bei elf Ansätzen eine Rolle, mit der Integration in die Produktions-IT-Landschaft befassen sich nur Zhou et al. (2021a), Kuhnle et al. (2021) und Waschneck (2020).

4.3 Zwischenfazit und Handlungsbedarf

Im zurückliegenden Abschnitt wurden Anforderungen an eine lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung definiert (vgl. Abschnitt 4.1). Diese gliedern sich in Anforderungen aus dem Anwendungszusammenhang (Anforderungen A1 – A3), an das Entscheidungsmodell (Anforderungen E1 – E5) und an die Systemarchitektur (Anforderungen S1 und S2). Damit konnte die zweite Teilforschungsfrage nach den Anforderungen an eine lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung beantwortet werden.

Auf diesen Anforderungen aufbauend wurden die im Stand der Wissenschaft und Technik identifizierten Ansätze methodisch bewertet und analysiert (vgl. Abschnitt 4.2). Dabei ist festzuhalten, dass die Ansätze interessante Aspekte aufweisen und auch Teile der Anforderungen berücksichtigen, jedoch nicht in der Lage sind, die definierte Forschungsfrage in allen Aspekten vollständig zu beantworten. Bei den Ansätzen zur Reaktion auf Ereignisse aus Wertschöpfungsnetzwerken liegen insbesondere Defizite hinsichtlich der Berücksichtigung des Fertigungsprinzips und der Lernfähigkeit, in Teilen auch insgesamt hinsichtlich eines Einbezugs der PPS, vor. Im Themenfeld der Ereignisverarbeitung in der Produktionssteuerung wiederum steht zwar die Produktionssteuerung im Fokus, es fehlt jedoch die Integration mit dem Wertschöpfungsnetzwerk. Auch eine Adaptivität und Lernfähigkeit der Entscheidungsmodelle ist hier meist nicht gegeben. Die Adaptivität und Lernfähigkeit stehen im Fokus der Ansätze zur Nutzung von bestärkendem Lernen in der Produktionssteuerung der Werkstattfertigung. Defizite weisen diesen Ansätze jedoch hinsichtlich der Integration mit dem Wertschöpfungsnetzwerk, der Berücksichtigung von Ereignissen in der Produktionssteuerung und der Integration in die Produktions-IT-Landschaft auf. Um die definierte Forschungsfrage in allen Aspekten vollständig zu beantworten, fehlt die Verbindung zwischen einer ereignisbasierten Integration von Wertschöpfungsnetzwerk und Produktionssteuerung sowie einem lernenden System zur darauf basierenden Optimierung der Produktionssteuerung in komplexen Werkstattfertigungen.

Ausgehend von diesen Erkenntnissen kann in Abschnitt 5 mit der Konzeption der lernenden ereignisbasierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung begonnen werden.

5 Konzeption der lernenden ereignisbasierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung

Aufbauend auf den definierten Anforderungen (vgl. Abschnitt 4) folgt in diesem Abschnitt mit der Konzeption der lernenden ereignisbasierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung der Kern der Arbeit. Hierfür wird die lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung mit ihren Artefakten strukturiert und detailliert beschrieben. Die anschließende Implementierung als Softwareartefakt mit darauf aufbauender Applikation folgt in Abschnitt 6.

Für die Konzeption der lernenden ereignisbasierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung wird zuerst das methodische Vorgehen dargestellt (vgl. Abschnitt 5.1). Anschließend folgen die Definition des Systemkonzepts (vgl. Abschnitt 5.2) und dessen Detailbeschreibung (vgl. Abschnitt 5.3). Die nachfolgenden Abschnitte bauen auf den Ausführungen in Bauer et al. (2021a), Bauer et al. (2020a) und Bauer et al. (2020b) auf.

5.1 Methodisches Vorgehen

Für die Konzeption der lernenden ereignisbasierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung wird methodisch anhand des in Abschnitt 1.4 dargestellten Information Systems Research Framework vorgegangen. Im Fokus steht dabei der in Abbildung 5.1 dargestellte, iterativ durchlaufene Gestaltungszyklus, in welchem die Artefakte konzeptioniert, implementiert und validiert werden. Über den angrenzenden

Relevanzzyklus fließen die Anforderungen ein (vgl. Abschnitt 4), über den Stringenzzyklus Grundlagen und Methoden der Wissensbasis (vgl. Abschnitt 2).

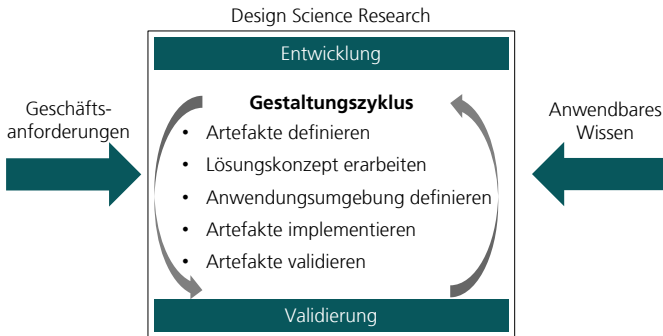


Abbildung 5.1: Gestaltungszyklus für die lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung

Der erste Schritt des Gestaltungszyklus umfasst die Definition der Artefakte (vgl. Abschnitt 5.2). Diese bilden die Basis für die darauf aufbauenden Entwicklungsarbeiten des Lösungskonzepts. Das Lösungskonzept beschreibt die Struktur, die Wechselwirkungen und die Funktionsweise der Artefakte (vgl. Abschnitt 5.3).

Anschließend wird eine Anwendungsumgebung definiert und charakterisiert (vgl. Abschnitt 6.1). Diese stellt die Rahmenbedingungen für die Implementierung und Validierung dar. Die Implementierung der Artefakte und damit des Lösungskonzepts umfasst die Umsetzung als Software (vgl. Abschnitt 6.2). Bei der Validierung werden die Funktionsfähigkeit und Eignung für den Anwendungszweck geprüft (vgl. Abschnitt 6.3).

Abhängig von den Ergebnissen der Validierung wird jeweils entschieden, ob alle Anforderungen erfüllt sind oder der Gestaltungszyklus erneut durchlaufen wird. Die nachfolgenden Abschnitte zeigen die Ergebnisse nach allen iterativen Durchläufen des Gestaltungszyklus.

5.2 Definition des Systemkonzepts

Ausgehend von der Problemstellung (vgl. Abschnitt 1.2), dem Handlungsbedarf und den Anforderungen (vgl. Abschnitt 2.4; Abschnitt 4.3) verfolgt die lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung das Ziel,

die Produktionssteuerung in komplexen Werkstattfertigungen bei Ereignissen aus dem Wertschöpfungsnetzwerk in Form eines lernenden Systems zu optimieren. Die hierfür erforderlichen Artefakte sind, wie in Abbildung 5.2 dargestellt, der *Regelkreis*, die *Methode* und die *Architektur* der lernenden ereignisbasierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung.

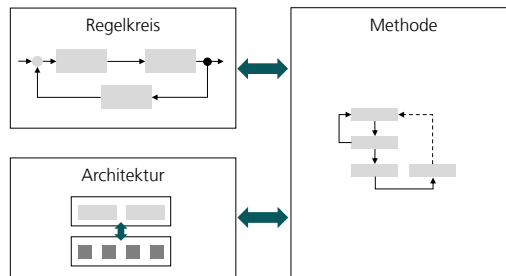


Abbildung 5.2: Artefakte der lernenden ereignisbasierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung

Der Zweck der lernenden ereignisbasierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung liegt in der Verbesserung der Liefertermineinhaltung bei im Wertschöpfungsnetzwerk auftretenden Ereignissen. Hierfür müssen im Falle eines Ereignisses Entscheidungen hinsichtlich Korrekturmaßnahmen für eine übergeordnete Priorisierung von Aufträgen getroffen werden, welche anschließend von der Produktionssteuerung an den einzelnen Maschinengruppen umgesetzt werden. Logisch abgebildet wird dies im Regelkreis, dessen Entwicklung und Anwendung durch die Schritte der Methode beschrieben wird. Die Architektur bildet die Grundlage für die Umsetzung in einem IT-System.

Der *Regelkreis* (vgl. Abschnitt 5.3.1) beschreibt das logische Zielbild zur Integration von Wertschöpfungsnetzwerk und Produktionssteuerung für die Mitigation der Auswirkungen von Ereignissen aus dem Wertschöpfungsnetzwerk in einem Produktionssystem. Hierfür werden Wertschöpfungsnetzwerk und Produktionssteuerung nicht länger nur über Planungsobjekte verknüpft, sondern direkt über Ereignisse. Ereignisse aus dem Wertschöpfungsnetzwerk werden dabei als Störgröße betrachtet, der Produktionsplan als Führungsgröße und die erwarteten Liefertermine der Aufträge als Regelgröße. Der zugrunde liegende Regelkreis ist als Abtastregelung, also mit einer Größenerfassung in festgelegten Zeitab-

ständen, und adaptiv, also veränderlich in Abhängigkeit der Betriebszustände, gestaltet. Der Regler basiert auf dem ML-Ansatz des bestärkenden Lernens. Die resultierende Aktion des Reglers, welche in Form der höchsten zu erwartenden Belohnung ausgewählt wird, stellt die Stellgröße dar.

Die *Methode* (vgl. Abschnitt 5.3.2) beschreibt in diskreten Schritten die Gestaltung eines solchen Regelkreises. Dabei ist die Methode in drei Phasen gegliedert: Design, Training und Applikation. In der Design-Phase werden das Problem und die Zielsetzung mathematisch beschrieben sowie Voraussetzungen geschaffen, um die Methode auf einen konkreten Anwendungsfall applizieren zu können. Im Rahmen des Trainings wird aufbauend auf dem Design das Entscheidungsmodell für den Anwendungsfall gelernt. Die Phase der Applikation beschreibt die Überführung des trainierten Entscheidungsmodells in den Produktivbetrieb. Die Schritte der Methode beinhalten dabei an diversen Stellen Startpunkte für die nächste Iteration, sodass die Schritte und Phasen der Methode erneut durchlaufen werden. Das Durchlaufen der Schritte der Methode bildet auch die Grundlage für die Skalierbarkeit hinsichtlich des Anwendungsfalls – es können dadurch jederzeit Zielgrößen und Einflussfaktoren beeinflusst, hinzugefügt oder entfernt werden.

Die Einbettung in eine bestehende Produktions-IT-Landschaft eines Unternehmens beschreibt die *Architektur* (vgl. Abschnitt 5.3.3). Sie zeigt einerseits die Komponenten der lernenden ereignisbasierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung und andererseits deren Verknüpfung mit IT-Systemen für das Management von Wertschöpfungsnetzwerken und der PPS. Die Architektur ist modular aufgebaut, sodass es möglich ist, die lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung schrittweise einzuführen, die für den Anwendungsfall notwendigen Komponenten auszuwählen und mit bestehenden IT-Systemen zu kombinieren. Des Weiteren ist die Architektur durch standardisierte Schnittstellen und definierte Datenmodelle offen gestaltet. Dies erlaubt es einerseits, zusätzliche Elemente des Produktionssystems einzubinden und andererseits, kompatible Schnittstellen in kommerzielle IT-Systeme zu integrieren. In potenziellen Anwendungen können durch die offene Architektur Vendor-Lock-in-Effekte vermieden werden.

5.3 Detailbeschreibung des Systemkonzepts

Wie in Abschnitt 5.2 dargestellt, besteht die lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung aus dem Regelkreis (vgl. Abschnitt 5.3.1), der Methode (vgl. Abschnitt 5.3.2) und der Architektur (vgl. Abschnitt 5.3.3), welche nachfolgend detailliert beschrieben werden.

5.3.1 Regelkreis

Die Interaktion von Produktionssteuerung und Produktion lässt sich als Regelkreis beschreiben. Hierfür existieren diverse etablierte Ansätze, welche sich anhand ihres Reglertyps unterscheiden (vgl. Abschnitt 2.3.4). Bezogen auf die Fertigung kommt dem Regelkreis zwischen Produktionssteuerung und Produktion die Aufgabe zu, mit Störungen in Form von Ereignissen aus der Produktion umzugehen¹. Des Weiteren existieren diverse Ansätze, um basierend auf Ereignissen eine Anpassung des Ablaufplans durch die Produktionssteuerung vorzunehmen (vgl. Abschnitt 3.3). Obwohl die Formulierung als Regelkreis hinsichtlich der Automatisierung von Entscheidungen sinnvoll ist, ist den Ansätzen aus der Literatur gemein, dass Ereignisse aus dem Wertschöpfungsnetzwerk nicht oder nur marginal berücksichtigt werden. Eine direkte Verknüpfung zwischen Wertschöpfungsnetzwerk und Produktionssteuerung, welche über den Austausch von Planungsobjekten hinausgeht, ist nicht gegeben.

Diese funktionale Lücke wird über den Aufbau eines Regelkreises als Adaptiver Regelkreis für ein integriertes Wertschöpfungsnetzwerk (ARIW), dem Zielbild für die Umsetzung einer lernenden ereignisbasierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung, geschlossen. Der ARIW ist ein diskreter, ereignisbasierter, adaptiver und lernender Regelkreis, der das Wertschöpfungsnetzwerk mit der Produktion integriert, konkreter das Management des Wertschöpfungsnetzwerks mit der Produktionssteuerung. Er zielt darauf ab, die Folgen von Ereignissen, die durch das Wertschöpfungsnetzwerk in der Produktion verursacht werden, zu mitigieren. Aufgebaut ist der ARIW analog zu bekannten

¹Auf die Fertigung entfallen insbesondere in komplexen Werkstattfertigungen meist ein Großteil der Durchlaufzeit und ein großer Kostenanteil, weshalb sich die Untersuchung im Rahmen dieser Arbeit auf die Fertigung fokussiert (vgl. Abschnitt 2.1.1). Aufgrund der Konsistenz zur Literatur wird dennoch weiterhin von Produktionssteuerung und Produktion gesprochen, auch wenn die Montage nicht betrachtet wird.

Regelkreisen aus Regeleinrichtung, Regelstrecke und Messeinrichtung (vgl. Abbildung 5.3). Die Regeleinrichtung, der sogenannte ARIW-Regler, umfasst die Maßnahmenauswahl als eigentliche Entscheidung und die nachgelagerte, darauf basierende, Plananpassung. Produktionssteuerung und Produktion bilden die Regelstrecke, BDE und MDE die Messeinrichtung. Ereignisse aus dem Wertschöpfungsnetzwerk werden als Störgröße explizit berücksichtigt und in die Entscheidungen des Reglers einbezogen (vgl. Anforderung A1; Anforderung E1).

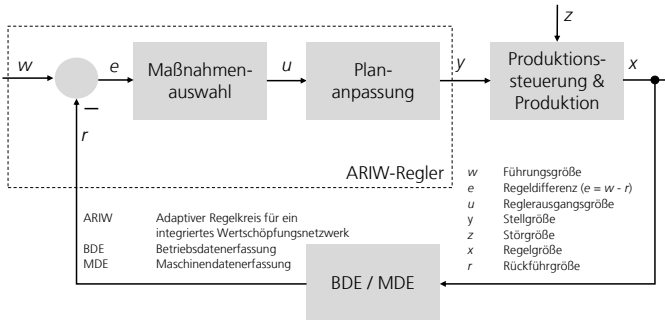


Abbildung 5.3: Der adaptive Regelkreis für ein integriertes Wertschöpfungsnetzwerk

Der ARIW (vgl. Abbildung 5.4, roter Regelkreis) komplementiert damit die bestehende Produktionsregelung hinsichtlich interner Ereignisse aus der Produktion (vgl. Abbildung 5.4, grüner Regelkreis) mit der Verarbeitung von Ereignissen aus dem Wertschöpfungsnetzwerk. Zwar nutzen beide Regelkreise den Produktionsplan als Führungsgröße w , sie unterscheiden sich jedoch in den weiteren Größen. Als Stellgröße y passt der ARIW diesen Produktionsplan für einzelne Aufträge in der Produktion, basierend auf externen Ereignissen, einwirkend in Form der Störgröße z , an (vgl. Anforderung E2). Damit wird fortlaufend auf die Regelgröße x der Liefertermine eingewirkt, um die Liefertermineinhaltung gegenüber dem Wertschöpfungsnetzwerk trotz der Ereignisse möglichst aufrechtzuerhalten, was insbesondere in der komplexen Werkstattfertigung eine Herausforderung darstellt (vgl. Anforderung A1; Anforderung A3). Abgesehen von eventuell auftretenden Messverzerrungen durch BDE und MDE entspricht die Rückführgröße r auch der Regelgröße x .

Die Integration des ARIW-Reglers in die PPS ist sinnvoll und zielführend, da bislang weder die Produktionsplanung, noch die Produktionssteuerung in der Lage sind, Ereignisse aus dem Wertschöpfungsnetzwerk in der notwendigen Form zu verarbeiten (vgl. Anforderung

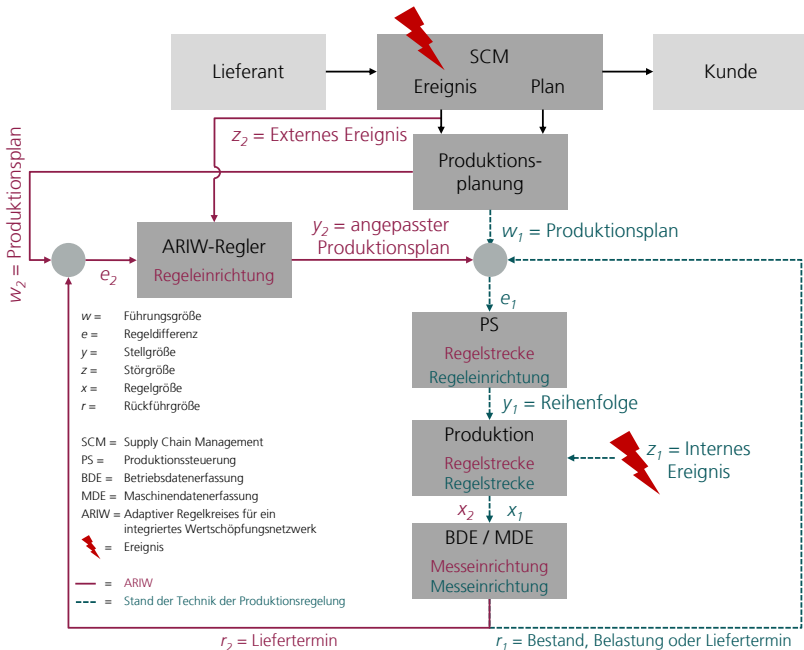


Abbildung 5.4: Der adaptive Regelkreis für ein integriertes Wertschöpfungsnetzwerk eingebettet in die PPS (eigene Darstellung in Anlehnung an Bauer et al. 2020b, S. 94)

rung A1; Anforderung E1). Die Produktionsplanung arbeitet hierfür in einer zu langsamen Frequenz bzw. mit zu großen Planungsperioden. Des Weiteren liegen Daten meist nicht ausreichend feingranular vor, um Entscheidungen über Korrekturmaßnahmen zu treffen. Insbesondere geht jedoch zum Zeitpunkt der Auftragsfreigabe die Ablaufsteuerung auf die Produktionssteuerung über, sodass sich bereits in der Produktion befindliche Lose nicht mehr von der Produktionsplanung angepasst werden. Damit ist eine Mitigation von Ereignissen aus dem Wertschöpfungsnetzwerk, welche bereits gestartete Lose betreffen, nicht möglich. Der Produktionssteuerung hingegen fehlt bislang meist eine direkte Verknüpfung zum Wertschöpfungsnetzwerk, diese ist nur über Planungsobjekte vorgesehen. Des Weiteren sind die Entscheidungsmodelle der Produktionssteuerung oftmals nicht lernfähig, sodass keine Anpassung an sich stetig verändernde Situationen möglich ist. Auch durch die Produktionssteuerung ist somit keine effektive und effiziente Mitigation von Ereignissen aus

dem Wertschöpfungsnetzwerk möglich. Der ARIW-Regler schließt genau diese Lücke und ermöglicht eine automatisierte situationsabhängige Entscheidung über Korrekturmaßnahmen zur Minimierung der Konsequenzen von Ereignissen aus dem Wertschöpfungsnetzwerk. Die Gestaltung als separate Komponente, welche nicht in die Produktionsplanung und die Produktionssteuerung integriert ist, ermöglicht dabei eine Ergänzung bestehender PPS-Architekturen (vgl. Abschnitt 5.3.3). Wichtig ist, dass die Entscheidungen des ARIW über Korrekturmaßnahmen auch in die Produktionsplanung rückgekoppelt werden, damit keine gegenläufigen Anpassungen entstehen.

Die Handlungsalternativen des ARIW-Reglers ergeben sich aus der Charakteristik der Werkstattfertigung und insbesondere der komplexen Werkstattfertigung (vgl. Anforderung A2). Zum Ersten aus dem Vorteil der Werkstattfertigung mit ihrer flexiblen Anpassung an unterschiedliche Aufträge und deren Bearbeitungsreihenfolge. Aufträge, die vor einzelnen Maschinen warten, können jederzeit in ihrer Reihenfolge getauscht werden. Zum Zweiten ergibt sich zusätzliche Flexibilität aus den aus Parallelmaschinen bestehenden Maschinengruppen und den wiedereintretenden Prozessflüssen der komplexen Werkstattfertigung (vgl. Abschnitt 2.1.1). Selbst die lange Durchlaufzeit als vermeintlicher Nachteil der komplexen Werkstattfertigung wird hier genutzt, da somit dem ARIW-Regler während des Durchlaufs eines Auftrags durch die Fertigung auch ein entsprechender Zeitraum zur Verfügung steht, damit Maßnahmen nicht nur umgesetzt werden, sondern diese auch ihre Wirkung entfalten können (vgl. Abschnitt 2.2.2).

Durch den Einsatz von RL wird der ARIW adaptiv und lernend (vgl. Anforderung E3; Anforderung E4). Dem ARIW-Regler werden weder ein regelungstechnisches Modell der Regelstrecke, noch gelabelte Daten oder Strategien für die Entscheidung über Aktionen vorgegeben (vgl. Abschnitt 2.3.2). Der RL-basierte Agent lernt die Strategie selbständig und datenbasiert aus der Beobachtung seiner Umgebung, in diesem Fall der Auswirkungen seiner Entscheidungen auf die erwarteten Liefertermine in der Produktion (vgl. Anforderung S1). Das Lernen funktioniert dabei über Versuch und Irrtum, was sich für die komplexe Werkstattfertigung dahingehend gut eignet, dass hier keine mathematisch exakte Lösung der Ablaufplanung möglich ist und Versuch und Irrtum gute Annäherungen ermöglicht. Auch die Möglichkeit des Umgangs von RL mit verzögerten Belohnungen spiegelt sich in der komplexen Werkstattfertigung wider – die finale Liefertermineinhaltung (oder -

nichteinhaltung) zeigt sich erst, wenn der Durchlauf eines Auftrags durch die Produktion abgeschlossen ist. Diese Eigenschaften in Kombination mit der mit RL einhergehenden Fähigkeit zur Generalisierung auf unbekannte Daten eignen sich deshalb sehr gut, um die Dynamik der komplexen Werkstattfertigung im Entscheidungsmodell des ARIW-Reglers abzubilden.

Voraussetzungen für den Einsatz von RL sind die Modellierung als Markov-Entscheidungsprozess und in der komplexen Werkstattfertigung die Nutzung eines Simulationsmodells. Der Markov-Entscheidungsprozess beschreibt Zustände, Aktionen und Belohnungen für den Agenten. Damit kommt dieser Modellierung eine besondere Bedeutung zu, denn der Agent wird sich anschließend so verhalten, dass er unter diesen Rahmenbedingungen maximal belohnt wird. Das Simulationsmodell ist notwendig, da für das Training des Agenten eine sehr große Anzahl an Iterationen benötigt wird, bevor dieser zufriedenstellende Ergebnisse liefert. Das Training kann deshalb nicht beim Einsatz in einem realen Produktionssystem durchgeführt werden. Nach Abschluss des Trainings ist der Agent jedoch in der Lage, Entscheidungen für das reale Produktionssystem in hoher Geschwindigkeit zu treffen und über den Regelkreis umzusetzen (vgl. Anforderung E5). Diverse Quellen bezeichnen die Entscheidungsgeschwindigkeit eines RL-Agenten deshalb auch als Echtzeit (Kuhnle 2020, S. 54; Waschneck 2020, S. 76), der Begriff soll aufgrund der nicht eindeutigen Definition in dieser Arbeit jedoch nicht weiter verwendet werden. Während der Anwendung passt sich der Agent dabei kontinuierlich an Veränderungen an. Die Schaffung der Voraussetzungen sowie die Gestaltung und die Anwendung des ARIW zeigt die Methode im nachfolgenden Abschnitt 5.3.2.

5.3.2 Methode

Die Methode zur Gestaltung des ARIW beschreibt in insgesamt 16 Schritten und drei Phasen das Design, das Training und die Applikation dieses Regelkreises (vgl. Abbildung 5.5). Sie ermöglicht damit den Aufbau und die Anwendung einer lernenden ereignisbasierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung zur Mitigation von Ereignissen aus dem Wertschöpfungsnetzwerk auf der Ebene der Produktionssteuerung. Die Schritte der Methode bieten an diversen Stellen Startpunkte für die nächste Iteration,

sodass einzelne Schritte und gesamte Phasen der Methode erneut durchlaufen werden können. Dadurch können jederzeit Zielgrößen und Einflussfaktoren hinzugefügt oder entfernt werden, um den ARIW hinsichtlich des Anwendungsfalls und seiner Fortentwicklung anzupassen. Des Weiteren sind, abhängig vom Anwendungsfall, gegebenenfalls einzelne Schritte der Methode oder Teile davon bereits bekannt und in einem Unternehmen umgesetzt, sodass diese übersprungen und deren Ergebnisse in die folgenden Schritte der Methode integriert werden können.

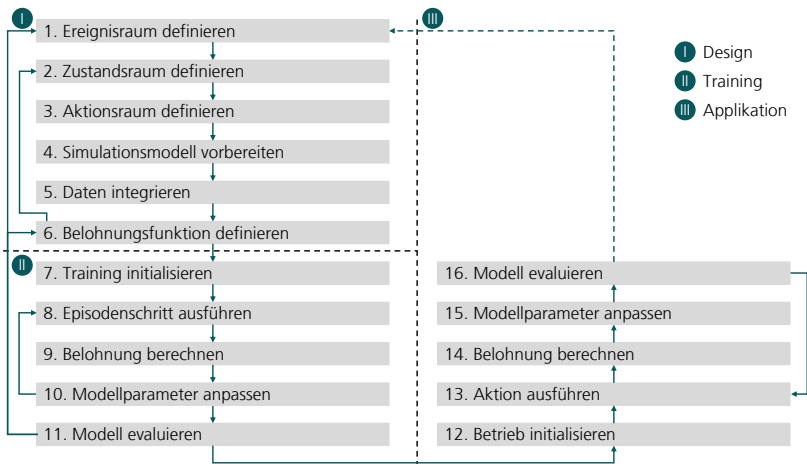


Abbildung 5.5: Methode zur Gestaltung des adaptiven Regelkreises für ein integriertes Wertschöpfungsnetzwerk

Das Design (vgl. Abbildung 5.5, Phase I) dient der Schaffung von Voraussetzungen, um die Methode für die Anwendung in einem konkreten Anwendungsfall anzupassen. Dazu gehören insbesondere die Problemdefinition, deren mathematische Formulierung und die Vorbereitung der IT-Umsetzung. Als zentrale Herausforderung ist das Design der Belohnungsfunktion anzusehen. Davon hängen die Trainingsdauer und insbesondere der Erfolg in der Applikation des Entscheidungsmodells ab.

Während des Trainings (vgl. Abbildung 5.5, Phase II) lernt der Agent ein Entscheidungsmodell für den Anwendungsfall. Hierfür sind initiale Parameter zu definieren, eine Anzahl von Trainingsläufen zu absolvieren sowie die Ergebnisse zu evaluieren. Eine besondere Bedeutung kommt dabei der Definition und Variation der Parameter zu, da diese einen weit-

reichenden Einfluss auf das resultierende Entscheidungsmodell und Ergebnis haben können. Wird das Entscheidungsmodell positiv evaluiert, kann es anschließend in die Applikation überführt werden.

In der Applikation (vgl. Abbildung 5.5, Phase III) wird das trainierte Entscheidungsmodell in den Produktivbetrieb für ein Produktionssystem überführt. Der Agent läuft dann kontinuierlich parallel zur Produktion mit dem Ziel, die Liefertermineinhaltung gegenüber dem Wertschöpfungsnetzwerk bei auftretenden externen Ereignissen zu verbessern. Bei signifikanten Änderungen im Produktionssystem, die ein Neutraining des Entscheidungsmodells notwendig machen, können die Phasen des Designs und des Trainings erneut durchlaufen werden.

In allen drei Phasen der Methode werden Entscheidungen getroffen, die einen signifikanten Einfluss auf die Ergebnisse haben können. Für die Reproduzierbarkeit der Ergebnisse ist deshalb die Dokumentation der Modellparameter und Einflussfaktoren notwendig (Henderson et al. 2018, S. 3207). Nachfolgend werden, gegliedert nach den Phasen Design (vgl. Abschnitt 5.3.2.1), Training (vgl. Abschnitt 5.3.2.2) und Applikation (vgl. Abschnitt 5.3.2.3), die Schritte der Methode im Detail beschrieben. Die Beschreibung jedes Schritts beinhaltet dabei eine Übersichtstabelle mit dem Ziel des Schritts sowie den Eingaben und Ausgaben hinsichtlich Daten und Wissen. Entspricht eine Eingabe dabei einem nachfolgenden Schritt, so kommt diese nicht beim erstmaligen Durchlaufen des Schritts, sondern erst bei weiteren iterativen Durchläufen des Schritts, zum Tragen. Des Weiteren enthält jede Übersichtstabelle auch eine Kurzdarstellung der Teilschritte, mit welchen unter Berücksichtigung der Eingaben das Ziel des Schritts erreicht wird.

5.3.2.1 Phase I: Design

Das Design besteht aus der Definition von Ereignis-, Zustands- und Aktionsraum, der Vorbereitung des Simulationsmodells, der Integration von Daten und der Definition der Belohnungsfunktion (vgl. Abbildung 5.6). Diese sechs Schritte werden nachfolgend eingehend beschrieben. Ergebnis des Designs ist eine vorbereitete Umgebung mit vollständig formuliertem Markov-Entscheidungsprozess als Grundlage für die Interaktion mit dem Agenten im anschließenden Training.

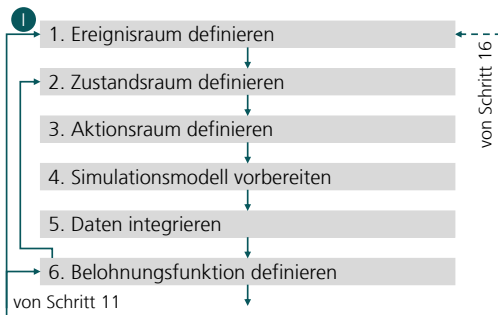


Abbildung 5.6: Phase I: Design der Methode zur Gestaltung des adaptiven Regelkreises für ein integriertes Wertschöpfungsnetzwerk

Schritt 1: Ereignisraum definieren

Der erste Schritt befasst sich mit der Festlegung relevanter Ereignistypen, auf die mit dem Entscheidungsmodell reagiert werden soll (vgl. Tabelle 5.1). Hierfür ist es notwendig, mögliche Ereignistypen zu identifizieren, anschließend die relevanten Ereignistypen zu definieren, mathematisch zu formulieren und Strategien zur Ereigniserkennung festzulegen. Die Teilschritte können erneut durchlaufen werden, wenn die Zielsetzung anhand der Evaluation nicht erreicht wurde. Grundlage für die Anpassung stellt dann die detaillierte Analyse der Modellevaluation aus Schritt 11 bzw. Schritt 16 dar.

Tabelle 5.1: Schritt 1: Ereignisraum definieren

Ziel	Festlegen der Ereignistypen, auf die mit dem Entscheidungsmodell automatisiert reagiert werden soll und wie diese im Betrieb erkannt werden.
Eingabe (Daten)	–
Ausgabe (Daten)	Ereignisraum E (zu Schritt 2, 4)
Eingabe (Wissen)	<i>Opt: Detaillierte Analyse der Modellevaluation (von Schritt 11)</i> <i>Opt: Detaillierte Analyse der Modellevaluation (von Schritt 16)</i>
Ausgabe (Wissen)	Strategie zur Erkennung relevanter Ereignisse im Betrieb (zu Schritt 5)
Teilschritte	Ereignistypen identifizieren → Relevante Ereignistypen definieren → Ereignisraum mathematisch formulieren → Strategien zur Ereigniserkennung definieren

Bei der Identifikation von Ereignistypen geht es darum, die Menge der potenziell auftretenden Ereignistypen im Wertschöpfungsnetzwerk festzulegen. Dieser Prozess kann mittels

Erfahrungswissen, Methoden des ML oder einer Kombination beider Ansätze erfolgen. Für die Identifikation mittels Erfahrungswissen legen Experten die Ereignistypen manuell fest. Hierfür ist ein enormes Erfahrungswissen erforderlich, um trotz der Komplexität eines Wertschöpfungsnetzwerks gute Ergebnisse zu erzielen. Dafür ist das Verfahren jedoch schnell umsetzbar und erreicht durch den Einbezug relevanter Experten üblicherweise eine hohe Akzeptanz. Jedoch werden nur solche Ereignistypen identifiziert, die auch bislang bereits als Ereignisse betrachtet wurden. Liegen entsprechende Daten aus dem Wertschöpfungsnetzwerk vor, kann auch mit Methoden des ML, konkret des unüberwachten Lernens, gearbeitet werden. Durch eine Anomalieerkennung können dabei auch Ereignistypen identifiziert werden, welche bislang nicht betrachtet wurden. Eine Kombination beider Verfahren ist ebenfalls möglich (Steenwinckel et al. 2021, S. 36).

Im zweiten Teilschritt werden relevante Ereignistypen definiert, auf die mit dem Entscheidungsmodell reagiert werden soll - entsprechend ist dies eine Teilmenge der zuvor identifizierten potenziellen Ereignistypen. Ziel ist dabei nicht die Definition von Schwellwerten, bei deren Überschreiten auf das Ereignis reagiert werden soll. Das ist Teil des Entscheidungsmodells und wird in der Phase des Trainings gelernt. Bei der Definition relevanter Ereignistypen geht es darum, jene Ereignistypen festzulegen, für welche (noch) keine adäquate Reaktion besteht. Entsprechend ist hierfür eine Analyse der Geschäftsprozesse notwendig. In einem Beispiel, detailliert beschrieben in Bauer et al. (2021a, S. 13–14), können eingehende Kundenaufträge bereits automatisiert über die IT-Systeme des Wertschöpfungsnetzwerks und der PPS verarbeitet werden, Abweichungen zwischen der Prognose von Kundenaufträgen und dem tatsächlichen Bestellverhalten jedoch nur mit einem aufwändigen manuellen Prozess. Entsprechend ist in diesem Beispiel insbesondere der Ereignistyp einer Prognoseabweichung von Relevanz. Ereignisse sind dabei meist negativ konnotiert, können jedoch ebenso positiv sein und beispielsweise durch die Verschiebung eines Soll-Liefertermins in die Zukunft einen zusätzlichen Handlungsraum für die Produktionssteuerung ermöglichen.

Abhängig vom Anwendungsfall und der Anzahl der identifizierten Ereignistypen kann es sinnvoll sein, diese in Klassen einzuteilen. Im einfachsten Fall existieren dabei nur zwei Klassen: Ereignistypen, auf die reagiert werden soll und solche, auf die nicht reagiert werden soll. Eine weitere Aufteilung kann sinnvoll sein, um abhängig von der Ereignisklasse individuelle Strategien zur Erkennung festzulegen oder Vorverarbeitungsschritte auszuführen. Zwar

kann diese Einteilung in Klassen durch eine Clusteranalyse unterstützt werden, jedoch ist insbesondere hier das Erfahrungswissen von Experten notwendig. Klassen werden dabei meist nach ähnlichen Auswirkungen der Ereignistypen erstellt.

Angelehnt an die Begrifflichkeiten des Markov-Entscheidungsprozesses sollen die relevanten Ereignistypen, auf die mit dem Entscheidungsmodell reagiert werden soll, mathematisch als Ereignisraum E bezeichnet werden. Ein Ereignis zum Zeitpunkt t wird dann als $E_t \in E$ bezeichnet. Für Entscheidungen auf der Ebene der Produktionssteuerung ist es relevant, welcher Auftrag und damit welches Los mit welchem Produkt durch ein Ereignis betroffen ist sowie die Änderungen in Liefertermin und Stückzahl (vgl. Gleichung 5.1). Die Losnummer L ist dabei zur eindeutigen Identifikation eines Loses vorgesehen. Der Produkttyp P kann als sogenannter One-Hot-Vektor ² ausgedrückt werden: $P \in \{0, 1\}^{n_P}$ mit $n_P \in \mathbb{N}$. Damit wird auch festgelegt, dass jedes Los einen eindeutigen Produkttyp aufweist. Gleichzeitig entspricht der Produkttyp auch einem Arbeitsplan. Der geplante Liefertermin kann als relatives Maß über den Plan-Flussfaktor FF_{Plan} beschrieben werden, die vom betroffenen Auftrag umfasste Stückzahl als $n_{Stk} \in \mathbb{N}$. Deren Änderungen werden als $\Delta FF_{Plan} \in \mathbb{Z}$ und $\Delta n_{Stk} \in \mathbb{N}$ abgebildet.

$$E_t = (L, P, FF_{Plan}, n_{Stk}, \Delta FF_{Plan}, \Delta n_{Stk}) \quad (5.1)$$

Im letzten Teilschritt müssen Strategien zur Erkennung von Ereignissen aus dem Wertschöpfungsnetzwerk festgelegt werden. Grundsätzlich unterschieden werden können diese in reaktive und proaktive Ansätze. Für eine reaktive Erkennung von Ereignissen eignet sich die Definition von Schwellwerten, ab welchen ein Ereignis erkannt wird. Im obigen Beispiel ist dies die maximal tolerierte Abweichung zwischen Prognose und tatsächlichem Bestellverhalten. Das Vorgehen ist damit angelehnt an aus der Produktion bekannte Qualitätsregelkarten, deren Konzept bereits auf das Wertschöpfungsnetzwerk übertragen wurde (Bauer et al. 2019a, S. 37–38). Proaktive Ansätze ermöglichen in einem gewissen Rahmen die Prognose von Ereignissen vor deren Eintreten, um entsprechende Maßnahmen bereits frühzeitig initiieren zu können. Hierfür können Trendanalysen eingesetzt werden

²In einem One-Hot-Vektor sind Daten so kodiert, dass ein einziges Element den Wert 1 aufweisen darf, alle anderen wiesen den Wert 0 auf. Am Beispiel des Produkttyps P gilt dann $\sum_{i=1}^{n_P} P_i = 1$. Entsprechend eignen sich One-Hot-Vektoren für die Kodierung kategorialer Daten.

(Fernández et al. 2016, S. 31). Wurde im zweiten Teilschritt eine Klassifikation festgelegt, welche über die Unterscheidung in relevante und nicht relevante Ereignisse hinausgeht, ist nach der Erkennung noch eine Klassifikation der Ereignisse notwendig. Um diesen Prozess zu automatisieren, können Methoden des überwachten Lernens aus dem ML eingesetzt werden (Bauer et al. 2019b, S. 91–92).

Schritt 2: Zustandsraum definieren

Der zweite Schritt beinhaltet die Definition des Zustandsraums und die anschließende mathematische Formulierung als Grundlage für den Markov-Entscheidungsprozess (vgl. Tabelle 5.2). Konnte die Belohnungsfunktion in Schritt 6 mit den vorliegenden Informationen nicht gebildet werden, kann die Definition des Zustandsraums erneut durchlaufen werden.

Tabelle 5.2: Schritt 2: Zustandsraum definieren	
Ziel	Definition des Zustandsraums, welcher dem Entscheidungsmodell Daten über den Zustand der Umgebung bereitstellt.
Eingabe (Daten) Ausgabe (Daten)	Ereignisraum E (von Schritt 1) Zustandsraum S (zu Schritt 4)
Eingabe (Wissen) Ausgabe (Wissen)	<i>Opt: Fehlende Kenngrößen zur Bildung der Belohnungsfunktion (von Schritt 6)</i> –
Teilschritte	Kenngrößen des Zustandsraums identifizieren → Zustandsraum mathematisch formulieren

Problemstellungen der PPS können entweder als einzelner Markov-Entscheidungsprozess oder in Form eines Multi-Agentensystems als n Markov-Entscheidungsprozesse beschrieben werden (Gabel & Riedmiller 2007, S. 6). Diverse Ansätze aus dem Stand der Wissenschaft und Technik, beispielsweise May et al. (2021), Kuhnle (2020) und Waschneck (2020), fokussieren auf das Dispatching und bauen hierfür ein Multi-Agentensystem mit meist ebenso vielen Agenten wie betrachteten Maschinen auf (vgl. Abschnitt 3.4). Für die in dieser Arbeit vorliegende Fragestellung der Verbesserung der Liefertermineinhaltung gegenüber dem Wertschöpfungsnetzwerk bei auftretenden Ereignissen bietet sich jedoch aufgrund des übergeordneten Regelungscharakters ein System mit einem Agenten und damit ein einzelner Markov-Entscheidungsprozess an (vgl. Anforderung A3).

Der erste Teilschritt umfasst die Identifikation der relevanten Kenngrößen zur Bildung des Zustandsraums S . Hierbei ist die Markov-Eigenschaft zu berücksichtigen, die besagt, dass der neue Zustand S_{t+1} nur vom vorherigen Zustand S_t und der Aktion A_t abhängt (vgl. Abschnitt 2.3.2). Entsprechend müssen jene Kenngrößen ausgewählt werden, welche notwendig sind, damit die Markov-Eigenschaft zutrifft. Ebenso ist es für den Erfolg des RL-Agenten entscheidend, dass die bestmöglich geeignete Aktion zu einem Zeitpunkt aus den kodierten Informationen im Zustand ausgewählt werden kann. Für die Fragestellung zur Verbesserung der Liefertermineinhaltung gegenüber dem Wertschöpfungsnetzwerk bei auftretenden Ereignissen besteht der Zustandsraum des Agenten auf der Ebene der Produktionssteuerung aus den Zuständen aller Lose in der Fertigung (vgl. Abbildung 5.7). Der Zustand eines Loses wiederum kann beschrieben werden durch den Produkttyp, die Maschinengruppe, den Fertigstellungsgrad, die Differenz zwischen Plan-Flussfaktor und Freigabe-Flussfaktor, zwischen Plan-Flussfaktor und Ist-Flussfaktor sowie den Steuer-Flussfaktor. Bei der Definition des Aktionsraums in Schritt 3 wird erläutert, warum die Stückzahl nicht weiter betrachtet wird.

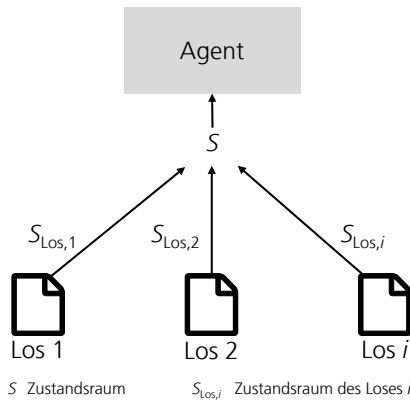


Abbildung 5.7: Zusammensetzung des Zustandsraums des Agenten

Im zweiten Teilschritt wird der Zustandsraum mathematisch formuliert. Der Zustandsraum S ergibt sich nach Gleichung 5.2 dabei aus der Kombination der Zustandsräume der einzelnen Lose S_{Los} .

$$S = (S_{Los,1}, S_{Los,2}, \dots, S_{Los,i}) \quad (5.2)$$

mit $i = n_L$ und

$$S_{Los} = \begin{pmatrix} P \\ MG \\ F \\ \Delta FF_{Plan-Freigabe} \\ \Delta FF_{Plan-Ist} \\ FF_{Steuer} \end{pmatrix} \quad (5.3)$$

Analog zum Ereignisraum wird auch für den Zustandsraum der Produkttyp P als One-Hot-Vektor ausgedrückt: $P \in \{0, 1\}^{n_P}$ mit $n_P \in \mathbb{N}$. Die Maschinengruppe MG wird ebenfalls als One-Hot-Vektor definiert zu $MG \in \{0, 1\}^{n_{MG}}$ mit $n_{MG} \in \mathbb{N}$. Anders ausgedrückt bedeutet das eine eindeutige Zuordnung eines Loses zu der Maschinengruppe, an welcher es sich derzeit befindet. Der Fertigstellungsgrad F ist ein Maß für den Bearbeitungsfortschritt eines Loses und berechnet sich anhand Gleichung 5.4 aus dem Verhältnis von bereits abgearbeiteten Durchführungszeiten zu der gesamten Durchführungszeit. Dadurch wird der Fertigstellungsgrad auch gleichzeitig normalisiert, was für robuste KNN von Vorteil ist (Goodfellow et al. 2016, S. 317).

$$F = \frac{\sum_{i=1}^{n_{OPa}} ZDF_i}{\sum_{i=1}^{n_{OP}} ZDF_i} \quad (5.4)$$

Ebenfalls normalisiert werden deshalb die Differenzen zwischen Plan-Flussfaktor und Freigabe-Flussfaktor, zwischen Plan-Flussfaktor und Ist-Flussfaktor sowie der Steuer-Flussfaktor. Der Freigabe-Flussfaktor $FF_{Freigabe}$ ist dabei jener Flussfaktor, welcher bei der Auftragsfreigabe vorgegeben wird. Er entspricht deshalb, ausgedrückt als relatives Maß, dem bei Auftragsfreigabe geplanten Liefertermin. Der aktuell geplante Liefertermin spiegelt sich im Plan-Flussfaktor FF_{Plan} wider. Ist das Los durch ein Ereignis betroffen, wird der Plan-Flussfaktor angepasst mit $FF_{Plan} = FF_{Plan} + \Delta FF_{Plan}$. Der Steuer-Flussfaktor FF_{Steuer} wird durch Aktionen des Agenten angepasst, um die Priorisierung von Losen zu verändern

(vgl. Schritt 3). Der Ist-Flussfaktor FF_{Ist} wiederum ist der gemessene Flussfaktor und lässt Rückschlüsse auf die Pünktlichkeit von Losen zu. Die Differenz zwischen Plan-Flussfaktor und Freigabe-Flussfaktor errechnet sich aus $\Delta FF_{Plan-Freigabe} = FF_{Plan} - FF_{Freigabe}$, jene zwischen Plan-Flussfaktor und Ist-Flussfaktor aus $\Delta FF_{Plan-Ist} = FF_{Plan} - FF_{Ist}$. Für negative $\Delta FF_{Plan-Ist}$, also $FF_{Ist} > FF_{Plan}$, ist das Los gegenüber seinem geplanten Liefertermin verspätet, für positive $\Delta FF_{Plan-Ist}$, also $FF_{Ist} < FF_{Plan}$, überpünktlich und für $\Delta FF_{Plan-Ist} = 0$, also $FF_{Ist} = FF_{Plan}$, pünktlich. Abgeleitet von Gleichung 2.2 kann die Liefertermineinhaltung LTE damit anhand Gleichung 5.5 definiert werden.

$$LTE = \frac{n_L \text{ mit } \Delta FF_{Plan-Ist} \geq 0}{n_L} \cdot 100 \quad (5.5)$$

Für KNN ist es wichtig, dass die Anzahl der Eingaben konstant gehalten wird. Entsprechend ist es notwendig, dass wenn die Anzahl der Lose n_L variiert und nicht beispielsweise über eine Conwip-Auftragsfreigabe (Lödding 2016, S. 373) konstant gehalten wird, leere Positionen im Zustandsvektor S mit Null aufgefüllt werden. Darüber hinaus ist es wichtig sicherzustellen, dass der Agent seine Entscheidungen nicht anhand der Position eines Loses im Zustandsvektor S lernt, sondern anhand der kodierten Kenngrößen bzw. den darin enthaltenen Daten. Hierfür ist es notwendig, dass der Zustandsvektor S nicht beispielsweise anhand der Losnummern oder eines anderen Ordnungskriteriums geordnet wird, sondern die Zustandsvektoren der Lose S_{Los} zufällig innerhalb des Zustandsvektors S positioniert werden.

Schritt 3: Aktionsraum definieren

Im dritten Schritt werden relevante Aktionen identifiziert und anschließend mathematisch als Aktionsraum formuliert (vgl. Tabelle 5.3). Aktionen können dabei nach ihrem Zeithorizont gegliedert werden (Wiendahl 1997, S. 13). Daraus ergeben sich kurzfristige, operative und langfristige strategische Maßnahmen (Bauer et al. 2019b, S. 90). Aufgrund der Automatisierung der Entscheidungen auf der Ebene der Produktionssteuerung erfolgt für diese Arbeit eine Einschränkung auf kurzfristige operative Maßnahmen (vgl. Anforderung A1; Anforderung E2; Anforderung S1).

Tabelle 5.3: Schritt 3: Aktionsraum definieren

Ziel	Definition des Aktionsraums, welcher die Handlungsalternativen des Entscheidungsmodells beschreibt.
Eingabe (Daten)	–
Ausgabe (Daten)	Aktionsraum $\mathcal{A}$ (zu Schritt 4)
Eingabe (Wissen)	–
Ausgabe (Wissen)	–
Teilschritte	Aktionen identifizieren → Aktionsraum mathematisch formulieren

Als Grundlage für die Identifikation von Aktionen im ersten Teilschritt kann teilweise auf die Literatur zurückgegriffen werden. Ausgehend vom Modell der Fertigungssteuerung nach Lödging (2016) formuliert Pielmeier (2020, S. 122–127) verschiedene Maßnahmen für die Auftragsfreigabe, die Kapazitätssteuerung und die Reihenfolgebildung. Für die Produktionssteuerung sind dabei insbesondere die Maßnahmen der Kapazitätssteuerung und der Reihenfolgebildung relevant. Ähnliche Maßnahmen schlägt Niehues (2017, S. 102–106) vor, kategorisiert diese jedoch anders und bezieht sie explizit auf die Störungsbehandlung in der Werkstattfertigung. Beiden ist jedoch gemein, dass die Maßnahmen allgemein formuliert sind und nicht die Charakteristik der komplexen Werkstattfertigung berücksichtigen (vgl. Abschnitt 2.1.1).

Die Charakteristik der komplexen Werkstattfertigung bedingt, dass die Maßnahmen überwiegend nur eingeschränkt übertragbar oder bereits durch die komplexe Werkstattfertigung abgebildet sind (vgl. Abschnitt A.5). Entsprechend ergibt sich aus der Berücksichtigung dieser Charakteristik der komplexen Werkstattfertigung (vgl. Anforderung A2), dass insbesondere Aktionen zur Änderung der Priorisierung von Losen relevant sind. Dies bedeutet, dass über Aktionen jedes Los, welches sich bereits im Produktionssystem befindet, in seiner Priorität erhöht oder verringert werden kann. Aufgrund des Aufrufs des Agenten zu diskreten Zeitpunkten ist es darüber hinaus notwendig, eine Aktion vorzusehen, welche keine Handlung vornimmt (vgl. Abbildung 5.8).

Für die mathematische Formulierung im zweiten Teilschritt folgt die Dimension des Aktionsraums $\mathcal{A}$ mit $2 \cdot n_L + 1$, wobei die zusätzliche Aktion die Alternative „keine Handlung“ repräsentiert. Entsprechend der Formulierung des Zustandsraums $\mathcal{S}$ ist es auch für den Aktionsraum $\mathcal{A}$ notwendig, dass die Dimension konstant gehalten wird. Nach Glei-

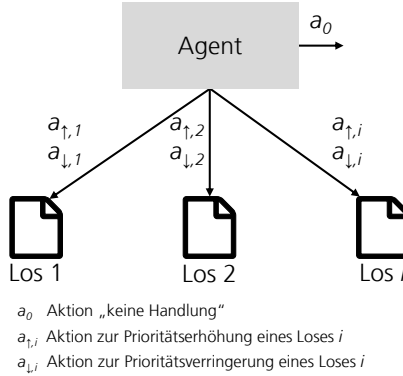


Abbildung 5.8: Zusammensetzung des Aktionsraums des Agenten

Abbildung 5.6 kann der Aktionsraum A dann definiert werden aus Aktionen a , konkret der Aktion „keine Handlung“ a_0 sowie je Los einer Aktion zur Prioritätserhöhung $a_{\uparrow}$ und zur Prioritätsverringern $a_{\downarrow}$.

$$A = \begin{pmatrix} a_0 \\ a_{\uparrow,1} \\ a_{\downarrow,1} \\ \dots \\ a_{\uparrow,i} \\ a_{\downarrow,i} \end{pmatrix} \quad \text{mit } i = n_L \quad (5.6)$$

Wird durch den Agenten eine Aktion zur Prioritätserhöhung $a_{\uparrow,i}$ oder zur Prioritätsverringern $a_{\downarrow,i}$ ausgewählt, wird der Steuer-Flussfaktor FF_{Steuer} des entsprechenden Loses angepasst mit $FF_{Steuer} = FF_{Steuer} + \Delta FF_{Steuer}$. Die Höhe der Anpassung ΔFF_{Steuer} wird dabei im Training festgelegt (vgl. Abschnitt 5.3.2.2). Im Produktionssystem umgesetzt im Sinne der Auswahl des nächsten zu bearbeitenden Auftrags an einer Maschine werden die Aktionen durch das existierende Dispatching, für welches der Agent in einer übergeordneten Regelung die Leitlinien vorgibt. Deshalb wird im Zustandsraum auch nur die Maschinengruppe MG erfasst, welche aus n_M Maschinen besteht, und nicht die einzelne Maschine.

Aktionen zur Anpassung der Stückzahl werden für die vorliegende Arbeit nicht weiter betrachtet. Zum einen deshalb, da Stückzahlreduzierungen aufgrund von Effizienzverlusten in der komplexen Werkstattfertigung in der Regel nicht umgesetzt werden, sondern die Lose die Fertigung mit unveränderter Stückzahl durchlaufen (vgl. Abschnitt A.5). Zum anderen da Stückzahlerhöhungen zusätzliche Losstarts bedingen, welche außerhalb des Betrachtungsrahmens liegen (vgl. Anforderung A1). Die Berücksichtigung von Stückzahl-anpassungen stellt jedoch eine interessante zukünftige Erweiterungsmöglichkeit dar (vgl. Abschnitt 8.2).

Schritt 4: Simulationsmodell vorbereiten

Die Vorbereitung eines Simulationsmodells des Produktionssystems ist die Aufgabe des vierten Schritts (vgl. Tabelle 5.4). Das Simulationsmodell ist notwendig, da für das Training des Agenten sehr viele Iterationen notwendig sind, welche beim Einsatz in einem realen Produktionssystem insbesondere zu Beginn des Trainings nicht ohne negative Auswirkungen durchgeführt werden könnten (vgl. Abschnitt 5.3.1). Dabei ist zu beachten, dass Modelle stets durch die Merkmale Abbildung, Verkürzung und Pragmatismus gekennzeichnet sind (Stachowiak 1973, S. 131–133). Insbesondere das Verkürzungsmerkmal ist hier von Bedeutung, da nicht alle Attribute des realen Produktionssystems im Simulationsmodell erfasst werden können. Das Simulationsmodell des Produktionssystems stellt deshalb immer eine vereinfachte Abbildung des realen Produktionssystems dar. Zur Vorbereitung eines Simulationsmodells werden die Teilschritte der Erstellung, der Modellierung von Ereignis-, Zustands- und Aktionsraum sowie der Datenkonnektivitätsdefinition durchlaufen.

Im ersten Teilschritt ist zu prüfen, ob bereits ein geeignetes Simulationsmodell des betrachteten Produktionssystems vorhanden ist. Für die Simulation von Produktionssystemen ist dabei seit vielen Jahren die Ereignisorientierte Simulation, fortlaufend mit dem englischen Akronym Discrete Event Simulation (DES) bezeichnet, weit verbreitet (Negahban & Smith 2014, S. 242). Bei der DES wird das Produktionssystem als eine Reihe von Ereignissen in Form von Zustandswechseln modelliert (Robinson 2004, S. 15). Deshalb eignet sich dieser Ansatz gut zur Kombination mit diskretem RL. Ist solch ein DES-Modell des Produktionssystems vorhanden, sind die nachfolgenden Teilschritte jeweils auf ihr Vorhandensein zu

Tabelle 5.4: Schritt 4: Simulationsmodell vorbereiten

Ziel	Vorbereitung eines Simulationsmodells des Produktionssystems, mit welchem das Entscheidungsmodell ohne Auswirkungen auf das reale Produktionssystem trainiert werden kann.
Eingabe (Daten)	Ereignisraum E (von Schritt 1) Zustandsraum S (von Schritt 2) Aktionsraum A (von Schritt 3)
Ausgabe (Daten)	Simulationsmodell mit E , S und A (zu Schritt 5)
Eingabe (Wissen)	–
Ausgabe (Wissen)	–
Teilschritte	Simulationsmodell erstellen → Ereignis-, Zustands- und Aktionsraum modellieren → Datenkonnektivität definieren

prüfen und das Simulationsmodell gegebenenfalls zu erweitern. Ist bislang kein (geeignetes) Simulationsmodell des Produktionssystems vorhanden, muss dieses zuvor erstellt werden, wenngleich der Modellierungsaufwand mitunter hoch sein kann und spezifisches Wissen des realen Produktionssystems sowie der Modellierung benötigt. Ein allgemeines Vorgehen zur Modellierung eines Produktionssystem mittels DES (Fishman 2001, S. 37–38) und konkret für die komplexe Werkstattfertigung (Fowler et al. 2015, S. 668–677) kann der Literatur entnommen werden. Zu beachten ist dabei insbesondere das Verkürzungsmerkmal und die damit einhergehende Festlegung, welche Attribute des realen Produktionssystems vernachlässigt werden können. Eine physikalische Modellierung einzelner Prozesse ist hier nicht notwendig, die funktionale Beschreibung des Produktionsablaufs ist ausreichend.

Neben dem Produktionsablauf muss das Simulationsmodell in der Lage sein, den Ereignisraum E , den Zustandsraum S und den Aktionsraum A abzubilden. Deren Modellierung ist Inhalt des zweiten Teilschritts. Die Modellierung des Ereignisraums umfasst die Erstellung entsprechender Objekte im Simulationsmodell, um Ereignisse aus dem Wertschöpfungsnetzwerk abbilden und kommunizieren zu können sowie deren Zuordnung zur Definition des Ereignisraums. Für den Zustandsraum müssen die definierten Kenngrößen als Variablen in das Simulationsmodell integriert und der Definition des Zustandsraums zugeordnet werden. Der Aktionsraum bedarf einer Zuordnung zwischen den definierten Aktionen und den dazugehörigen operativen Änderungen im Simulationsmodell. Durch die Modellierung von Ereignis-, Zustands- und Aktionsraum wird das Simulationsmodell zur Umgebung, mit der der Agent während des Trainings interagiert. Für einen Zeitpunkt t kann dann der

Zustand S_t ausgelesen bzw. eine Aktion A_t umgesetzt werden. Hierfür ist eine bidirektionale Schnittstelle zu schaffen, über welche der Agent mit seiner Umgebung in Form des Simulationsmodells interagieren kann.

Im Rahmen des dritten Teilschritts ist die Datenkonnektivität zu definieren. Die Mindestvoraussetzung dabei ist, dass das Simulationsmodell mit realitätsnahen Daten ausgeführt werden kann und somit eine Überführung in den Produktivbetrieb ab Schritt 12 möglich ist. Idealerweise ist das Simulationsmodell darüber hinaus als Digitaler Zwilling ausgeführt (Kuhn 2017, S. 440–441), um sich bei Verwendung realer Daten möglichst identisch zum realen Produktionssystem zu verhalten und das Simulationsmodell bei Änderungen des realen Produktionssystems anzupassen. Die Verwendung von Digitalen Zwillingen in der produzierenden Industrie ist derzeit jedoch noch gering (Lödding 2020, S. 20). Ist das Produktionsmodell nicht als Digitaler Zwilling ausgeführt, ist über geeignete Mechanismen sicherzustellen, dass sich Simulationsmodell und reales Produktionssystem unter Berücksichtigung der getroffenen Verkürzungen möglichst ähnlich verhalten. Ein mögliches Verfahren hierfür ist, das Simulationsmodell mit den Daten des realen Produktionssystems über einen bestimmten Zeitraum parallel auszuführen, Abweichungen zwischen Produktionssystem und Simulationsmodell zu identifizieren und Korrekturmaßnahmen am Simulationsmodell vornehmen, um diese Abweichungen zu minimieren. Idealerweise findet diese Anpassung im Produktivbetrieb kontinuierlich oder mindestens in fest definierten Abständen statt, um das Simulationsmodell dauerhaft aktuell zu halten.

Schritt 5: Daten integrieren

Der fünfte Schritt umfasst die Integration von Daten existierender IT-Systeme, um das Simulationsmodell mit realen Daten zu betreiben (vgl. Tabelle 5.5). Dies ist notwendig, um für das Training realistische Bedingungen zu schaffen und eine Überführung in den Produktivbetrieb ab Schritt 12 zu ermöglichen. Hierfür müssen Schnittstellen für den Ereignis-, Zustands- und Aktionsraum geschaffen werden.

Um den Aufwand zu diesem Zeitpunkt der Design-Phase zu minimieren, können Schnittstellen zur Integration von Daten auch dahingehend ausgeführt werden, dass die relevanten Daten aus den entsprechenden IT-Systemen exportiert und über eine Schnittstelle in das

Tabelle 5.5: Schritt 5: Daten integrieren

Ziel	Integration von Daten existierender IT-Systeme, um das Simulationsmodell mit realen Daten zu betreiben.
Eingabe (Daten)	Simulationsmodell mit E , S und A (von Schritt 4)
Ausgabe (Daten)	Simulationsmodell mit E , S und A (zu Schritt 6)
Eingabe (Wissen)	Strategie zur Erkennung relevanter Ereignisse im Betrieb (von Schritt 1)
Ausgabe (Wissen)	Schnittstellen zu relevanten Systemen der Produktions-IT-Landschaft (zu Schritt 12)
Teilschritte	Schnittstellen für Ereignisraum implementieren → Schnittstellen für Zustandsraum implementieren → Schnittstellen für Aktionsraum implementieren

Simulationsmodell integriert werden. Da die Schnittstellen jedoch spätestens bei der Überführung in den Produktivbetrieb zwingend benötigt werden, ist dennoch zu empfehlen, diese direkt zum jetzigen Zeitpunkt sowohl im Simulationsmodell als auch im korrespondierenden System der Produktions-IT-Landschaft, zu implementieren. Zum jetzigen Zeitpunkt sind jedoch unidirektionale Schnittstellen vom Simulationsmodell zum Agenten ausreichend, da Daten nur ausgelesen werden müssen.

Bei den Daten für den Ereignisraum E handelt es sich um Daten aus dem Wertschöpfungsnetzwerk, ergänzt um Daten aus dem Produktionssystem. Während erstere typischerweise aus einem SCM übernommen werden können, sind zweitere üblicherweise im MES vorhanden (vgl. Abschnitt 2.1.4; Abschnitt 2.2.4). Ereignisse aus dem Wertschöpfungsnetzwerk werden meist insbesondere dann schon identifiziert und verarbeitet, wenn auch ein SCEM vorhanden ist. Auf den Kernfunktionen des Überwachens und Meldens eines SCEM kann der Agent aufbauen und Maßnahmen zur Reaktion in der Produktionssteuerung ableiten und umsetzen. Sind die in Schritt 1 definierten Strategien zur Erkennung relevanter Ereignisse im Betrieb nicht vorhanden, muss das SCEM gegebenenfalls um diese Funktionalität erweitert werden. Die Schnittstelle kann dann unidirektional vom SCEM oder allgemeiner formuliert vom SCM zum Simulationsmodell gestaltet werden, um Daten zum Produkttyp P sowie zur Änderung von Plan-Flussfaktor ΔFF_{Plan} und Stückzahl Δn_{Stk} zu übertragen. Ergänzt werden die Daten in der Vorverarbeitung um die Losnummer L , den Plan-Flussfaktor FF_{Plan} und die Stückzahl n_{Stk} . Hierfür ist eine unidirektionale Schnittstelle vom MES zum Simulationsmodell notwendig. In SCM, MES und Simulationsmodell

müssen die Variablen des Ereignisraums dann mit den im jeweiligen IT-System vorhandenen Variablen verknüpft werden.

Wenn Ereignisse aus dem Wertschöpfungsnetzwerk in der Realität nur in geringer Häufigkeit auftreten, kann – um das Training des Entscheidungsmodells zu beschleunigen – ein Ereignis-Generator implementiert werden. Für eine einfache Übertragbarkeit ist dabei darauf zu achten, dass dieser die gleiche Schnittstelle aufweist, welche auch das SCEM zur Verfügung stellt. Für die Umsetzung des Ereignis-Generators eignet sich eine Poisson-Verteilung zur Generierung zufälliger, diskreter Ereignisse, welche unabhängig voneinander sind (Papula 2016, S. 367–369). Hierfür ist der Erwartungswert μ_P und das zugehörige Zeitintervall t_e zu definieren. Der Ereignis-Generator wählt dann anhand der Poisson-Verteilung je Zeitintervall so viele Ereignisse wie der Erwartungswert vorgibt. Anhand der Ereignisse können Lose um ΔFF_{Plan} beeinflusst werden.

Für den Zustandsraum S handelt es sich um Daten, welche typischerweise aus dem MES übernommen werden können (vgl. Abschnitt 2.2.4). Während der Produkttyp P , die Anzahl an Produkttypen n_P und die Anzahl an Maschinengruppen n_{MG} Stammdaten darstellen, zählen die übrigen Daten des Zustandsraums zu den Bewegungsdaten. Diese können weiter unterteilt werden in die Soll-Daten des Freigabe-Flussfaktors $FF_{Freigabe}$, des Plan-Flussfaktors FF_{Plan} und des Steuer-Flussfaktors FF_{Steuer} sowie die Ist-Daten der Maschinengruppe MG , des Fertigstellungsgrads F und des Ist-Flussfaktors FF_{Ist} . Zur Übertragung der Daten ist eine unidirektionale Schnittstelle vom MES zum Simulationsmodell notwendig. In MES und Simulationsmodell müssen die Variablen des Zustandsraums dann mit den im jeweiligen IT-System vorhandenen Variablen verknüpft werden.

Der Aktionsraum A benötigt für das Training keine Datenintegration mit der Produktions-IT Landschaft, da während des Trainings nur eine Interaktion zwischen Agent und Simulationsmodell stattfindet. Wird der Agent jedoch in den Produktivbetrieb überführt, interagiert dieser nicht mehr mit dem Simulationsmodell, sondern mit dem realen Produktionssystem. Um Aktionen umsetzen zu können, müssen alle Aktionstypen a mit der Veränderung einer entsprechenden Variable in der Produktionssteuerung, typischerweise im MES, verknüpft werden. Für Prioritätsanpassungen ist dies der Steuer-Flussfaktor FF_{Steuer} . Über die Schnittstelle zum MES findet auch die Rückkopplung der Aktionen in der Produktionsplanung statt. Entsprechend muss die Schnittstelle, welche das MES bereitstellt,

bidirektional ausgeführt sein. Vom MES zum Simulationsmodell für das Übertragen von Daten und vom Agenten zum MES für die Umsetzung von Aktionen im Produktivbetrieb.

Bei der Implementierung aller Schnittstellen sind offene Protokolle gegenüber proprietären Lösungen vorzuziehen, um die Austauschbarkeit und Erweiterbarkeit zu vereinfachen (Bauer et al. 2017, S. 201–202). Sogenannte Middlewares gehen die Herausforderung der Integration heterogener Schnittstellen und Datenformate an, wobei das Routing zwischen heterogenen Datenformaten die größte Herausforderung darstellt (Schel et al. 2018, S. 180–181). Dies gilt insbesondere auch für Digitale Zwillinge, wenn nicht nur Daten ausgetauscht werden müssen, sondern auch diverse Teilsimulationsmodelle zu einem Gesamtsimulationsmodell integriert werden müssen (Kuhn 2017, S. 442–443).

Schritt 6: Belohnungsfunktion definieren

Im sechsten Schritt wird die Belohnungsfunktion definiert (vgl. Tabelle 5.6). Dies ist einer der herausforderndsten und zugleich der für das Ergebnis kritischste Schritt bei der Modellierung einer Anwendung als RL-Problem (Sutton & Barto 2018, S. 53). Die Belohnungsfunktion hat einen maßgeblichen Einfluss auf die Stabilität des ARIW. Der Agent strebt im Training die Optimierung der langfristigen, kumulierten Belohnung an, weshalb es von besonderer Bedeutung ist, dass die Belohnung auch das tatsächliche Ziel widerspiegelt (vgl. Abschnitt 2.3.2). Für die Definition der Belohnungsfunktion ist es notwendig, die Belohnungen für in Bearbeitung befindliche Lose sowie abgeschlossene Lose zu definieren und anschließend die Belohnungsfunktion mathematisch zu formulieren.

Die Belohnungsfunktion im Markov-Entscheidungsprozess ist definiert als $R(S_t, A_t, S_{t+1})$ (vgl. Abschnitt 2.3.2). Das bedeutet, dass alle Kenngrößen, die zur Bestimmung einer Belohnung beim Übergang zwischen zwei Zuständen S_t und S_{t+1} notwendig sind, im Zustandsraum S enthalten sein müssen. Dabei können Belohnungsfunktionen jedoch sehr unterschiedlich gestaltet sein und unterscheiden sich insbesondere in der Dichte, in welcher sie Belohnungen zurückgeben. Belohnungsfunktionen mit sogenannter spärlicher Belohnung³ geben bei den meisten Zustandsübergängen null zurück und nur für sehr wenige Zustandsübergänge, oftmals am Ende einer Episode, eine Belohnung. Der Agent erfährt

³engl. sparse reward

Tabelle 5.6: Schritt 6: Belohnungsfunktion definieren

Ziel	Definition einer Belohnungsfunktion, um die Zielsetzung des Entscheidungsmodells möglichst effektiv und effizient zu erreichen.
Eingabe (Daten)	Simulationsmodell mit E , S und A (von Schritt 5) <i>Opt: Modellparameter M_{t+1} (von Schritt 11)</i>
Ausgabe (Daten)	Simulationsmodell mit E , S , A und R (zu Schritt 7) <i>Opt: Modellparameter M_{t+1} (zu Schritt 7)</i>
Eingabe (Wissen)	<i>Opt: Detaillierte Analyse der Modellevaluation (von Schritt 11)</i>
Ausgabe (Wissen)	Fehlende Kenngrößen zur Bildung der Belohnungsfunktion (zu Schritt 2)
Teilschritte	Belohnung für Los in Bearbeitung definieren → Belohnung für abgeschlossenes Los definieren → Belohnungsfunktion mathematisch formulieren

deshalb während des Trainings kein unmittelbares Feedback zur ausgewählten Aktion A_t . Zwar werden mit spärlichen Belohnungen Fehlanreize vermieden, die Trainingsdauer kann dadurch jedoch insbesondere bei großen Zustands- und Aktionsräumen massiv ansteigen. Als sogenannte dichte Belohnung⁴ werden Belohnungsfunktionen bezeichnet, welche bei sehr vielen oder allen Zustandsübergängen eine Belohnung zurückgeben. Während des Trainings erhält der Agent deshalb sehr engmaschig bei ausgewählten Aktionen A_t ein unmittelbares Feedback. Die Trainingsdauer sinkt dadurch signifikant, es können jedoch Fehlanreize zum Erreichen lokaler Optima entstehen. Damit wird auch das seit vielen Jahren bekannte sogenannte Credit Assignment Problem deutlich, welches besagt, dass jeder Zielzustand durch eine mitunter sehr große Anzahl von Aktionen erreicht wird. Während die Belohnung für den Zielzustand meist einfach zu definieren ist, sind die Belohnungen für Zwischenzustände herausfordernd (Minsky 1961, S. 20). Aufgrund der typischerweise sehr großen Zustands- und Aktionsräume zur Abbildung von Produktionssystemen muss für die vorliegende Anwendung zur Verbesserung der Liefertermineinhaltung trotz Ereignissen aus dem Wertschöpfungsnetzwerk dennoch eine dichte Belohnung formuliert und die Erreichung der Zielsetzung über eine geeignete Evaluation in Schritt 11 sichergestellt werden. Die Belohnungsfunktion wird dabei über alle Trainingsepisoden konstant gehalten. Sie kann jedoch angepasst werden und es können die nachfolgenden Teilschritte erneut durchlaufen werden, wenn die Zielsetzung anhand der Evaluation in Schritt 11 nicht erreicht wurde.

⁴engl. dense reward

Grundlage für die Anpassung stellt dann die detaillierte Analyse der Modellevaluation aus Schritt 11 dar.

Im ersten Teilschritt wird die Belohnung für Lose in Bearbeitung, die sich also im Bestand eines Produktionssystems befinden, definiert. Die Grundlage stellt die Differenz zwischen Plan-Flussfaktor und Ist-Flussfaktor $\Delta FF_{Plan-Ist}$ dar. Hat sich diese zwischen t und $t + \Delta t$ verbessert, wird der Agent dafür belohnt – im umgekehrten Fall für eine Verschlechterung bestraft. Die Belohnung wird dadurch dicht, der Agent erhält nach jedem Schritt ein unmittelbares Feedback.

Die Belohnung für abgeschlossene Lose wird im zweiten Teilschritt definiert. Diese ergibt sich aus der Anzahl der Lose, welche zwischen t und $t + \Delta t$ fertiggestellt wurden. Überpünktliche und pünktliche Lose mit $\Delta FF_{Plan-Ist} \geq 0$ werden dabei belohnt, verspätete Lose mit $\Delta FF_{Plan-Ist} < 0$ bestraft. Der Einfluss auf die gesamte Belohnung ist dabei größer als für Lose in Bearbeitung, denn zwischenzeitliche Verspätungen sind durchaus zu tolerieren, wenn bei Fertigstellung des Loses der Liefertermin gegenüber dem Wertschöpfungsnetzwerk eingehalten wird.

In der mathematischen Formulierung, dem dritten Teilschritt, ergibt sich die Belohnungsfunktion $R(S_t, A_t, S_{t+1})$ nach Gleichung 5.7 aus der Summe der Belohnungsfunktionen für sich in Bearbeitung befindliche Lose im Bestand und abgeschlossene Lose.

$$R(S_t, A_t, S_{t+1}) = \sum_{i=1}^{n_{L,B}} R_B(S_t, A_t, S_{t+1}) + \sum_{i=1}^{n_{L,A}} R_A(S_t, A_t, S_{t+1}) \quad (5.7)$$

mit der Belohnungsfunktion für Lose im Bestand

$$R_B(S_t, A_t, S_{t+1}) = \Delta FF_{Plan-Ist,t+1} - \Delta FF_{Plan-Ist,t} \quad (5.8)$$

und mit der Belohnungsfunktion für abgeschlossene Lose

$$R_A(S_t, A_t, S_{t+1}) = \begin{cases} 1 & \Delta FF_{Plan-Ist} \geq 0 \\ -1 & \Delta FF_{Plan-Ist} < 0 \end{cases} \quad (5.9)$$

Wurde die Belohnungsfunktion definiert und mathematisch formuliert, kann anschließend in Schritt 7 das Training initialisiert werden. Kann die Belohnungsfunktion mit den vorhandenen Kenngrößen im Zustandsraum S nicht gebildet werden, müssen die fehlen-

den Kenngrößen ermittelt werden. Anschließend kann die Methode ab Schritt 2 nochmals durchlaufen werden.

5.3.2.2 Phase II: Training

An das Design schließt nahtlos das Training an, welches zuerst initialisiert wird. Anschließend folgt eine iterative Schleife mit der Ausführung eines Episodenschritts, der Berechnung der Belohnung und der Anpassung der Modellparameter. Zuletzt folgt die Evaluation des Modells (vgl. Abbildung 5.9). Die fünf Schritte werden im Folgenden näher beschrieben. Ergebnis des Trainings ist ein positiv evaluiertes Entscheidungsmodell des Agenten, welches anschließend in den Produktivbetrieb übertragen werden kann.

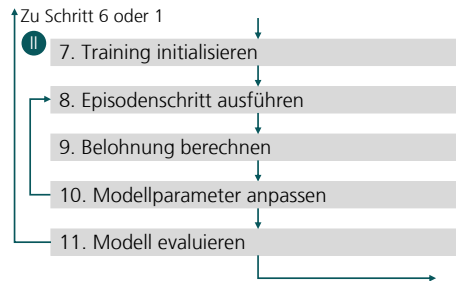


Abbildung 5.9: Phase II: Training der Methode zur Gestaltung des adaptiven Regelkreises für ein integriertes Wertschöpfungsnetzwerk

Schritt 7: Training initialisieren

Zu Beginn des Trainings im siebten Schritt müssen das Simulationsmodell und der Agent mit initialen Parametern für das Training initialisiert werden (vgl. Tabelle 5.7). Für die Initialisierung des Trainings ist es notwendig, den Agenten zu implementieren. Anschließend sind der initiale Zustand des Simulationsmodells und der Modellparameter zu definieren sowie Simulationsmodell und Agent damit zu initialisieren.

Die Implementierung des Agenten ist Inhalt des ersten Teilschritts. Dabei ist eine Verknüpfung mit den Schnittstellen des Simulationsmodells herzustellen, um auf den Zustandsraum S und den Aktionsraum A lesend bzw. schreibend zugreifen und den Wert der

Tabelle 5.7: Schritt 7: Training initialisieren

Ziel	Simulationsmodell und Agent mit initialen Parametern für das Training initialisieren.
Eingabe (Daten)	Simulationsmodell mit E , S , A und R (von Schritt 6) <i>Opt: Modellparameter M_{t+1} (von Schritt 6)</i>
Ausgabe (Daten)	Modellparameter M_0 und Zustand S_0 (zu Schritt 8)
Eingabe (Wissen)	–
Ausgabe (Wissen)	–
Teilschritte	Agent implementieren → initialen Zustand definieren → initiale Modellparameter definieren → Simulationsmodell und Agent initialisieren

Belohnungsfunktion R ermitteln zu können. Für die Interaktion zwischen Simulationsmodell und Agent sind Abstraktionsschichten üblich, welche das Simulationsmodell dann als Umgebung beschreiben. Weit verbreitet ist hierfür die Abstraktionsschicht der OpenAI Gyms (Brockman et al. 2016, S. 1–2), welche auch von diversen Open-Source-Frameworks für RL angesprochen werden kann (vgl. Abschnitt 2.3.2). Der Agent interagiert dann, wie in Abbildung 5.10 dargestellt, über die Abstraktionsschicht der Umgebung mit dem Simulationsmodell. So können auch gegebenenfalls nicht zu ersetzende proprietäre Schnittstellen eines Simulationsmodells abstrahiert und eine einheitliche Schnittstelle für verschiedene Algorithmen bereitgestellt werden. Der Agent muss dann so implementiert werden, dass die Eingaben und Ausgaben der Umgebung entsprechend verarbeitet werden können. Durch die Implementierung eines RL-Algorithmus wird der Agent adaptiv und lernfähig (vgl. Anforderung E3; Anforderung E4). Idealerweise ist der Agent dahingehend modular zu gestalten, neben verschiedenen Modellparametern auch verschiedene Algorithmen anwenden zu können. Der Algorithmus wird analog zu den unten beschriebenen statischen Modellparametern über das Training nicht variiert.

Nach derzeitigem Stand der Forschung kann kein universell vorteilhafter Algorithmus ausgewählt werden, dieser ist vielmehr von der Charakteristik der Anwendung abhängig. Des Weiteren entwickelt sich die Forschung im Themenfeld der RL-Algorithmen sehr schnell weiter, was zukünftig noch leistungsfähigere Algorithmen erwarten lässt. Unabhängig davon sind durch in Betracht kommende Algorithmen jedoch Mindestanforderungen zu erfüllen, damit sich diese für die vorliegende Problemstellung eignen. Hinsichtlich des Zustandsraums S müssen die Algorithmen in der Lage sein, kontinuierliche und diskrete

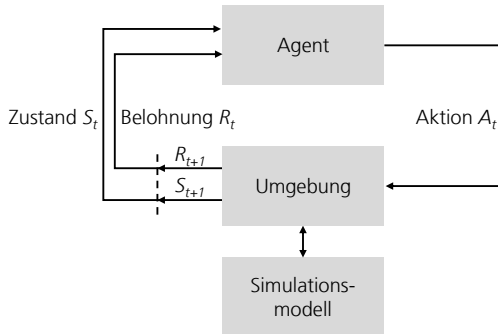


Abbildung 5.10: Interaktion eines Agenten mit einem Simulationsmodell über die Abstraktionsschicht der Umgebung

Werte zu verarbeiten sowie insbesondere bei einer hohen Anzahl von Losen n_L mit hochdimensionalen Eingangsdaten umgehen zu können (vgl. Schritt 2). Für den Aktionsraum A müssen diskrete Aktionen verarbeitet werden können (vgl. Schritt 3). Zur Nachvollziehbarkeit und Parameteroptimierung der Algorithmen ist eine saubere und vollständige Dokumentation notwendig. Darüber hinaus erleichtern und beschleunigen die Verfügbarkeit von Open-Source-Code oder die Integration in Open-Source-Frameworks die Erprobung der Algorithmen (vgl. Abschnitt 2.3.2). Vergleichen lassen sich Algorithmen, welche diese Mindestanforderungen erfüllen, hinsichtlich ihrer Eignung für die vorliegende Fragestellung der Verbesserung der Liefertermineinhaltung bei Ereignissen aus dem Wertschöpfungsnetzwerk. Anhand des Verlaufs der Belohnung über die Zeit kann die Konvergenz und Robustheit der Algorithmen sowie die Stichprobeneffizienz und die damit einhergehende Trainingsgeschwindigkeit bewertet werden (vgl. Abschnitt 2.3.2).

Der Zeitpunkt, welcher durch die initialen Parameter beschrieben wird, wird mit t_0 bezeichnet. Entsprechend ist im zweiten Teilschritt ein initialer Zustand S_0 zu definieren. Die Kenngrößen entsprechend dabei dem Zustandsraum S , jedoch muss deren Ausprägung zum Zeitpunkt t_0 noch festgelegt werden. Dabei ist darauf zu achten, den initialen Zustand so zu definieren, so dass mit den nachfolgenden Schritten im Training möglichst effizient gelernt werden kann. Dies bedeutet insbesondere, dass das Simulationsmodell nicht mit einem leeren Bestand startet – andernfalls ist zuerst eine Einschwingphase des Simulationsmodells zu erwarten, welche so in einem laufenden realen Produktionssystem nicht wiederzufinden

ist. Geprüft werden, ob die Einschwingphase des Simulationsmodells bereits abgeleistet wurde, kann anhand des Vergleichs von Plan-Flussfaktor FF_{Plan} und Ist-Flussfaktor FF_{Ist} . Während diese anfangs noch mit sehr unterschiedlichen Steigungen verlaufen, wird der Verlauf im eingeschwungenen Zustand annähernd parallel.

Im nächsten Teilschritt sind die initialen Modellparameter M_0 zu definieren, welche sich aus Hyperparametern, Agentenparametern und Trainingsparametern zusammensetzen. Einzelne Parameter können dabei statisch oder dynamisch sein. Statische Modellparameter werden über alle Episoden des Trainings konstant gehalten und nur gegebenenfalls nach der Evaluation und Abschluss einer Epoche in Schritt 11 verändert, wohingegen dynamische Modellparameter während des Trainings in Schritt 8 bis Schritt 10 jeweils verändert werden können. Wurden Modellparameter M_{t+1} von Schritt 6 übergeben, werden diese als Ausgangsbasis verwendet, um darauf aufbauend angepasste Modellparameter zu definieren, mit welchen dann, ab hier wiederum als Modellparameter M_0 bezeichnet, das nächste Training gestartet wird. Eine Übersicht der Modellparameter M_t ist Tabelle 5.8, deren detaillierte Erläuterung Abschnitt A.6, zu entnehmen. Aus den Abbruchbedingungen von Episode und Epoche ergibt sich, wie häufig Schritt 8 bis Schritt 10 dieser Methode durchlaufen werden. Bei einer Aktoranzahl n_{Akt} größer eins werden die Schritte separat durch jeden Akteur durchlaufen.

Im letzten Teilschritt werden das Simulationsmodell und der Agent initialisiert. Für das Simulationsmodell bedeutet das, dass die zugehörigen Daten zum Zustand S_0 mittels der in Schritt 5 definierten Schnittstellen in das Simulationsmodell geladen werden. Die Umgebung und der Agent werden mit den Modellparametern M_0 parametrisiert. Anschließend kann die erste Episode für das Training gestartet werden. In der Regel werden Schritt 8 bis Schritt 10 anschließend automatisch durch den Agenten, die Umgebung und das Simulationsmodell iterativ durchlaufen.

Tabelle 5.8: Übersicht der Modellparameter

Parameter	Typ
Hyperparameter	
Struktur des KNN D_{KNN}	statisch
Weitere Hyperparameter	Die weiteren Hyperparameter sind vom eingesetzten Algorithmus abhängig, weshalb sie erst nach der Auswahl eines Algorithmus bestimmt werden können.
Agentenparameter	
Modellgewichte θ	dynamisch
Flussfaktor-Anpassung ΔF_{Steuer}	statisch
Schrittweite Δt	statisch
Trainingsparameter	
Abbruchbedingung Episode	statisch
Abbruchbedingung Epoche	statisch
Aktoranzahl n_{Akt}	statisch

Schritt 8: Episodenschritt ausführen

Für das Training muss im achten Schritt der Methode der Episodenschritt ausgeführt werden, um den nachfolgenden Zustand zu ermitteln (vgl. Tabelle 5.9). Hierfür wird zuerst der derzeitige Zustand ermittelt, anschließend in Abhängigkeit von diesem eine Aktion ausgeführt und zuletzt der Episodenschritt simuliert, um den nachfolgenden Zustand ermitteln zu können.

Tabelle 5.9: Schritt 8: Episodenschritt ausführen

Ziel	Episodenschritt ausführen und nachfolgenden Zustand ermitteln.
Eingabe (Daten)	Modellparameter M_0 und Zustand S_0 (von Schritt 7) <i>Opt: Modellparameter M_{t+1} und Zustand S_{t+1} (von Schritt 10)</i>
Ausgabe (Daten)	Modellparameter M_t und Zustand S_{t+1} (zu Schritt 9)
Eingabe (Wissen)	–
Ausgabe (Wissen)	–
Teilschritte	Zustand ermitteln → Aktion ausführen → Episodenschritt simulieren → Nachfolgenden Zustand ermitteln

Im ersten Teilschritt muss der Agent den Zustand S_t ermitteln. Die Umgebung des Agenten abstrahiert dabei das Simulationsmodell und stellt den Zustandsvektor zur Verfügung.

Beim ersten Episodenschritt ist dies der Zustand S_0 (von Schritt 7), bei allen weiteren Episodenschritten der Zustand S_{t+1} (von Schritt 10).

Der Algorithmus des Agenten wählt anschließend im zweiten Teilschritt eine Aktion aus. Die ausgewählte Aktion $A_t(S_t)$ mit $A_t \in A$ ist dabei abhängig von der Strategie π . Über die Umgebung wird die ausgewählte Aktion danach im Simulationsmodell ausgeführt.

Für die Simulation des Episodenschritts im dritten Teilschritt wird das Simulationsmodell bis zum nächsten Zeitpunkt $t + \Delta t$ ausgeführt. Abhängig von der ausgeführten Aktion verändert sich dabei der Zustand. Dieser nachfolgende Zustand S_{t+1} wird im vierten Teilschritt durch den Agenten ermittelt. Der Zustand S_t , die ausgeführte Aktion A_t und der nachfolgende Zustand S_{t+1} bilden die Grundlage für die Ermittlung der Belohnung in Schritt 9.

Schritt 9: Belohnung berechnen

Im neunten Schritt wird abhängig vom Zustand und der Aktion die zugehörige Belohnung berechnet (vgl. Tabelle 5.10).

Tabelle 5.10: Schritt 9: Belohnung berechnen

Ziel	Belohnung berechnen, um dem Agenten eine Bewertung der durchgeführten Aktion zu ermöglichen.
Eingabe (Daten)	Modellparameter M_t und Zustand S_{t+1} (von Schritt 8)
Ausgabe (Daten)	Modellparameter M_t , Zustand S_{t+1} und Belohnung R_t (zu Schritt 10)
Eingabe (Wissen)	–
Ausgabe (Wissen)	–
Teilschritte	Belohnung berechnen

Zur Berechnung der Belohnung wird die in Schritt 6 definierte Belohnungsfunktion herangezogen. Für die Variablen werden dabei die aktuellen Werte aus den Zuständen S_t und S_{t+1} eingesetzt, um den numerischen Wert der Belohnung R zu berechnen (vgl. Code 1).

Code 1 Berechnung der Belohnung

```
1: function reward( $S_t, A_t, S_{t+1}$ )
2:    $R_B \leftarrow 0$ 
3:   while  $i \leq n_{L,B}$  do
4:      $R_B \leftarrow R_B + (\Delta FF_{Plan-Ist,t+1,i} - \Delta FF_{Plan-Ist,t,i})$ 
5:   end while
6:    $R_A \leftarrow 0$ 
7:   while  $j \leq n_{L,A}$  do
8:     if  $\Delta FF_{Plan-Ist,j} \geq 0$  then
9:        $R_A \leftarrow R_A + 1$ 
10:    else
11:       $R_A \leftarrow R_A - 1$ 
12:    end if
13:  end while
14:   $R = R_B + R_A$ 
15:  return  $R$ 
16: end function
```

Schritt 10: Modellparameter anpassen

Das Lernen und die darauf basierende Anpassung der dynamischen Modellparameter (vgl. Schritt 7) findet im zehnten Schritt statt (vgl. Tabelle 5.11). Zuerst findet hierfür das eigentliche Lernen statt, bei welchem auch die dynamischen Modellparameter angepasst werden. Anhand einer Prüfung der Abbruchbedingungen für Episoden und die Epoche (vgl. Schritt 7) wird anschließend über das weitere Vorgehen entschieden.

Tabelle 5.11: Schritt 10: Modellparameter anpassen

Ziel	Lernvorgang des Algorithmus ausführen und basierend darauf die dynamischen Modellparameter anpassen.
Eingabe (Daten)	Modellparameter M_t , Zustand S_{t+1} und Belohnung R_t (von Schritt 9)
Ausgabe (Daten)	Modellparameter M_{t+1} und Zustand S_{t+1} (zu Schritt 8) Modellparameter M_{t+1} und Belohnung R_t (zu Schritt 11)
Eingabe (Wissen)	–
Ausgabe (Wissen)	–
Teilschritte	Entscheidungsmodell lernen $\rightarrow$ Abbruchbedingungen prüfen

Im ersten Teilschritt lernt der Algorithmus basierend auf dem Zustand S_t , der ausgeführten Aktion A_t , der Belohnung R_t und dem nachfolgenden Zustand S_{t+1} ein Entscheidungsmodell. Hierfür wird der dynamische Modellparameter der Modellgewichte θ angepasst, um, abhängig vom Algorithmus, die Strategie π , die Wertfunktion $V^\pi(S_t)$, die Qualitätsfunktio-

on $Q^\pi(S_t, A_t)$, die Advantage-Funktion $A^\pi(S_t, A_t)$ oder eine Kombination daraus zu lernen (vgl. Abschnitt 2.3.2). Neben den Modellgewichten werden die weiteren dynamischen Modellparameter abhängig von der Schrittzahl der Epoche angepasst. Der Verlauf ist dabei linear, jeweils von deren anfänglichem Wert bis zu deren finalem Wert (vgl. Code 2).

Code 2 Lineare Anpassung von dynamischen Modellparametern

```

1: function schedule(min, max, steps, step)
2:   decayRate  $\leftarrow -(max - min) / steps$ 
3:   parameter  $\leftarrow min + (max - min) \cdot decayRate \cdot step$ 
4:   return parameter
5: end function

```

Im zweiten Teilschritt werden die Abbruchbedingungen für die Episoden und die Epoche geprüft, um über das weitere Vorgehen zu entscheiden (vgl. Abbildung 5.11). Ist keine der Abbruchbedingungen für Episoden erreicht, wird das Training mit Schritt 8 fortgesetzt, indem die Modellparameter M_{t+1} und der Zustand S_{t+1} übergeben werden. Ist eine der Abbruchbedingungen für Episoden, jedoch nicht für die Epoche, erreicht, wird die Episode beendet und das Training mit einer neuen Episode gestartet. Hierfür wird neben den Modellparametern M_{t+1} der Zustand S_0 an Schritt 8 übergeben. Ist zusätzlich eine der Abbruchbedingungen der Epoche erreicht, wird das Training für diese Epoche beendet und keine neue Episode gestartet. Anschließend werden die Modellparameter M_{t+1} und die Belohnung R_t für die Evaluation an Schritt 11 übergeben.

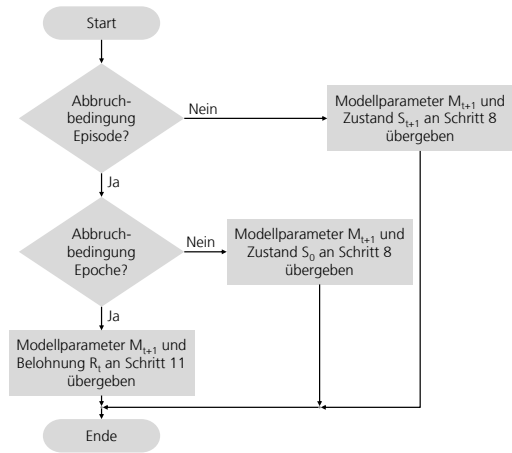


Abbildung 5.11: Prüfung der Abbruchbedingungen von Episoden und der Epoche (Notation nach DIN 66001)

Schritt 11: Modell evaluieren

Im elften Schritt findet die Evaluation des gelernten Entscheidungsmodells statt (vgl. Tabelle 5.12). Hierfür wird das Modell statistisch und kontextuell evaluiert. Anschließend wird die Evaluation geprüft, um über das weitere Vorgehen zu entscheiden.

Tabelle 5.12: Schritt 11: Modell evaluieren

Ziel	Prüfung des Modells, ob mit den ermittelten Modellparametern der Modellzweck erreicht wird.
Eingabe (Daten)	Modellparameter M_{t+1} und Belohnung R_t (von Schritt 10)
Ausgabe (Daten)	Modellparameter M_{t+1} (zu Schritt 12) <i>Opt: Modellparameter M_{t+1} (zu Schritt 6)</i>
Eingabe (Wissen)	–
Ausgabe (Wissen)	<i>Opt: Detaillierte Analyse der Modellevaluation (zu Schritt 1 oder 6)</i>
Teilschritte	Statistisch evaluieren → Kontextuell evaluieren → Evaluation prüfen

Der erste Teilschritt umfasst die statistische Evaluation des Entscheidungsmodells. Während beim überwachten Lernen de-facto Standards wie der F1-Score oder die Wurzel der mittleren Fehlerquadratsumme existieren, ist die Evaluation beim RL stärker von der Problemstellung abhängig – deshalb auch die kontextuelle Evaluation (Goodfellow et al. 2016, S. 108–109, 423–424; Poole & Mackworth 2017, S. 559–560). Beim RL wird den-

noch üblicherweise die Belohnung über die Zeit, konkret über die Schritte einer Epoche, aufgetragen, um den Trainingserfolg zu bewerten (Poole & Mackworth 2017, S. 559–560). Exemplarisch ist dies in Abbildung 5.12 dargestellt. Damit kann geprüft werden, ob die Belohnung im Verlauf der Epoche konvergiert und wie hoch die Volatilität der erreichten Belohnung ist, was auch die Stabilität des ARIW widerspiegelt. Konvergiert die Belohnung, entspricht der Wert, zu welchem die Belohnung konvergiert, gleichzeitig auch der maximal erreichten Belohnung. Daneben können die Steigung der Belohnung und der Zeitpunkt, ab welchem die Belohnung konvergiert, bestimmt werden. Beide Kenngrößen lassen aber insbesondere eine Einschätzung des Trainingsverlaufs zu und weniger der Eignung für die Problemstellung.

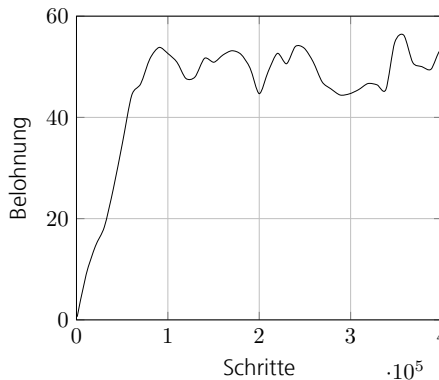


Abbildung 5.12: Exemplarischer Verlauf einer Belohnung über die Schritte einer Epoche

Im zweiten Teilschritt wird deshalb eine kontextuelle Evaluation des Entscheidungsmodells vorgenommen. Dabei wird die Eignung des Entscheidungsmodells für die vorliegende Fragestellung der Verbesserung der Liefertermineinhaltung bei Ereignissen aus dem Wertschöpfungsnetzwerk geprüft. Hierfür eignen sich die Kenngrößen des Durchlaufdiagramms eines Produktionssystems (vgl. Abschnitt 2.2.2), kombiniert mit einer Information zur Termintreue der Lose (vgl. Abbildung 5.13). Die Abgangskurve wird hierfür anhand der Differenz von Plan-Flussfaktor und Ist-Flussfaktor unterteilt. Für $FF_{Ist} > FF_{Plan}$ ist das Los gegenüber seinem geplanten Liefertermin zu spät, für $FF_{Ist} < FF_{Plan}$ zu früh und für $FF_{Ist} = FF_{Plan}$ pünktlich. Treffen auf dieses Produktionssystem Ereignisse aus dem Wertschöpfungsnetzwerk, steigt der Anteil der zu späten Lose, da die Produktions-

steuerung diese Ereignisse nicht effektiv mitgieren kann – die Liefertermineinhaltung wird geringer. Das gelernte Entscheidungsmodell eignet sich dann für die Anwendung, wenn der Anteil der zu späten Lose weniger stark steigt und die Liefertermineinhaltung damit gegenüber dem Fall ohne eingreifendes Entscheidungsmodell verbessert werden kann (vgl. Abbildung 5.14).

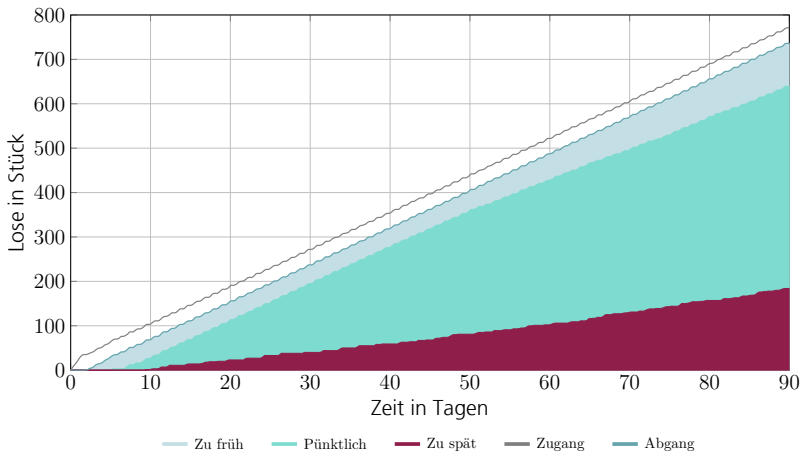


Abbildung 5.13: Exemplarisches Durchlaufdiagramm eines Produktionssystems

Im letzten Teilschritt wird anhand der Ergebnisse der Evaluation über das weitere Vorgehen entschieden (vgl. Abbildung 5.15). Ist die Evaluation erfolgreich, werden die Modellparameter M_{t+1} zur Vorbereitung des Produktivbetriebs an Schritt 12 übergeben. Bei einer nicht erfolgreichen Evaluation können die Belohnungsfunktion R sowie die initialen statischen und dynamischen Modellparameter M_0 für die nächste Epoche angepasst werden. Die derzeitigen Modellparameter M_{t+1} werden hierfür mitsamt der detaillierten Analyse der Modellevaluation an Schritt 6 übergeben, um die Methode anschließend erneut zu durchlaufen. Kann die Evaluation auch nach einer Vielzahl von Episoden nicht erfolgreich abgeschlossen werden, ist zu prüfen, welche Einschränkungen vorgenommen werden können, um die Freiheitsgrade des Entscheidungsmodells zu begrenzen und damit dessen Performanz zu verbessern. Die Einschränkungen beziehen sich dabei insbesondere auf den Ereignisraum E , den Zustandsraum S und den Aktionsraum A . Hierfür wird das Design ab Schritt 1 erneut durchlaufen, bevor wieder mit dem Training begonnen wird.

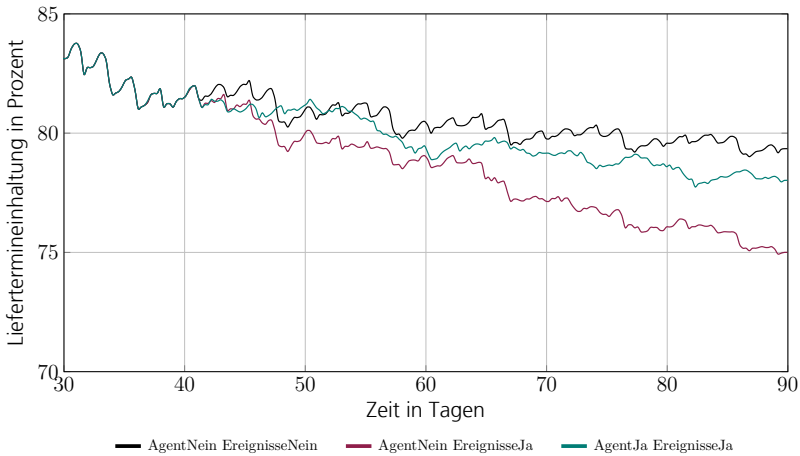


Abbildung 5.14: Exemplarische Liefertermineinhaltung eines Produktionssystems

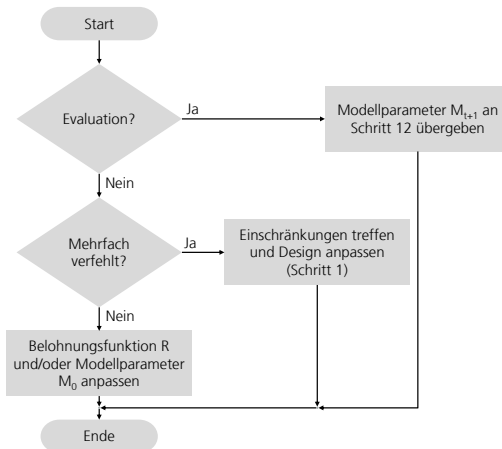


Abbildung 5.15: Prüfung der Modellevaluation im Training (Notation nach DIN 66001)

5.3.2.3 Phase III: Applikation

An das Design schließt die Überführung in den Produktivbetrieb an. Hierfür wird zuerst der Betrieb initialisiert, anschließend folgt eine iterative Schleife mit der Ausführung einer Aktion, der Berechnung der Belohnung, der Anpassung der Modellparameter und der Evaluation des Modells (vgl. Abbildung 5.16). Dies findet kontinuierlich und parallel zur Produktion statt.

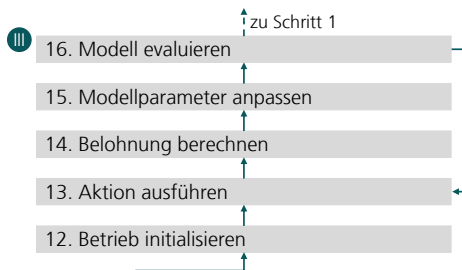


Abbildung 5.16: Phase III: Applikation der Methode zur Gestaltung des adaptiven Regelkreises für ein integriertes Wertschöpfungsnetzwerk

Schritt 12: Betrieb initialisieren

Zu Beginn der Applikation im zwölften Schritt muss der Betrieb initialisiert werden, indem das Entscheidungsmodell für den Produktivbetrieb vorbereitet und überführt wird (vgl. Tabelle 5.13). Dieser Schritt wird auch als sogenanntes Deployment bezeichnet. In dessen Rahmen werden gegebenenfalls noch fehlende Schnittstellen implementiert, die initialen Modellparameter für den Produktivbetrieb definiert und anschließend der Agent für den Betrieb initialisiert.

Bevor das trainierte Entscheidungsmodell in Form des Agenten in den Produktivbetrieb überführt werden kann, müssen Schnittstellen geschaffen sein, über welche der Agent mit dem Produktionssystem interagiert. Im ersten Teilschritt ist daher zu prüfen, ob alle in Schritt 5 beschriebenen Schnittstellen zu relevanten Systemen der Produktions-IT-Landschaft bereits implementiert sind. Nach Beendigung des Trainings interagiert der Agent nicht länger mit einem Simulationsmodell, sondern mit dem realen Produktionssystem, welches

Tabelle 5.13: Schritt 12: Betrieb initialisieren

Ziel	Vorbereitung und Überführung des Entscheidungsmodells in den Produktivbetrieb.
Eingabe (Daten)	Modellparameter M_{t+1} (von Schritt 11)
Ausgabe (Daten)	Modellparameter $M_{Prod,0}$ und Zustand S_0 (zu Schritt 13)
Eingabe (Wissen)	Schnittstellen zu relevanten Systemen der Produktions-IT-Landschaft (von Schritt 5)
Ausgabe (Wissen)	–
Teilschritte	Schnittstellen implementieren → initiale Modellparameter definieren → Agent initialisieren

naturgemäß keine einheitliche Schnittstelle aufweist. Entsprechend werden hierfür die in Schritt 5 definierten Schnittstellen zu relevanten Systemen der Produktions-IT-Landschaft benötigt – einerseits als Quellsysteme zum Auslesen von Daten (für den Ereignisraum E und den Zustandsraum S), andererseits in Form des MES auch als Zielsystem (für den Aktionsraum A). Damit wird eine neue Umgebung geschaffen, über welche der Agent mit dem Produktionssystem interagieren kann (vgl. Abbildung 5.17). Noch nicht implementierte Schnittstellen müssen daher zu diesem Zeitpunkt anhand der Spezifikation aus Schritt 5 in den entsprechenden IT-Systemen implementiert werden.

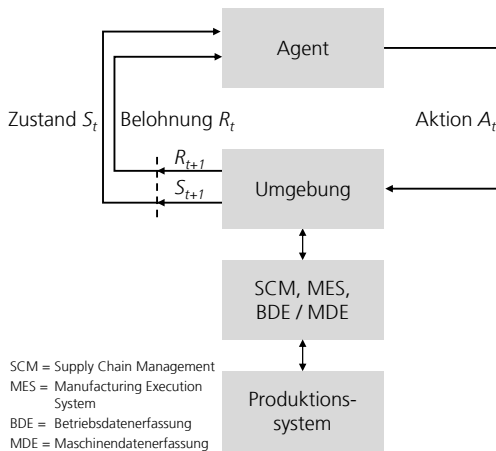


Abbildung 5.17: Interaktion eines Agenten mit einem Produktionssystem über die Abstraktionsschicht der Umgebung

Im zweiten Teilschritt müssen die initialen Modellparameter für den Produktivbetrieb $M_{Prod,0}$ festgelegt werden. Grundlage hierfür sind die aus Schritt 11 des Trainings nach positiver Evaluation übergebenen Modellparameter M_{t+1} . Die initialen Modellparameter für den Produktivbetrieb setzen sich, analog zu den Modellparametern des Trainings, ebenfalls aus Agentenparametern, Hyperparametern und Trainingsparametern zusammen. Die Trainingsparameter werden jedoch nicht weiter benötigt, da der Agent kontinuierlich und ohne Abbruchzeitpunkt mit dem Produktionssystem interagiert und dieses beeinflusst. Andere Modellparameter, wie beispielsweise die Lernrate, müssen angepasst werden, in diesem Fall kleiner und konstant gehalten werden. Damit kann kontrolliert werden, dass der Agent im Betrieb zwar kontinuierlich lernt und sich an veränderte Gegebenheiten anpassen kann, jedoch gleichzeitig stabil bleibt und nicht durch einzelne, spezielle Zustands-Aktions-Kombinationen negativ beeinflusst wird. Alle Modellparameter, auch die im Betrieb nicht weiter benötigten Trainingsparameter, werden dennoch in die initialen Modellparameter $M_{Prod,0}$ und damit auch in die Modellparameter der nachfolgenden Zustände $M_{Prod,t}$ integriert. Im Falle eines notwendigen Redesigns oder Neutrainings (vgl. Schritt 16) stehen diese dann als Grundlage für die Weiterentwicklung zur Verfügung.

Im dritten Teilschritt wird der Agent mit den Modellparametern $M_{Prod,0}$ initialisiert. Des Weiteren wird mittels der implementierten Schnittstellen über die Umgebung der Zustand S_0 geladen. Dieser initiale Zustand entspricht dabei im Gegensatz zum Training keinem vordefinierten Zeitpunkt, sondern jenem Zeitpunkt und damit Zustand des Produktionssystems, zu welchem der Betrieb des ARIW und damit des Entscheidungsmodells des Agenten gestartet wird. Schritt 13 bis Schritt 15 werden in der Regel automatisch und iterativ durch den Agenten durchlaufen, ohne dass ein menschliches Eingreifen notwendig ist.

Schritt 13: Aktion ausführen

Der dreizehnte Schritt beinhaltet die Ermittlung und Ausführung der, abhängig vom derzeitigen Zustand, bestmöglichen Aktion (vgl. Tabelle 5.14). Hierfür wird zuerst der derzeitige Zustand ermittelt, anschließend in Abhängigkeit von diesem eine Aktion ausgeführt und zuletzt der nachfolgende Zustand ermittelt.

Tabelle 5.14: Schritt 13: Aktion ausführen

Ziel	Bestmögliche Aktion in Abhängigkeit vom derzeitigen Zustand ermitteln und ausführen.
Eingabe (Daten)	Modellparameter $M_{Prod,0}$ und Zustand S_0 (von Schritt 12) <i>Opt: Modellparameter $M_{Prod,t+1}$ und Zustand S_{t+1} (von Schritt 16)</i>
Ausgabe (Daten)	Modellparameter $M_{Prod,t}$ und Zustand S_{t+1} (zu Schritt 14)
Eingabe (Wissen)	–
Ausgabe (Wissen)	–
Teilschritte	Zustand ermitteln → Aktion ausführen → Nachfolgenden Zustand ermitteln

Im ersten Teilschritt muss der Agent den Zustand S_t ermitteln. Zum Zeitpunkt t_0 ist dies der Zustand S_0 (von Schritt 12), zu allen nachfolgenden Zeitpunkten der Zustand S_{t+1} (von Schritt 16). Das Produktionssystem wird dabei über die entsprechenden Systeme der Produktions-IT-Landschaft abstrahiert, diese wiederum über die Umgebung, welche dezidiert den Zustandsvektor zur Verfügung stellt.

Der Algorithmus des Agenten wählt anschließend im zweiten Teilschritt eine Aktion aus. Im Gegensatz zum Training wird die Aktion $A_t(S_t)$ mit $A_t \in \mathcal{A}$ dabei rein anhand der Strategie π ausgewählt, eine weitere Exploration findet nicht statt. Umgesetzt wird die ausgewählte Aktion über die Anpassung von Werten im MES mittels der Umgebung, darüber dann wiederum im Produktionssystem.

Im dritten Teilschritt wird der nachfolgende Zustand ermittelt. Zum Zeitpunkt $t + \Delta t$ wird dabei der Zustand S_{t+1} ermittelt, wobei das Produktionssystem natürlich nicht stillsteht, sondern auch parallel zu Schritt 13 bis Schritt 16 produziert. Voraussetzung ist deshalb, dass die benötigte Zeitdauer des Agenten zur Auswahl und Umsetzung einer Aktion sehr viel geringer ist als Δt . Durch die Trennung in Training und Applikation ist dies gegeben. Für die Ermittlung der Belohnung im nächsten Schritt bilden der Zustand S_t , die ausgeführte Aktion A_t und der nachfolgende Zustand S_{t+1} die Grundlage.

Schritt 14: Belohnung berechnen

Im vierzehnten Schritt wird abhängig vom Zustand und der Aktion die zugehörige Belohnung berechnet (vgl. Tabelle 5.15).

Tabelle 5.15: Schritt 14: Belohnung berechnen	
Ziel	Belohnung berechnen, um dem Agenten eine Bewertung der durchgeführten Aktion zu ermöglichen.
Eingabe (Daten)	Modellparameter $M_{Prod,t}$ und Zustand S_{t+1} (von Schritt 13)
Ausgabe (Daten)	Modellparameter $M_{Prod,t}$, Zustand S_{t+1} und Belohnung R_t (zu Schritt 15)
Eingabe (Wissen)	–
Ausgabe (Wissen)	–
Teilschritte	Belohnung berechnen

Zur Berechnung der Belohnung wird die in Schritt 6 definierte Belohnungsfunktion herangezogen. Für die Variablen werden dabei die aktuellen Werte aus den Zuständen S_t und S_{t+1} eingesetzt, um den numerischen Wert der Belohnung R zu berechnen. Die Vorgehensweise ist in Schritt 9 beschrieben.

Schritt 15: Modellparameter anpassen

Das Lernen parallel zum Betrieb und die darauf basierende Anpassung der dynamischen Modellparameter (vgl. Schritt 7) findet im fünfzehnten Schritt statt (vgl. Tabelle 5.16). Zuerst findet hierfür das eigentliche Lernen statt, anschließend werden die dynamischen Modellparameter angepasst.

Der Algorithmus lernt zur Weiterentwicklung seines Entscheidungsmodells basierend auf dem Zustand S_t , der ausgeführten Aktion A_t , der Belohnung R_t und dem nachfolgenden Zustand S_{t+1} . Analog zu den Ausführungen in Schritt 10 werden hierfür die Modellgewichte θ angepasst. Im Training dynamische Modellparameter wurden in Schritt 12 als im Betrieb konstante Parameter festgelegt, sodass keine weitere Anpassung stattfindet. Anschließend kann mit der Evaluation des Modells in Schritt 16 fortgefahren werden.

Tabelle 5.16: Schritt 15: Modellparameter anpassen

Ziel	Lernvorgang des Algorithmus ausführen und basierend darauf die dynamischen Modellparameter anpassen.
Eingabe (Daten)	Modellparameter $M_{Prod,t}$, Zustand S_{t+1} und Belohnung R_t (von Schritt 14)
Ausgabe (Daten)	Modellparameter $M_{Prod,t+1}$, Zustand S_{t+1} und Belohnung R_t (zu Schritt 16)
Eingabe (Wissen)	–
Ausgabe (Wissen)	–
Teilschritte	Entscheidungsmodell lernen

Schritt 16: Modell evaluieren

Die kontinuierliche Evaluation des Entscheidungsmodells parallel zum Betrieb findet in Schritt sechzehn statt (vgl. Tabelle 5.17). Hierfür wird das Modell kontinuierlich statistisch und kontextuell evaluiert. Anschließend wird die Evaluation geprüft, um über das weitere Vorgehen zu entscheiden. Kontinuierliche Evaluation bedeutet hier, dass das Ergebnis des Entscheidungsmodells nach jedem Schritt (im Training passiert dies nach jeder Episode) geprüft wird, um bei Abweichungen jederzeit eingreifen zu können. Entsprechend muss die Evaluation automatisiert ablaufen und darf nur bei negativem Ergebnis ein menschliches Eingreifen erfordern.

Tabelle 5.17: Schritt 16: Modell evaluieren

Ziel	Prüfung, ob mit den ermittelten Modellparametern weiterhin der Modellzweck erreicht wird.
Eingabe (Daten)	Modellparameter $M_{Prod,t}$, Zustand S_{t+1} und Belohnung R_t (von Schritt 15)
Ausgabe (Daten)	Modellparameter $M_{Prod,t+1}$ und Zustand S_{t+1} (zu Schritt 13)
Eingabe (Wissen)	–
Ausgabe (Wissen)	<i>Opt: Detaillierte Analyse der Modellevaluation (zu Schritt 1)</i>
Teilschritte	Statistisch evaluieren → Kontextuell evaluieren → Evaluation prüfen

Für die statistische Evaluation kann wiederum auf den Verlauf der Belohnung über die Zeit zurückgegriffen werden, um kontinuierlich die Stabilität des ARIW zu analysieren. Im Unterschied zum Training geht es jedoch nicht darum, die Konvergenz zu prüfen – ein in Schritt 11 positiv evaluiertes Entscheidungsmodell ist hinsichtlich der Belohnung bereits

konvergiert. Vielmehr ist es notwendig zu prüfen, dass die erreichte Belohnung nicht beginnt, zunehmend von diesem Wert abzuweichen. Kurzzeitige Abweichungen können dabei auf unbekannte Zustände hindeuten, die trotz der Generalisierung des Entscheidungsmodells nicht den bestmöglichen Aktionen zugeordnet werden konnten. Insbesondere zunehmende längerfristige Abweichungen lassen vermuten, dass Veränderungen im Produktionssystem stattgefunden haben, welche durch das Entscheidungsmodell nicht ausreichend gelernt wurden.

Überprüft werden kann dies mittels der kontextuellen Evaluation des Entscheidungsmodells im zweiten Teilschritt. Hierfür wird der Anteil der zu späten, pünktlichen und zu frühen Lose an der Abgangskurve des Produktionssystems bewertet. Steigt der Anteil der zu späten Lose kontinuierlich, sinkt dadurch die Liefertermineinhaltung. Die Aufgabe des Entscheidungsmodells zur Verbesserung der Liefertermineinhaltung bei Ereignissen aus dem Wertschöpfungsnetzwerk wird dann mutmaßlich nicht mehr erfüllt.

Anhand der Ergebnisse der Evaluation wird im dritten Teilschritt über das weitere Vorgehen entschieden (vgl. Abbildung 5.18). Ist die Evaluation erfolgreich, das Entscheidungsmodell erfüllt also weiterhin seinen Modellzweck und weicht im Betrieb nicht davon ab, werden die Modellparameter $M_{Prod,t+1}$ an Schritt 13 übergeben. Sie bilden dann die Grundlage für die Auswahl der Aktion zum nächsten Zeitpunkt. Ist die Evaluation nicht erfolgreich, wird die Methode ab Schritt 1 erneut durchlaufen. Entsprechend muss hier eine Benachrichtigung an einen menschlichen Entscheider stattfinden, da insbesondere beim Design, jedoch auch beim Training, einige Teilschritte durch menschliches Handeln auszuführen sind. Bei der Bearbeitung der Schritte ist dann zu entscheiden, ob auch das Design angepasst werden muss oder die Veränderungen auch durch ein reines Neutraining ausreichend berücksichtigt werden können.

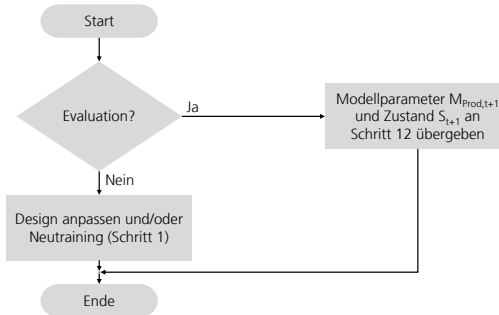


Abbildung 5.18: Prüfung der Modellevaluation im Betrieb (Notation nach DIN 66001)

5.3.3 Architektur

Damit der ARIW die Produktionssteuerung effektiv unterstützen und damit bestehende Produktionsregelungen ergänzen kann (vgl. Abschnitt 5.3.1), muss er nahtlos in die Produktions-IT-Landschaft eines Unternehmens integriert werden (vgl. Anforderung S2). Der im Rahmen der vorliegenden Arbeit entwickelte und in Abbildung 5.19 dargestellte modulare Ansatz ist durch eine Referenzarchitektur für die Ereignisverarbeitung in der PPS inspiriert (vgl. Abschnitt 2.2.4). Diese ist dahingehend angepasst, eine Verarbeitung von Ereignissen aus dem Wertschöpfungsnetzwerk auf der Ebene der Produktionssteuerung zu ermöglichen und eine geeignete Basis für die Integration eines RL-basierten Reglers zu schaffen. Der Aufbau der Architektur orientiert sich an der Automatisierungspyramide mit der Fertigungsebene, der Fertigungsleitebene und der Unternehmensleitebene, um die in Schritt 5 der Methode (vgl. Abschnitt 5.3.2.1) definierten Schnittstellen zu implementieren. Die einzelnen IT-Systeme sind dabei nicht als monolithische IT-Systeme zu verstehen, sondern können und werden zukünftig vermehrt durch serviceorientierte Architekturen repräsentiert (Bauer et al. 2017, S. 200–201). Als Querschnittsfunktionen werden die Ebenen um eine Datenschicht und einen Digitalen Zwilling ergänzt. Um Medienbrüche zu vermeiden und die Kommunikation zu automatisieren, sind alle Schnittstellen als Anwendungsprogrammierschnittstellen, fortan mit dem englischen Akronym Application Programming Interface (API) abgekürzt, ausgeführt (vgl. Anforderung S1). Teilweise sind darüber hinaus Mensch-Maschine-Schnittstellen für die Nutzung vorgesehen.

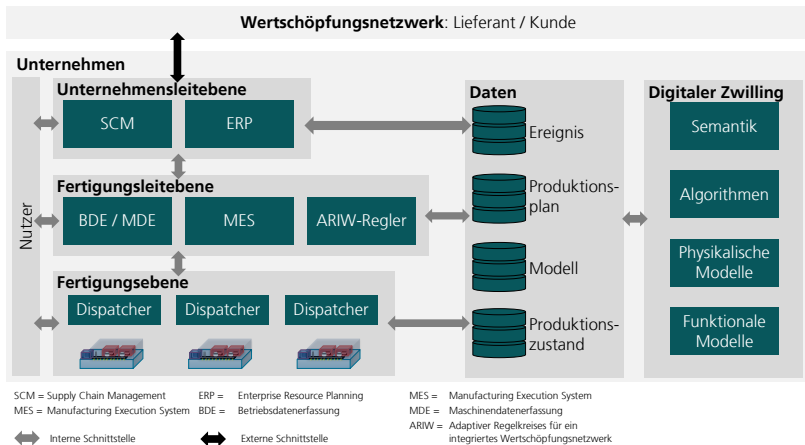


Abbildung 5.19: Architektur zur Einbettung des adaptiven Regelkreises für ein integriertes Wertschöpfungsnetzwerk in die Produktions- IT -Landschaft (eigene Darstellung in Anlehnung an Bauer et al. 2021a, S. 11)

Die Fertigungsebene besteht aus den Betriebsmitteln, also den Maschinen und Transportsystemen, deren Sensoren, Aktoren und IT-Schnittstellen. Da neuere Ansätze der Produktionssteuerung das Dispatching als dezentrale Aufgabe an den Maschinengruppen sehen, wird dies in der Architektur im Gegensatz zur klassischen Automatisierungspyramide entsprechend berücksichtigt. Es wird dennoch davon ausgegangen, dass der ARIW-Regler über das MES mit der Fertigungsebene interagiert und nicht mit den einzelnen dezentralen Dispatchern.

Während die Fertigungsebene hauptsächlich aus physischen Komponenten besteht, sind die Fertigungsleitebene und die Unternehmensleitebene IT-basiert und damit für die Datenverarbeitung zuständig. Die Fertigungsleitebene besteht aus BDE und MDE, um eine datenbasierte Verbindung zur Fertigungsebene herzustellen. Diese Verbindung muss modular, flexibel und skalierbar sein, was technisch durch eine Integrationsschicht wie einen Manufacturing Service Bus erreicht werden kann. Darüber hinaus ermöglicht solch ein Manufacturing Service Bus eine Integration mit cyber-physischen Systemen innerhalb einer intelligenten Fabrik (Schel et al. 2018, S. 182). Zusätzlich wird das MES als zentrale Einheit der Produktionssteuerung einbezogen. Der ARIW-Regler komplementiert die Fertigungsleit-

ebene und integriert den ARIW mit seinem RL-basierten Regler (vgl. Abschnitt 5.3.1) in die Architektur.

Auf der Unternehmensleitebene sind die Systeme zur Steuerung des Unternehmens, ERP, und des Wertschöpfungsnetzwerks, SCM, zu finden. Folglich stellt die Unternehmensleitebene auch die maßgebliche Schnittstelle zum Wertschöpfungsnetzwerk dar.

Die Entscheidung des ARIW, wie auf Ereignisse aus dem Wertschöpfungsnetzwerk reagiert werden soll, wird auf der Fertigungsleitebene durch den ARIW-Regler getroffen. Dabei werden Daten aus der Unternehmensleitebene (Ereignisse) und der Fertigungsebene (Zustände) explizit berücksichtigt und in die Entscheidung einbezogen. Der Entscheidungsprozess wird darüber hinaus durch die Datenschicht und den Digitalen Zwilling unterstützt. Über die Architektur klassischer ereignisbasierter IT-Systeme hinausgehend, benötigt ein lernendes System wie der ARIW in der Datenschicht neben den Ereignissen und der Historie der Produktionspläne und Produktionszustände auch seine Modellparameter. Diese Daten werden, mit Ausnahme der Modellparameter, von den bestehenden Systemen der Produktions-IT-Landschaft gespeist. Der Digitale Zwilling adressiert den Bedarf eines RL-basierten Agenten, eine sichere Umgebung für das Lernen zu schaffen und gleichzeitig Daten und Systemverhalten so nah wie möglich an das reale Produktionssystem anzulehnen (vgl. Schritt 5 in Abschnitt 5.3.2.1). Der Digitale Zwilling verwendet deshalb Semantik und Algorithmen, um die Daten aus der Datenschicht für Simulationsmodelle vorzubereiten, welche die physikalischen und funktionalen Eigenschaften des Produktionssystems abbilden.

Wie in Abbildung 5.20 dargestellt, besteht der ARIW-Regler aus verschiedenen Subkomponenten. Welche dies sind und wie diese mit bestehenden Systemen der Produktions-IT-Landschaft interagieren, hängt davon ab, ob sich der ARIW-Regler im Training (vgl. Abschnitt 5.3.2.2) oder der Applikation (vgl. Abschnitt 5.3.2.3) befindet. Beiden gemein ist der Agent mit seinem Entscheidungsmodell (vgl. Schritt 7 in Abschnitt 5.3.2.2), der Modellevaluator und die Umgebung. Der Modellevaluator ist dabei jedoch jeweils anders ausgestaltet und zielt entweder auf eine Bewertung des Trainings des Entscheidungsmodells (vgl. Schritt 11 in Abschnitt 5.3.2.2) oder des kontinuierlichen Betriebs (vgl. Schritt 16 in Abschnitt 5.3.2.3) ab. Auch die Umgebung ist jeweils anders ausgestaltet und erlaubt entweder eine Interaktion mit den Systemen der Produktions-IT-Landschaft (vgl. Schritt 12 in Abschnitt 5.3.2.3) oder dem Simulationsmodell (vgl. Schritt 7 in Abschnitt 5.3.2.2),

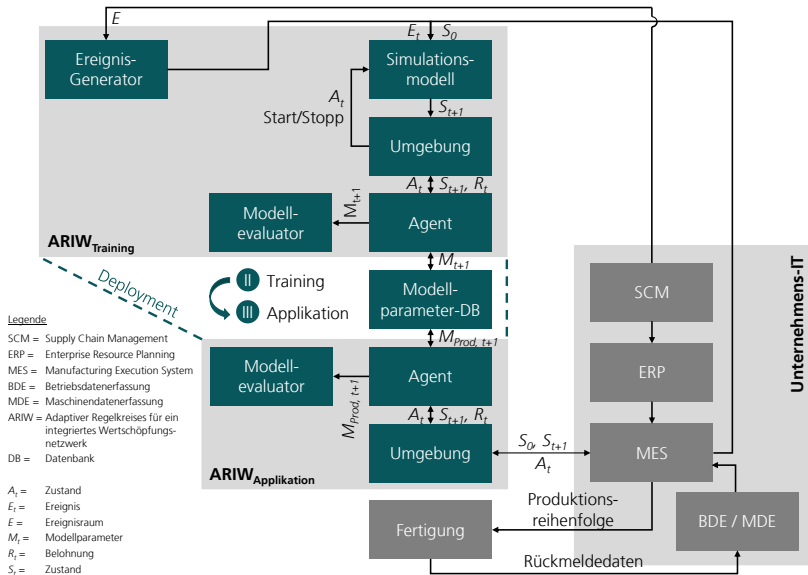


Abbildung 5.20: Architektur des adaptiven Regelkreises für ein integriertes Wertschöpfungsnetzwerk

welches in Abbildung 5.19 einem funktionalen Modell des Digitalen Zwillings entspricht. Die Modellparameterdatenbank des ARIW-Reglers findet sich in derselben Abbildung in der Datenschicht wieder und speichert die Parameter, welche in Schritt 10 bzw. Schritt 15 der Methode (vgl. Abschnitt 5.3.2) bestimmt werden.

5.4 Zwischenfazit

Aufbauend auf den in Abschnitt 4 definierten Anforderungen wurde im zurückliegenden Abschnitt die lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung konzipiert. Die hierfür erforderlichen Artefakte sind der Regelkreis, die Methode und die Architektur. Diese drei Artefakte wirken zusammen, um die Liefertermineinhaltung bei Ereignissen aus dem Wertschöpfungsnetzwerk zu verbessern und hierfür Entscheidungen hinsichtlich Korrekturmaßnahmen zur Priorisierung von Losen eines Produktionssystems zu treffen (vgl. Abschnitt 5.2). Damit konnte die dritte Teilforschungsfrage nach dem Aufbau des Konzepts einer lernenden ereignisbasierten

Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung beantwortet werden.

Der Regelkreis (vgl. Abschnitt 5.3.1) beschreibt in Form des ARIW das logische Zielbild zur Integration von Wertschöpfungsnetzwerk und Produktionssteuerung für die Mitigation der Auswirkungen von Ereignissen aus dem Wertschöpfungsnetzwerk in einem Produktionssystem. Produktion und Produktionssteuerung werden dabei als Regelstrecke betrachtet, um die Regelgröße der Liefertermine durch entsprechende Handlungsalternativen zu beeinflussen. Die Entscheidung über die Auswahl von Handlungsalternativen und damit über Korrekturmaßnahmen für Ereignisse aus dem Wertschöpfungsnetzwerk trifft der ARIW-Regler. Durch den Einsatz von RL wird dieser adaptiv und lernend, indem die Strategie für Entscheidungen datenbasiert aus Beobachtungen der Umgebung gelernt wird.

Die Methode (vgl. Abschnitt 5.3.2) beschreibt in 16 diskreten Schritten und gegliedert in die drei Phasen Design, Training und Applikation die Gestaltung des ARIW. Eine besondere Bedeutung kommt dabei der Design-Phase zu, in welcher das Problem und die Zielsetzung mathematisch beschrieben sowie Voraussetzungen geschaffen werden, um die Methode auf einen konkreten Anwendungsfall applizieren zu können. Mit der Definition von Ereignis-, Zustands- und Aktionsraum werden dabei einerseits die Charakteristik der komplexen Werkstattfertigung abgebildet sowie andererseits der relevante Betrachtungsraum und der Aktionsraum zur Einwirkung auf diskrete Aufträge integriert. Aufbauend auf dem Design wird im Training das Entscheidungsmodell für den Anwendungsfall anhand der Belohnungsfunktion gelernt und in der Applikation in den Produktivbetrieb überführt.

Die Architektur ermöglicht die Einbettung des ARIW in eine bestehende Produktions-IT-Landschaft eines Unternehmens (vgl. Abschnitt 5.3.3). Entlang der Fertigungsebene, der Fertigungsleitebene und der Unternehmensleitebene, ergänzt um die Querschnittsfunktionen der Datenschicht und des Digitalen Zwillings, bildet die Architektur die Komponenten des ARIW und deren Interaktion mit IT-Systemen für das Management von Wertschöpfungsnetzwerken und der PPS ab. Die modulare, flexible und skalierbare Gestaltung ermöglicht die Einbindung zusätzlicher Elemente sowie die schrittweise Einführung des ARIW. Durch die konsequente Nutzung von APIs wird das System automatisiert, indem maschinenlesbare Schnittstellen Medienbrüche, Übertragungsfehler und Übertragungsverzögerungen vermeiden.

Auf Grundlage des entwickelten Konzepts kann in Abschnitt 6 die Applikation der lernenden ereignisbasierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung erfolgen.

6 Applikation der lernenden ereignisbasierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung

Die Anwendbarkeit und der Nutzen der in Abschnitt 5 entwickelten lernenden ereignisbasierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung soll anhand deren Applikation aufgezeigt werden. Im ersten Schritt wird hierfür die Charakteristik der Anwendungsumgebung Halbleiterfertigung beschrieben (vgl. Abschnitt 6.1). Nachfolgend wird im zweiten Schritt die Implementierung dargestellt (vgl. Abschnitt 6.2). Die Anwendung und Validierung in der Halbleiterfertigung findet im dritten Schritt statt (vgl. Abschnitt 6.3).

6.1 Charakteristik der Anwendungsumgebung der Halbleiterfertigung

Das Fertigungsprinzip der Werkstattfertigung eignet sich insbesondere für die Einmal-, Wiederhol- und Variantenfertigung. Durch die komplexe Werkstattfertigung, welche einen Einsatz unter Bedingungen der Serien- und Massenfertigung erlaubt, kommt dem Fertigungsprinzip eine steigende Bedeutung zu (vgl. Abschnitt 2.1.1). Damit wird das Fertigungsprinzip für weitere Branchen interessant, beispielsweise die Automobilindustrie (Bauernhansl 2017a, S. 17) oder die Halbleiterindustrie (Waschneck et al. 2017, S. 3).

Die Halbleiterherstellung, die Herstellung von miniaturisierten, integrierten Schaltungen bestehend aus tausenden von Komponenten, ist zugleich die industriell wichtigste Anwendung der komplexen Werkstattfertigung (Waschneck 2020, S. 24–25; Waschneck et al. 2016, S. 2; Mönch et al. 2013, S. 11; Mason et al. 2002, S. 248). Auch das Wertschöpfungsnetzwerk der Halbleiterindustrie ist global verteilt und komplex. Wenngleich

die Halbleiterindustrie zweifelsohne eine High-Tech-Industrie ist, so ist gerade der Prozess zur Reaktion auf Ereignisse aus dem Wertschöpfungsnetzwerk oftmals durch viele Beteiligte, häufige Medienbrüche und eine Vielzahl manueller Entscheidungen auf diversen Stufen des Prozesses gekennzeichnet (Bauer et al. 2021a, S. 13–14). Die lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung wird deshalb für die Halbleiterfertigung, eingebettet in das Wertschöpfungsnetzwerk der Halbleiterindustrie, angewendet. Zur näheren Charakterisierung werden nachfolgend das Wertschöpfungsnetzwerk (vgl. Abschnitt 6.1.1) und die Fertigung (vgl. Abschnitt 6.1.2) der Halbleiterindustrie eingehend beschrieben.

6.1.1 Das Wertschöpfungsnetzwerk der Halbleiterindustrie

Die Halbleiterherstellung gliedert sich typischerweise in die vier Schritte Wafer Fabrication, Wafer Test, Assembly und Final Test (vgl. Abbildung 6.1). Ausgangspunkt hierfür ist ein Roh-Wafer, Ergebnis sind fertige und getestete Chips der integrierten Schaltungen (Mönch et al. 2018, S. 2; Mönch et al. 2013, S. 12; Uzsoy et al. 1992, S. 48). In den ersten beiden Schritten, die als Front-End-of-Line (Front-End) bezeichnet werden, werden multiple Schichten auf einen Wafer aufgebracht, um die sogenannten Dies herzustellen und anschließend zu testen. Dieser Teil des Herstellungsprozesses ist der technologisch anspruchsvollste und kapitalintensivste Schritt (Uzsoy et al. 1992, S. 48). Das Back-End-of-Line (Back-End) umfasst das Montieren der Dies zu Chips und deren finalen Test (Mönch et al. 2013, S. 25). Das Front-End ist dabei oftmals in hochindustrialisierten Ländern angesiedelt, das Back-End aufgrund des höheren manuellen Aufwands meist in Ländern mit niedrigen Lohnkosten (Mönch et al. 2013, S. 12). Zwischen Front-End und Back-End befindet sich die sogenannte Die Bank, welche mit einem Lager getesteter Wafer zum Abfedern von Nachfrageschwankungen dient (Uzsoy et al. 1992, S. 50). Nach der Terminologie aus Abschnitt 2.1 ist der gesamte Herstellungsprozess, dargestellt in Abbildung 6.1, als Produktion zu bezeichnen. Das Front-End beinhaltet die Fertigung, die Montage findet im Back-End statt.

Das Wertschöpfungsnetzwerk der Halbleiterindustrie ist charakterisiert durch eine starke Economy of Scale, globale Verteilung, kurze Produktlebenszyklen und allgegenwärtige Unsicherheit (Mönch et al. 2018, S. 4–5). Während erstere zur statischen Komplexität des

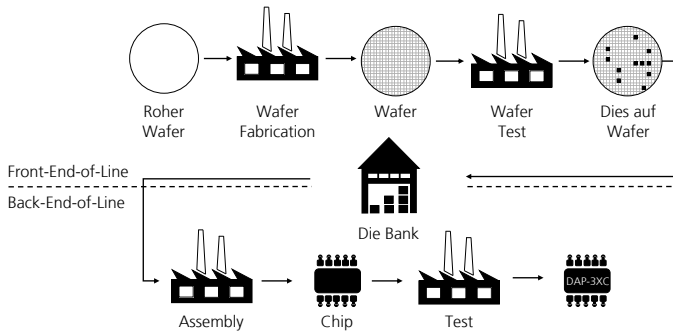


Abbildung 6.1: Stufen der Halbleiterherstellung (eigene Darstellung in Anlehnung an Mönch et al. 2018, S. 2; Mönch et al. 2013, S. 12; Uzsoy et al. 1992, S. 50)

Wertschöpfungsnetzwerks beiträgt, führen die weiteren Kriterien zu einer zunehmenden dynamischen Komplexität (vgl. Abschnitt 2.1.2). Die Halbleiterindustrie erscheint deshalb oftmals als „getrieben durch deterministisches Chaos“ (eigene Übersetzung von Kallrath & Maindl 2006, S. 107). Insbesondere in solchen Wertschöpfungsnetzwerken nimmt die Planungsgenauigkeit mit zunehmendem Zeithorizont deutlich ab (Schuh et al. 2013, S. 482). Eine Studie mit 52 multinationalen Halbleiterherstellern mit einem Jahresumsatz von über 400 Mio. USD hat gezeigt, dass unzureichende Prognosen und deren Auswirkungen das größte Problem im Wertschöpfungsnetzwerk darstellen, was zu unnötig hohen Lagerbeständen und unzuverlässigen Lieferungen führt (Kremers 2011, S. 3–6). Auch wenn moderne Ansätze verbesserte Prognosen ermöglichen, weisen sie weiterhin Verzerrungen und Fehler auf (Chien et al. 2020, S. 11–16). Die daraus resultierenden Abweichungen zwischen Prognosen und tatsächlichen Bedarfen führen zu vielzähligen Ereignissen, die entsprechende Reaktionen erfordern.

6.1.2 Die Fertigung in der Halbleiterindustrie

Der Fertigungsprozess der Wafer Fabrication im Front-End der Halbleiterindustrie umfasst sechs maßgebliche Prozessschritte, die in wiedereintretenden Prozessflüssen mehrfach durchlaufen werden (vgl. Abbildung 6.2). Bei der *Oxidation/Diffusion* entsteht bzw. wird eine Materialschicht auf einen Wafer aufgebracht. Der Hochtemperaturprozess der Diffusion wird meist in Chargen ausgeführt und hat die längste Prozessdauer aller beschriebenen

Schritte. Bei der *Deposition* werden dielektrische oder metallische Schichten auf den Wafer abgeschieden. Hierfür kommen Verfahren der chemischen oder physikalischen Gasphasenabscheidung, Exptaxie oder Metallisierung zum Einsatz. Die *Fotolithographie* durchläuft jeder Wafer ein Mal pro Schicht. Da diese Anlagen die teuersten und deshalb nur in begrenzter Anzahl verfügbar sind, entsteht hier häufig ein Engpass, also der durchsatzbegrenzende Prozessschritt des Front-End. Im ersten Schritt der Fotolithographie wird der Wafer mit einem fotosensitiven Polymerlack beschichtet. Anschließend wird die Schaltkreisstruktur mittels ultravioletter Licht durch eine strukturierte Maske, das sogenannte Reticle, auf den Wafer übertragen. Zuletzt folgt die Entwicklung des Wafers, wobei die polymerisierten Bereiche des Lacks entfernt werden. Beim *Ätzen* werden Bereiche des Wafers entfernt, welche nicht von fotosensitivem Polymerlack bedeckt sind. Hierbei kann zwischen nassem Ätzen mit Flüssigkeiten und trockenem Ätzen unter Zuhilfenahme von Gasen unterschieden werden. Die *Ionenimplantation* dient dazu, partiell elektrische Eigenschaften des Wafers durch Einbringen von Ionen zu verändern. Das Dotierungsmaterial wird in den Bereichen des Wafers abgelagert, die geätzt wurden. Im Schritt der *Planarisierung* wird die Oberfläche des Wafers gereinigt und geebnet. So wird durch chemisch-mechanische Verfahren die Dicke des Wafers kontrolliert und verringert, bevor eine neue Schicht aufgetragen wird (Mönch et al. 2013, S. 20–21; Mönch et al. 2011, S. 584–585).

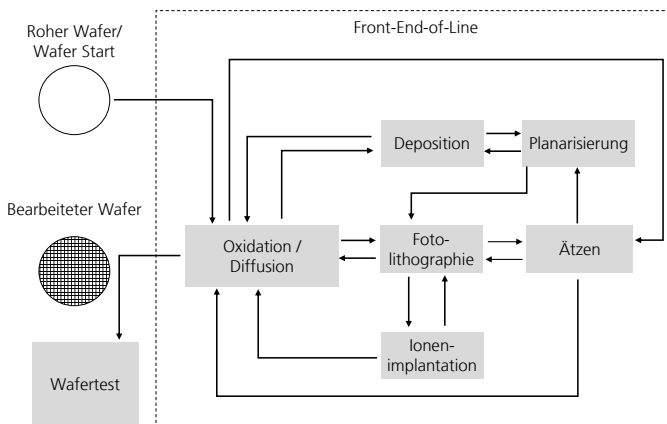


Abbildung 6.2: Prozessschritte der Front-End-Fertigung in der Halbleiterindustrie (eigene Darstellung in Anlehnung an Mönch et al. 2013, S. 22)

Insbesondere zwischen Prozessen, die flüssige oder gasförmige Chemikalien beinhalten, bestehen dabei sequenzabhängige Rüstzeiten. So müssen beispielsweise bei der Ionenimplantation je nach zu dotierenden Ionen andere Prozessgase verwendet werden, weshalb bei einem Gaswechsel eine Reinigung der Prozesskammer und eine Wiederauffüllung mit einem neuen Gas notwendig ist (Mason et al. 2002, S. 249). Dazu existieren zwischen den Prozessschritten auch Zeitabhängigkeiten, welche eine maximale Wartedauer vor dem nächsten Arbeitsplatz vorgeben (Waschneck et al. 2016, S. 2).

Wie auch für andere Branchen wird es für die Halbleiterindustrie zunehmend wichtiger, das Produktportfolio schnell an veränderte Kundenanforderungen anzupassen. Die Variantenzahl im Produktmix steigt dadurch an, die Dauer eines Produktlebenszyklus sinkt. Der Fertigungsprozess im Front-End ist dabei sehr aufwändig. Um bis zu 40 Schichten pro Chip herzustellen, sind bis zu 900 Prozessschritte auf mehr als 100 Maschinen notwendig, die teilweise sehr teuer sind und deshalb bestmöglich ausgelastet werden sollen. Die einzelnen Maschinen werden dabei für die verschiedenen Schichten mehrfach durchlaufen, weshalb auch Wafer unterschiedlichen Fertigstellungsgrades um die gleichen Maschinen konkurrieren. Die Durchlaufzeit beträgt insgesamt je nach Produkt zwei bis vier Monate. Bei dieser Durchlaufzeit beträgt der Flussfaktor etwa 3, bei hoch priorisierten Eillosen können Werte von etwa 1,5 erreicht werden. Zur Bearbeitung werden typischerweise 25 Wafer zu einem Los zusammengefasst, die je nach Prozesstyp einzeln oder in Chargen bearbeitet werden. Zwischen den Prozessen existieren deutliche Zeitunterschiede von wenigen Minuten bis hin zu 12 Stunden, wobei dies meist auf Chargen-Prozesse zutrifft, welche in etwa ein Drittel der Fertigung repräsentieren. Dazu wird die Fertigung aufgrund des hohen Kapitalbedarfs für Prototypen und Produktivlose gleichzeitig genutzt. Aufgrund der Dynamik im Fertigungsablauf variiert die Liefertermineinhaltung bei den neun größten Halbleiterherstellern um 40 Prozent, was eine höhere Liefertermineinhaltung zu einem zentralen Anliegen der PPS im Front-End der Halbleiterfertigung macht (Waschneck 2020, S. 31; Mönch et al. 2018, S. 2; Waschneck et al. 2017, S. 7; Mönch et al. 2013, S. 11; Mönch et al. 2011, S. 583–585; Hintz et al. 2011, S. 42; Mason et al. 2002, S. 249; Uzsoy et al. 1992, S. 49–50).

6.2 Implementierung des Systemkonzepts in der Halbleiterfertigung

Ausgehend von der Charakteristik der Anwendungsumgebung Halbleiterfertigung in Abschnitt 6.1 beschreibt die Implementierung des Systemkonzepts dessen Umsetzung als Software für die Anwendung in der Halbleiterfertigung. Hierfür wird nachfolgend auf das Simulationsmodell des betrachteten Produktionssystems der Halbleiterfertigung (vgl. Abschnitt 6.2.1), auf den Regler des ARIW (vgl. Abschnitt 6.2.2) und die Auswahl eines Algorithmus für den ARIW-Regler (vgl. Abschnitt 6.2.3) eingegangen.

6.2.1 Simulationsmodell des Produktionssystems

Für die Implementierung des Simulationsmodells eines Produktionssystems, konkret einer komplexen Werkstattfertigung im Front-End der Halbleiterindustrie, wird anhand der in Abbildung 6.3 dargestellten Vorgehensweise vorgegangen. Der Modellzweck ist dabei die Abbildung eines Produktionssystems zur Simulation der Auswirkungen von Ereignissen aus dem Wertschöpfungsnetzwerk sowie möglicher Korrekturmaßnahmen, welche durch den ARIW-Regler ausgewählt werden. Das Modell beschränkt sich auf die funktionale Modellierung des Front-End einer Halbleiterfertigung (vgl. Abschnitt 6.1.2), physikalische Prozesse werden analog zu den Ausführungen in Abschnitt 5.3.2.1 nicht modelliert. Aufgebaut wird das Modell auf das Referenzmodell¹ Measurement and Improvement of Manufacturing Capacity (MIMAC) von Fowler & Robinson (1995) und Feigin et al. (1994), welches in der Literatur weiter verbreitet und seit Jahren etabliert ist (Kopp et al. 2020, S. 524).

Das Konzeptmodell² setzt sich aus den zwei Klassen der Auftragsfreigabe und der komplexen Werkstattfertigung zusammen, welche sich wiederum in Subklassen gliedern. Parametriert wird das Modell mittels eines Datensatzes an Stammdaten. Der Zustand des modellierten Produktionssystems spiegelt sich in den Bewegungsdaten wider. Zusätzlich ist eine Schnittstelle zur Interaktionen mit externer Software, hier dem ARIW-Regler, vorgesehen (vgl. Abbildung 6.4).

Die Klasse der Auftragsfreigabe steuert den Zugang zum Produktionssystem und besteht aus den Subklassen der Conwip-Steuerung und des Front Opening Unified Pod (FOUP). Die

¹Ein Referenzmodell stellt eine Reihe von vereinheitlichenden Konzepten, Axiomen und Beziehungen innerhalb einer bestimmten Domäne dar, unabhängig von spezifischen Standards, Technologien, Imple-

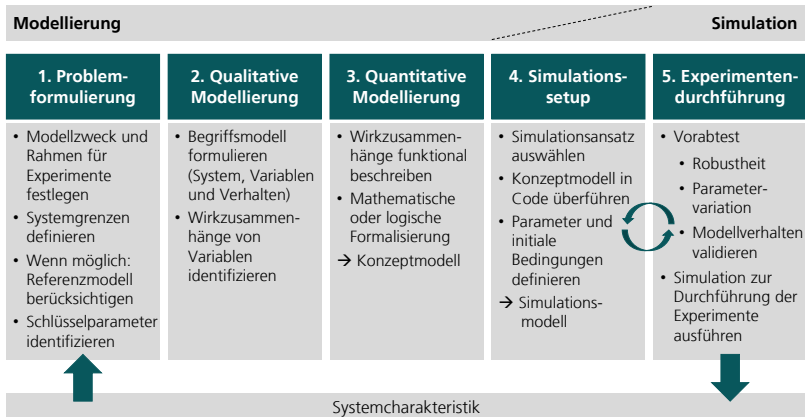


Abbildung 6.3: Schritte zur Modellierung und Simulation (eigene Darstellung in Anlehnung an Bauer et al. 2019c, S. 228; Robinson 2004, S. 51–54; Fishman 2001, S. 37–38)

Conwip-Steuerung sorgt für einen gleichmäßigen Bestand im Produktionssystem und gibt Aufträge entsprechend des Produktionsplans mit den vorgesehenen Lieferterminen frei. FOUP entsprechen den Transporteinheiten für Wafer in einer Halbleiterfertigung, wobei vereinfacht ein FOUP einem Los entsprechen soll. Entsprechend wird diese Subklasse für jedes Los instantiiert und mit dem Arbeitsplan des zugehörigen Produkts verknüpft.

Die Klasse der komplexen Werkstattfertigung beinhaltet die Subklasse der Maschinenengruppe, welche in Häufigkeit der Anzahl der Maschinenengruppen instantiiert wird. Die Maschinenengruppe wiederum besteht aus den Subklassen des Dispatchers, des Eingangspuffers, des Ausgangspuffers und der Maschine. Der Dispatcher beinhaltet dabei die Produktionssteuerung mit dem Verfahren zur Auswahl des nächsten zu bearbeitenden Loses, sobald eine Maschine frei wird. Eingangspuffer und Ausgangspuffer repräsentieren das Lager vor bzw. nach dem Bearbeitungsprozess. Die Maschine stellt den eigentlichen Bearbeitungsprozess dar, modelliert über die Rüstzeit und die Bearbeitungszeit. Instantiiert wird die Subklasse der Maschine in Häufigkeit der Anzahl der Parallelmaschinen innerhalb der jeweiligen Maschinenengruppe.

mentierungen oder anderen konkreten Details. Ein Referenzmodell ist deshalb ein Modell, das einen allgemein gültigen Charakter haben soll (Krcmar 2015, S. 39–40; Nakagawa et al. 2011, S. 209).

²Ein Konzeptmodell ist eine nicht softwarebezogene Beschreibung des zu entwickelnden Modells. Es enthält die Ziele, Eingangsgrößen, Ausgangsgrößen, Inhalte, Annahmen und Vereinfachungen des Modells (Robinson 2004, S. 51–52).

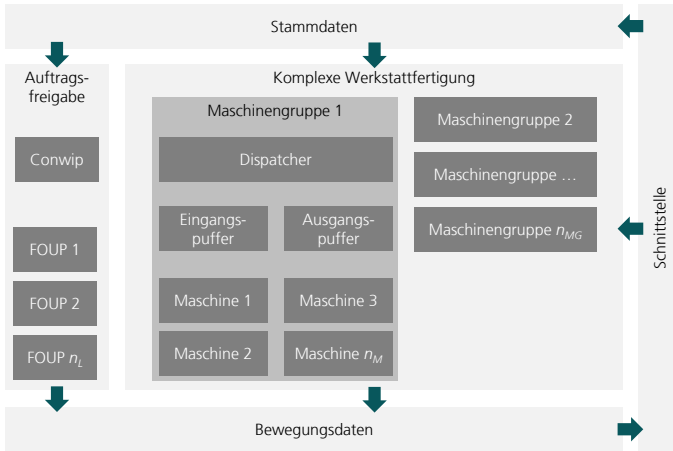


Abbildung 6.4: Struktur des Konzeptmodells für das untersuchte Produktionssystem

Die Stammdaten des Modells umfassen die Struktur der komplexen Werkstattfertigung mit der Anzahl der Maschinengruppen sowie jeweils deren Anzahl an parallelen Maschinen. Des Weiteren müssen die Produkte und dazugehörigen Arbeitspläne definiert werden. In den Bewegungsdaten wird der Zustand von Losen jeweils beim Eintreffen (Eingangspuffer) und Verlassen (Ausgangspuffer) einer Maschinengruppe festgehalten.

Zur Sicherstellung der Übertragbarkeit soll das Konzeptmodell entgegen dem in der Literatur weit verbreiteten SimPy (Müller et al. 2020) in einer Simulationssoftware implementiert werden, welche auch in der Industrie breite Anwendung findet. Implementiert wird das Konzeptmodell deshalb als Simulationsmodell³ in der Software Siemens Tecnomatix Plant Simulation in der Version 14. Als einer der Marktführer bietet Plant Simulation einen objektorientierten Aufbau zur Abbildung von Klassen und Objekten in einer DES, vielfältige vordefinierte Bausteine und die Möglichkeit zur Erweiterung der Bausteine mittels der Programmiersprache SimTalk (Bangsow 2011, S. 9). Durch externe Software kann Plant Simulation mittels einer COM-Schnittstelle gesteuert werden, Daten können unter anderem in einem offenen Format als kommagetrennte Werte exportiert werden (Bangsow 2015, S. 667–683).

³Ein Simulationsmodell ist die Implementierung eines Konzeptmodells in einer Simulationssoftware und damit die softwarebezogene Umsetzung eines Modells (Robinson 2004, S. 51–53).

Für die Validierung in Abschnitt 6.3 werden Ausschnitte aus komplexen Werkstattfertigungen modelliert, welche einem realen Produktionssystem ähnlich sind. Die Ähnlichkeit wurde anhand der Durchlaufdiagramme geprüft (vgl. Abschnitt 2.2.2). Diese Einschränkung muss vorgenommen werden, um bei der in Abschnitt 6.2.2 beschriebenen Implementierung des Entscheidungsmodells mit derzeit verfügbaren Technologien eine akzeptable Trainingsdauer zu erreichen. Entsprechend stimmen die absoluten Werte der Kenngrößen in der Simulation nicht mit der Realität überein, deren Charakteristik jedoch schon – die Ergebnisse der Simulation sind deshalb prinzipiell übertragbar.

6.2.2 Regler des Adaptiven Regelkreises für ein integriertes Wertschöpfungsnetzwerk

Die Programmiersprache Python ist im ML sehr populär, was eine hohe Bandbreite an verfügbaren Bibliotheken mit sich bringt (Raschka et al. 2020, S. 2). Die Implementierung des ARIW-Reglers wird deshalb in Python aufgebaut und besteht aus der Umgebung und dem Agenten (vgl. Abbildung 6.5).

Die Umgebung implementiert in der Klasse *cojosim*⁴ eine Umgebung nach der Spezifikation von OpenAI Gym (Brockman et al. 2016). Hierfür sind Methoden zur Initialisierung der Umgebung (`__init__()`), zur Ausführung eines Episodenschritts (`step()`), zum Zurücksetzen (`reset()`) und zum Schließen der Umgebung (`close()`) notwendig. Der Umgebung kommt dabei die Aufgabe zu, die Kommunikation zwischen dem Agenten und dem Simulationsmodell zu abstrahieren. Hierfür kommt zusätzlich die Klasse *interface* zum Einsatz, welche mittels einer COM-Schnittstelle mit dem auf Plant Simulation basierten Simulationsmodell interagiert (vgl. Abschnitt 6.2.1). Zusätzlich wird ein Ereignis-Generator implementiert, welcher in der Klasse *cojosim* Ereignisse anhand einer Poisson-Verteilung generiert und diese mittels der Klasse *interface* im Simulationsmodell ausführt. Alle Klassen sind dabei modular und parametrisiert aufgebaut, sodass diese in der Design-Phase an verschiedene Anwendungsumgebungen angepasst werden können.

Die Implementierung des Agenten baut auf dem an der University of California in Berkeley entwickelten RL-Framework Ray Rllib auf, welches wiederum zur Umsetzung der

⁴engl. Complex Job Shop Simulation (CoJoSim)

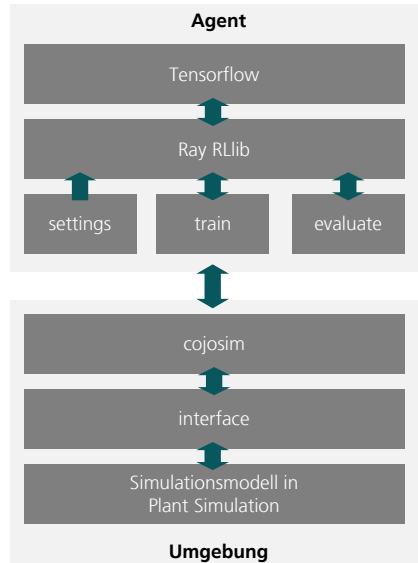


Abbildung 6.5: Implementierung der Umgebung und des Agenten

Deep-Learning-Algorithmen PyTorch oder Tensorflow nutzt (Liang et al. 2018). In den Untersuchungen dieser Arbeit wird das maßgeblich von Google entwickelte Tensorflow genutzt (Abadi et al. 2016). Die Klasse *settings* implementiert alle Einstellungen, welche für das Training und die Evaluation des Agenten notwendig sind. Neben den Modellparametern (vgl. Abschnitt 5.3.2.2) sind dies insbesondere Einstellungen zur Konfiguration der verwendeten Hardware und zur Parametrierung der Umgebung. Das Training des Agenten ist in der Prozedur *train* implementiert. Unter Nutzung von Ray Rllib und dessen OpenAI-Gym-Schnittstelle wird darüber mit der Umgebung interagiert, Episodenschritte ausgeführt, Belohnungen ermittelt und anschließend die dynamischen Modellparameter angepasst. Für die Evaluation des Agenten kommt die Prozedur *evaluate* zum Einsatz, welche den trainierten Agenten auf die Umgebung anwendet und die Modellevaluation durchführt.

Prinzipiell kann die Implementierung des ARIW-Reglers auf handelsüblicher Hardware ausgeführt werden. Das Training des Agenten profitiert jedoch von einer Central Processing Unit (CPU) mit vielen Kernen, da die Umgebung und damit auch das Simulationsmodell dann parallel ausgeführt werden können, um Daten für den Agenten in kürzerer Zeit

zu generieren. Die Umgebungen und der lernende Agent stehen dann in einer $n : 1$ -Beziehung. Der Agent wiederum profitiert von einer Graphics Processing Unit (GPU), welche die Compute Unified Device Architecture (CUDA) API des Herstellers NVIDIA unterstützt. Das rechenaufwändige Lernen von KNN kann damit um bis zu Faktor 20 beschleunigt werden (LeCun et al. 2015, S. 439). CUDA wird von Ray RLLib und Tensorflow unterstützt und damit das Lernen von Algorithmen signifikant beschleunigt (Liang et al. 2018, S. 3; Abadi et al. 2016, S. 275). Im Rahmen dieser Arbeit wird die Implementierung auf einer Workstation mit Intel Xeon Platinum 8268 CPU (48 Kerne, 2,9 GHz) und der GPU NVIDIA Quadro RTX 8000 ausgeführt. Der Hardware-Markt entwickelt sich derzeit in hoher Geschwindigkeit weiter, was zukünftig noch leistungsfähigere Implementierungen erwarten lässt.

6.2.3 Auswahl eines Algorithmus

Zur Auswahl eines Algorithmus für den Agenten des ARIW-Reglers (vgl. Abschnitt 6.2.2) wird anhand der Anforderungen aus Schritt 7 in Abschnitt 5.3.2.2 vorgegangen. Betrachtet werden in der Literatur gut dokumentierte und etablierte Algorithmen, welche die Mindestanforderungen erfüllen. Darunter Algorithmen, die wertbasiert sind und solche, die Eigenschaften wertbasierter und strategiebasierter Algorithmen kombinieren. Die Einordnung hinsichtlich der weiteren Anforderungen wird experimentell durchgeführt. Tabelle 6.1 zeigt eine Übersicht der Algorithmen, Abschnitt A.7 eine detaillierte Darstellung des experimentellen Auswahlprozesses.

Aus den derzeit verfügbaren Algorithmen wird A3C für die Implementierung und Validierung im Rahmen der vorliegenden Arbeit ausgewählt. A3C erlaubt die Abbildung von diskreten und kontinuierlichen Größen im Zustandsraum, einen diskreten Aktionsraum, ist in der Literatur sehr gut dokumentiert und in diversen Open-Source-Frameworks verfügbar. Auch Ray RLLib (vgl. Abschnitt 6.2.2) beinhaltet eine Implementierung von A3C. Im Vergleich zu allen anderen betrachteten Algorithmen erreicht A3C die höchste Verbesserung der Liefertermineinhaltung, konvergiert robust und mit wenigen Schritten sowie in einer akzeptablen Rechenzeit. Im Unterschied zur A3C-Implementierung von Mnih et al. (2016) wird als Schätzer für die Advantage-Funktion der Generalized Advantage Estimator (GAE) $A_t^{GAE(\gamma, \lambda)}$ nach Schulman et al. (2016) genutzt, um die Varianz zu minimieren und die Leistung zu

Tabelle 6.1: Übersicht der betrachteten Algorithmen								
Algorithmus	Anforderungen							
	Zustandsraum: diskret und kontinuierlich	Aktionsraum: diskret	Dokumentation in der Literatur	Open-Source-Code	Verbesserung der Liefertermineinhaltung	Konvergenz	Robustheit	Trainingsgeschwindigkeit
Wertbasierte Algorithmen								
DQN (Mnih et al. 2013)	●	●	●	●	○	○	○	○
DQN Rainbow (Hessel et al. 2018)	●	●	◐	◐	○	◐	○	○
Ape-X (Horgan et al. 2018)	●	●	◐	◐	○	○	○	○
Ape-X Rainbow (Liang et al. 2018)	●	●	◐	◐	○	○	○	○
Wert- und Strategiebasierte Algorithmen								
A3C (Mnih et al. 2016)	●	●	●	●	●	●	●	◐
IMPALA (Espeholt et al. 2018)	●	●	◐	◐	◐	●	◐	●
PPO (Schulman et al. 2017)	●	●	●	●	◐	●	●	◐
APPO (Liang et al. 2018)	●	●	◐	◐	◐	●	◐	●
Legende: ○ nicht erfüllt ◐ teilweise erfüllt ● erfüllt								

verbessern. Die Funktionsweise von A3C mit GAE ist in Code 3 dargestellt. Die Änderung durch GAE besteht dabei einzig im veränderten Schätzer für die Advantage-Funktion in Zeile 17, weshalb der Algorithmus in der vorliegenden Arbeit weiterhin als A3C bezeichnet wird.

Die Forschung im Themenfeld der RL-Algorithmen entwickelt sich sehr schnell weiter, was zukünftig noch leistungsfähigere Algorithmen erwarten lässt. Es ist deshalb sinnvoll, bei zukünftigen Implementierungen erneut einen Vergleich der verfügbaren Algorithmen durchzuführen.

Code 3 A3C mit GAE (in Anlehnung an Mnih et al. 2016, Appendix S. 4; Schulman et al. 2016, S. 5)

```

1: // Assume global shared parameter vectors  $\theta_\pi$  and  $\theta_v$  and global shared counter  $t_g = 0$ 
2: // Assume thread-specific parameter vectors  $\theta_\pi'$  and  $\theta_v'$ 
3: Initialize thread step counter  $t \leftarrow 1$ 
4: while  $t_g \leq t_{g,max}$  do
5:   Reset gradients:  $d\theta_\pi \leftarrow 0$  and  $d\theta_v \leftarrow 0$ 
6:   Synchronize thread-specific parameter vectors  $\theta_\pi' = \theta_\pi$  and  $\theta_v' = \theta_v$ 
7:    $t_{start} = t$ 
8:   Get state  $S_t$ 
9:   while  $S_t \neq \text{terminal } S_t$  and  $t - t_{start} \neq t_{max}$  do
10:     Perform action  $A_t$  according to policy  $\pi(A_t|S_t, \theta_\pi')$ 
11:     Receive reward  $R_t$  and new state  $S_{t+1}$ 
12:     Update counters  $t \leftarrow t + 1$  and  $t_g \leftarrow t_g + 1$ 
13:   end while
14:    $R = \begin{cases} 0 & \text{for terminal } S_t \\ V(S_t, \theta_v') & \text{for non-terminal } S_t // \text{bootstrap from last state} \end{cases}$ 
15:   for  $i \in t - 1, \dots, t_{start}$  do
16:     Update cumulative reward by  $R \leftarrow R_i + \gamma R$ 
17:     Accumulate gradients wrt  $\theta_\pi'$ :  $d\theta_\pi \leftarrow d\theta_\pi + \nabla_{\theta_\pi} \log \pi(A_i|S_i, \theta_\pi') (A_i^{GAE(\gamma, \lambda)} + \beta \nabla_{\theta_\pi} H(\pi(S_i, \theta_\pi')))$ 
18:     Accumulate gradients wrt  $\theta_v'$ :  $d\theta_v \leftarrow d\theta_v + \partial(R - V(S_i, \theta_v'))^2 / \partial \theta_v'$ 
19:   end for
20:   Perform asynchronous update of  $\theta_\pi$  using  $d\theta_\pi$  and of  $\theta_v$  using  $d\theta_\pi$ 
21: end while

```

6.3 Anwendung und Validierung in der Halbleiterfertigung

Die Anwendung und Validierung der Implementierung in der Halbleiterfertigung wird anhand von drei Schritten dargestellt. Im ersten Schritt werden die Anwendungsumgebung und das Design des implementierten ARIW-Reglers für eben jene beschrieben (vgl. Abschnitt 6.3.1). Anschließend folgt das Training des Entscheidungsmodells (vgl. Abschnitt 6.3.2). Im dritten Schritt wird mit der Applikation des Reglers die Eignung zur Verbesserung der Liefertermineinhaltung bei Ereignissen aus dem Wertschöpfungsnetzwerk analysiert (vgl. Abschnitt 6.3.3).

6.3.1 Design des Reglers für die Anwendungsumgebung

Für die Anwendungsumgebung wird das implementierte Simulationsmodell einer komplexen Werkstattfertigung in der Halbleiterindustrie (vgl. Abschnitt 6.2.1) genutzt, welches die

Charakteristik der Halbleiterfertigung (vgl. Abschnitt 6.1.2) abbildet. Das Simulationsmodell wird dabei mit den Stammdaten des in der Literatur weit verbreiteten und seit Jahren etablierten MIMAC-1 (Fowler & Robinson 1995) parametrisiert. Aufgrund der Leistungsfähigkeit der derzeit verfügbaren Technologien für den ARIW-Regler werden nur relevante Ausschnitte aus MIMAC-1 modelliert, um eine akzeptable Trainingsdauer zu erreichen (vgl. Abschnitt 6.2.2). Betrachtet werden hierfür neun Maschinengruppen mit insgesamt 28 Maschinen, welche sich in den Häufigkeiten eins bis zehn auf die Maschinengruppen verteilen, zwei Produkte mit unterschiedlichen Arbeitsplänen sowie 45 Lose im Bestand des untersuchten Produktionssystems. Die Arbeitspläne beider Produkte enthalten wiedereintretende Prozessflüsse, sodass Lose beider Produkte und unterschiedlicher Fertigungsgrade um dieselben Maschinen konkurrieren. Der Bestand des Produktionssystems wird über eine Conwip-Steuerung reguliert, welche Aufträge mit unterschiedlichen Lieferterminen für das Produktionssystem freigibt. Für das Dispatching an den Maschinengruppen kommt eine Kombination aus der Differenz zwischen Plan-Flussfaktor und Ist-Flussfaktor sowie der kürzesten Restbearbeitungszeit zum Einsatz. Die hiermit bei einer Betrachtung ohne Ereignisse aus dem Wertschöpfungsnetzwerk erzielbare Liefertermineinhaltung liegt bei 84 Prozent (vgl. Abbildung 6.6). Die originale MIMAC-1 -Implementierung erreicht mit FIFO eine Liefertermineinhaltung von 72 Prozent, mit demselben kombinierten Dispatching-Verfahren 87 Prozent. Aufgrund der vergleichbaren Charakteristik sind die Ergebnisse deshalb prinzipiell übertragbar.

Für das Design des ARIW-Reglers für die Anwendungsumgebung wird entsprechend Schritt 1 bis Schritt 6 in Abschnitt 5.3.2.1 vorgegangen. Ereignisse in Wertschöpfungsnetzwerken resultieren aus Marktunsicherheiten, Abweichungen zwischen Prognosen und tatsächlichem Verhalten sowie Nachfrageverstärkungen und parallelen Interaktionen. Daraus folgende Prioritätsänderungen von Kundenaufträgen treten häufig mindestens ein Mal pro Schicht auf und haben einen signifikanten Einfluss auf die Leistung des Produktionssystems (vgl. Abschnitt 2.1.3), weshalb diese im Ereignisraum abgebildet werden. Die Ereignisse des Wertschöpfungsnetzwerks der Anwendungsumgebung werden deshalb mittels einer Poisson-Verteilung bei einer Schichtdauer von acht Stunden mit dem Erwartungswert $\mu_P = 1$ und dem Zeitintervall $t_e = 28800s$ ausgelöst. Der Zustandsraum umfasst die in Schritt 2 in Abschnitt 5.3.2.1 beschriebenen Kenngrößen für die 45 Lose im Bestand

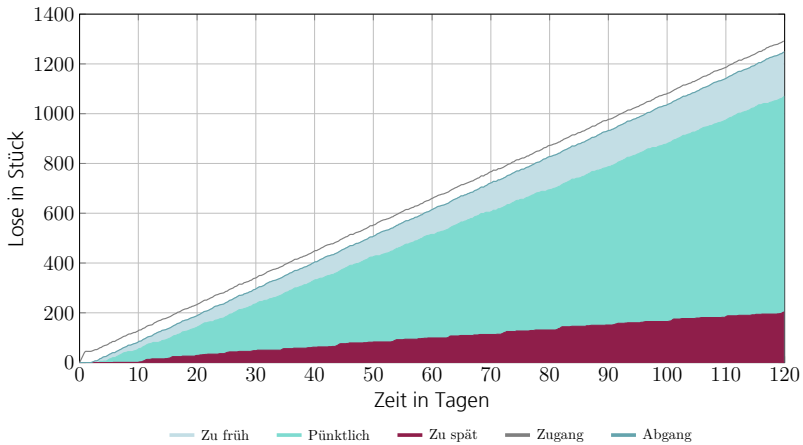


Abbildung 6.6: Durchlaufdiagramm des Produktionssystems der Anwendungsumgebung

der Anwendungsumgebung. Daraus resultiert auch der Aktionsraum mit insgesamt 91 Aktionen, welche dem Entscheidungsmodell des ARIW-Reglers zur Verfügung stehen (vgl. Schritt 3 in Abschnitt 5.3.2.1). Ebenso wird damit auch festgelegt, dass für die Gestaltung der Belohnungsfunktion die 45 Lose im Bestand betrachtet werden (vgl. Schritt 6 in Abschnitt 5.3.2.1). Die Vorbereitung des Simulationsmodells und die Integration der Daten wurde bereits in Abschnitt 6.2 dargestellt.

Mit den oben beschriebenen Ereignissen wird der Plan-Flussfaktor FF_{Plan} dabei um $\Delta FF_{Plan} = \pm 0,5$ aus dem Ereignis E_t angepasst. In der Konsequenz sinkt die Liefertermineinhaltung des Produktionssystems unter dem Einfluss dieser Ereignisse auf 80 Prozent. In Abschnitt 6.3.2 werden der Erwartungswert, das Zeitintervall und die Flussfaktoränderung der Ereignisse konstant gehalten, in Abschnitt 6.3.3.2 findet eine Variation der Ereignisse statt.

6.3.2 Training des Regler-Entscheidungsmodells

Für das Training des Entscheidungsmodells des ARIW-Reglers wird entsprechend Schritt 7 bis Schritt 11 in Abschnitt 5.3.2.2 vorgegangen. Die einzelnen Trainingsläufe werden dabei mit dem Algorithmus A3C bis $T_{Epoche} = 1 \cdot 10^7$ Schritte und, sofern nicht anders angegeben,

mit Episoden bis $T = 300$ Schritte, trainiert. Die Aktoranzahl $n_{Akt} = 4$ mit jeweils 10 Anwendungsumgebungen wird so gewählt, dass die in Abschnitt 6.2.2 dargestellte Hardware ausgelastet ist. Angelehnt an Pfund et al. (2006, S. 221) und Schritt 5 in Abschnitt 5.3.2.1 werden die Ereignisse während des Trainings anhand einer Poisson-Verteilung mit dem Erwartungswert $\mu_P = 1$ und dem Zeitintervall $t_e = 28800s$ sowie $\Delta FF_{Plan} = 0,5$ generiert⁵. Die Rechenzeit einer Epoche beträgt damit etwa 20 Stunden.

Anhand der Literatur wird ein Standardparametersatz gebildet, mit welchem der Agent in der sogenannten Basislinie⁶ trainiert wird. Gegenüber dieser Basislinie werden die einzelnen Hyperparameter und Agentenparameter, ebenfalls angelehnt an die Literatur, jeweils in einem Trainingslauf variiert. Dies ist notwendig, da aufgrund kontinuierlich anpassbarer Parameter grundsätzlich beliebig viele Variationen denkbar wären. Zuerst findet dann eine Optimierung der Hyperparameter statt (vgl. Abschnitt 6.3.2.1), anschließend werden die Agentenparameter variiert (vgl. Abschnitt 6.3.2.2). Vor der nachfolgenden Anwendung des ARIW-Reglers werden die Ergebnisse des Trainings bewertet (vgl. Abschnitt 6.3.2.3).

6.3.2.1 Hyperparameter-Optimierung

Hyperparameter-Optimierung beschreibt die Anpassung der Fähigkeiten eines Entscheidungsmodells an die Komplexität einer Problemstellung (Goodfellow et al. 2016, S. 428). Ausgehend von der Auswahl eines Algorithmus (vgl. Abschnitt 6.2.3) ermöglicht die Hyperparameter-Optimierung ein Feintuning abhängig von der Anwendungsumgebung (vgl. Abschnitt 6.3.1). Die zu optimierenden Hyperparameter sind dabei vom Algorithmus abhängig (vgl. Schritt 7 in Abschnitt 5.3.2.2). Für A3C sind dies die Hyperparameter Lernrate α , Entropiefaktor β , Diskontierungsfaktor γ und Struktur des KNN D_{KNN} .

Die Basislinie und die Variation der Hyperparameter werden anhand der Literatur gebildet (Naik et al. 2019, S. 6; Mnih et al. 2016, S. 1931–1932). Eine Übersicht der Hyperparameter-Optimierung ist Tabelle 6.2 zu entnehmen, eine detaillierte Darstellung des Einflusses aller Variationen dem Abschnitt A.8. Nachfolgend sind die Ergebnisse der Hyperparameter-

⁵Im Folgenden wird für Ereignisse eine Notation mit dem Erwartungswert, dem Zeitintervall und der Anpassung des Plan-Flussfaktors genutzt: $\mu_P - t_e - \Delta FF_{Plan}$. Für die genannten Werte entspricht dies der Notation 1–28800–0,5.

⁶engl. baseline

Optimierung für die Lernrate, den Entropiefaktor, den Diskontierungsfaktor und die Struktur des KNN kurz zusammengefasst.

Tabelle 6.2: Untersuchte Variation der Hyperparameter

Parameter	Typ	Basislinie	Variation
Lernrate α	dynamisch	$1 \cdot 10^{-4}$	$1 \cdot 10^{-2}$; $1 \cdot 10^{-3}$; $5 \cdot 10^{-4}$; $1 \cdot 10^{-5}$; $[5 \cdot 10^{-4}, 1 \cdot 10^{-5}]$
Entropiefaktor β	statisch	0,01	0,02; 0
Diskontierungsfaktor γ	statisch	0,99	0,9999; 0,9
Struktur des KNN D_{KNN}	statisch	256–256	512–512; 512–128; 256–128; 128–128; 256–256–256; 256; 256–256–256–LSTM; 256–256–LSTM

Lernrate Höhere Werte für die Lernrate α erlauben zwar eine Beschleunigung des Trainings durch Konvergenz in weniger Schritten, bergen jedoch die Gefahr eines instabilen Agenten. Niedrige Werte für die Lernrate führen hingegen zu robusten Agenten mit weniger volatilen Belohnungsverläufen, erfordern jedoch längere Epochen und eine signifikant höhere Rechenzeit zur Erreichung desselben Maximalwerts der Belohnung⁷ (vgl. Abschnitt A.8). Durch die dynamische Kombination mit $\alpha = [5 \cdot 10^{-4}, 1 \cdot 10^{-5}]$ wird derselbe Maximalwert der Belohnung wie in der Basislinie erreicht, jedoch bei schnellerer Konvergenz und einem stabileren Verhalten im konvergierten Zustand (vgl. Abbildung 6.7).

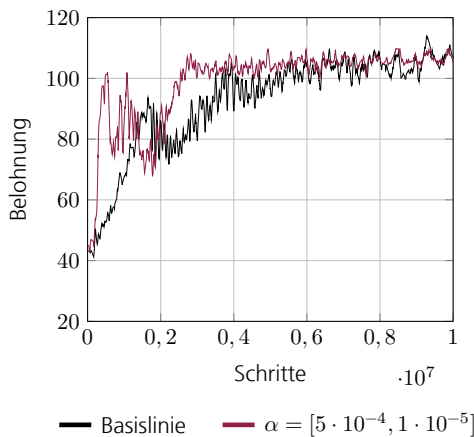


Abbildung 6.7: Einfluss der Variation der Lernrate

Entropiefaktor Während bei der Variation des Entropiefaktors β durch eine stärkere Berücksichtigung der Entropie keine beschleunigte Konvergenz zu erreichen ist, verlangsamt sich diese jedoch auch mit $\beta = 0$ nicht (vgl. Abschnitt A.8). Gleichzeitig wird dadurch ein höherer Maximalwert der Belohnung gegenüber der Basislinie erzielt (vgl. Abbildung 6.8).

⁷Mit Maximalwert der Belohnung ist hier nicht eine einzelne Spitze bezeichnet, sondern der durchschnittliche Verlauf der Belohnungskurve im konvergierten Zustand. Folglich entspricht der als Maximalwert der Belohnung bezeichnete Wert bei sehr stabilem Verhalten und deren Belohnungskurven, etwa dem Kurvenverlauf, bei instabilem Verhalten dem durchschnittlichen Kurvenverlauf in Form des Mittelwerts positiver und negativer Abweichungen.

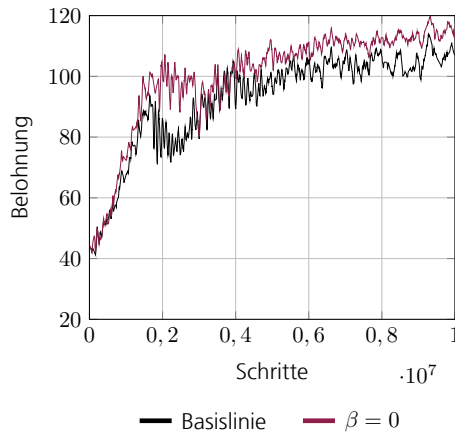


Abbildung 6.8: Einfluss der Variation des Entropiefaktors

Diskontierungsfaktor Bei der Variation des Diskontierungsfaktors γ kann für die vorliegende Anwendungsumgebung keine Verbesserung der maximalen Belohnung oder der Stabilität des Verhaltens erreicht werden. Mit $\gamma = 0,99$ erreicht die Basislinie hinsichtlich des Diskontierungsfaktors das beste Ergebnis (vgl. Abschnitt A.8).

Struktur des KNN Ebenso kann auch bei der Variation der Struktur des KNN D_{KNN} für die vorliegende Anwendungsumgebung keine Verbesserung der maximalen Belohnung oder der Stabilität des Verhaltens erzielt werden. Das beste Ergebnis wird in der Basislinie mit $D_{KNN} = 256-256^8$ erreicht (vgl. Abschnitt A.8).

6.3.2.2 Parametervariation der Agentenparameter

Im Rahmen der Variation der Agentenparameter werden jene Parameter variiert, welche keine Hyperparameter des Algorithmus sind, jedoch direkt das Verhalten des Agenten beeinflussen. Die Basislinie bleibt dabei gegenüber der Hyperparameter-Optimierung (vgl. Abschnitt 6.3.2.1) unverändert. Die Modellgewichte θ , für A3C konkret θ_π und θ_v , werden während des Lernens durch den Algorithmus angepasst, weshalb dies kein Parameter

⁸Die Notation der Struktur des KNN D_{KNN} repräsentiert die Anzahl der Neuronen in den versteckten Schichten des KNN. $256-256$ bedeutet damit, dass das KNN zwei gleich große versteckte Schichten mit jeweils 256 Neuronen beinhaltet.

für die Variation der Agentenparameter ist (vgl. Abschnitt 6.2.3). Variiert werden die Flussfaktor-Anpassung ΔFF_{Steuer} und die Schrittweite Δt . In der Basislinie entspricht die Flussfaktor-Anpassung $\Delta FF_{Steuer} = 0,25$ einer zehnprozentigen Beschleunigung oder Verlangsamung eines Loses durch den Agenten gegenüber dem durchschnittlichen Flussfaktor. Die Wahl der Schrittweite $\Delta t = 3600s$ entspricht in der Basislinie der niedrigsten Durchführungszeit eines Arbeitsvorgangs ZDF des Arbeitsplans, also der niedrigsten Summe aus Bearbeitungszeit und Rüstzeit je Arbeitsvorgang. Eine Übersicht der Parametervariation der Agentenparameter ist Tabelle 6.3 zu entnehmen, eine detaillierte Darstellung des Einflusses aller Variationen dem Abschnitt A.8. Nachfolgend sind die Ergebnisse für die Flussfaktor-Anpassung und die Schrittweite kurz zusammengefasst.

Tabelle 6.3: Untersuchte Variation der Agentenparameter

Parameter	Typ	Basislinie	Variation
Flussfaktor-Anpassung ΔFF_{Steuer}	statisch	0,25	0,125; 0,5; 0,75; 1,0
Schrittweite Δt	statisch	3600s	1800s; 7200s

Flussfaktor-Anpassung Bei der Variation der Flussfaktor-Anpassung ΔFF_{Steuer} wird deutlich, dass es bei kleineren Werten für die Flussfaktor-Anpassungen nicht möglich ist, Lose in einem ausreichenden Maß zu beeinflussen. Mit $\Delta FF_{Steuer} = 0,5$ wird gegenüber der Basislinie ein höherer Maximalwert der Belohnung erzielt (vgl. Abbildung 6.9). Bei weiter zunehmendem ΔFF_{Steuer} kann jedoch keine weitere Verbesserung erreicht werden (vgl. Abschnitt A.9).

Schrittweite Bei der Variation der Schrittweite Δt kann für die vorliegende Anwendungsumgebung keine Verbesserung der maximalen Belohnung oder der Stabilität des Verhaltens erreicht werden. Mit $\Delta t = 3600s$ erreicht die Basislinie hinsichtlich der Schrittweite das beste Ergebnis (vgl. Abschnitt A.9).

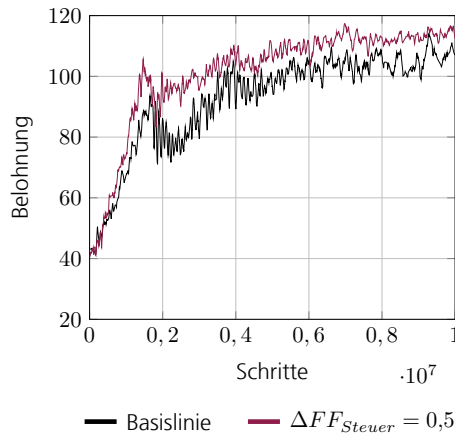


Abbildung 6.9: Einfluss der Variation der Flussfaktor-Anpassung

6.3.2.3 Trainingsbewertung

Im Rahmen des Trainings wurden sechs Parameter variiert, darunter vier Hyperparameter (vgl. Abschnitt 6.3.2.1) und zwei Agentenparameter (vgl. Abschnitt 6.3.2.2). Dabei zeigt sich über alle variierten Parameter hinweg, dass Konvergenz auch bei den Variationen gegeben ist. Die einzige Ausnahme weist die Lernrate auf – bei zu großen Werten konvergiert die Belohnungskurve nicht mehr und das Verhalten wird instabil. Die Unterschiede zwischen den variierten Parametern werden insbesondere an der Dauer bis zur Konvergenz, dem erreichten Maximalwert der Belohnung und der Stabilität der Belohnungskurve deutlich. Es wird jedoch auch deutlich, dass mit der Basislinie bereits ein sehr gutes Verhalten hinsichtlich der Belohnungskurve erzielt werden kann und sowohl die Hyperparameter-Optimierung, als auch die Variation der Agentenparameter, eine Feinabstimmung darstellen.

Eine Verbesserung gegenüber der Basislinie kann für die vorliegende Anwendungsumgebung der Halbleiterfertigung durch eine Anpassung der Lernrate, des Entropiefaktors und der Flussfaktor-Anpassung erreicht werden. Tabelle 6.4 zeigt eine Gegenüberstellung der Parameter der Basislinie und der nach Abschluss des Trainings angepassten Parameter $\alpha = [5 \cdot 10^{-4}, 1 \cdot 10^{-5}]$, $\beta = 0$ und $\Delta FF_{Steuer} = 0,5$, welche fortan als Trainingslinie bezeichnet werden sollen. Zwar kann für die vorliegende Anwendungsumgebung durch die Variation der Struktur des KNN keine Verbesserung erzielt werden, bei deutlich größeren

Anwendungsumgebungen ist jedoch zu erwarten, dass dieser Parameter einen signifikanten Stellhebel darstellt, wenngleich dann möglicherweise andere Algorithmen besser geeignet sein könnten (Ota et al. 2021, S. 4–8). Es gilt auch generell für alle Modellparameter, dass diese von der Anwendungsumgebung abhängig sind und entsprechend bei anderen Anwendungsumgebungen andere Werte zielführender sein können (Goodfellow et al. 2016, S. 428). Es ist deshalb zu empfehlen, die Modellparameter bei anderen Anwendungsumgebungen anhand des dargestellten Vorgehens erneut zu variieren. Die gegenüber der Basislinie optimierte Trainingslinie der vorliegenden Arbeit stellt hierfür die Grundlage dar.

Tabelle 6.4: Vergleich der Hyperparameter und Agentenparameter der Basislinie und der Trainingslinie

Parameter	Basislinie	Trainingslinie
Lernrate α	$1 \cdot 10^{-4}$	$[5 \cdot 10^{-4}, 1 \cdot 10^{-5}]$
Entropiefaktor β	0,01	0
Diskontierungsfaktor γ	0,99	0,99
Struktur des KNN D_{KNN}	256–256	256–256
Flussfaktor-Anpassung ΔFF_{Steuer}	0,25	0,5
Schrittweite Δt	3600s	3600s

Trainiert unter den selben Bedingungen kann mit der Trainingslinie gegenüber der Basislinie für die vorliegende Anwendungsumgebung der Halbleiterfertigung ein etwa fünf Prozent höherer Maximalwert der Belohnungskurve erreicht werden. Gleichzeitig ist das Verhalten im konvergierten Zustand stabiler und dieser wird schneller erreicht (vgl. Abbildung 6.10).

6.3.3 Applikation des Reglers

Zur Applikation des ARIW-Reglers wird anhand von Schritt 12 bis Schritt 16 in Abschnitt 5.3.2.3 vorgegangen. Im Rahmen dieser Arbeit wird jedoch nicht mit einem realen Produktionssystem interagiert, sondern weiterhin mit dem Simulationsmodell einer komplexen Werkstattfertigung der Halbleiterfertigung. Zur Abbildung der immanenten Unsicherheit beim Übergang vom Training in den Betrieb, werden die Prozesszeiten der Arbeitspläne zufallsverteilt im Intervall ± 10 Prozent angepasst. Die Umgebung des Agenten entspricht damit in

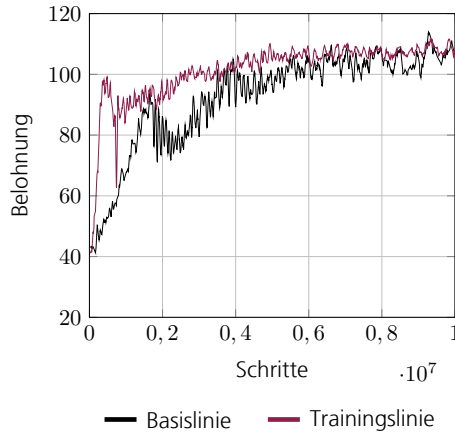


Abbildung 6.10: Vergleich der Basislinie und der Trainingslinie

der Applikation nicht mehr exakt der des Trainings, sodass die Fähigkeit zur Generalisierung des Agenten innerhalb des ARIW-Reglers und damit der lernenden ereignisbasierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung geprüft werden kann.

Während im Training Episoden mit $T = 300$ Schritten, bei einer Schrittweite $\Delta t = 3600s$ entsprechend 12,5 Tage, trainiert wurden (vgl. Abschnitt 6.3.2), wird in der Applikation anhand von Episoden mit $T = 2880$ Schritten, entspricht 120 Tagen, validiert. Dies entspricht typischen Durchlaufzeiten in der Halbleiterindustrie (vgl. Abschnitt 6.1.2). Gestartet wird dabei mit einem eingeschwungenen Simulationsmodell zum Zeitpunkt $t_0 = 30$ Tage. Die Rechenzeit in der Validierung beträgt etwa 2 Minuten, womit die Rechenzeit pro Schritt gegenüber dem Training um knapp 90 Prozent geringer ist.

Für die Applikation des ARIW-Reglers wird im ersten Schritt auf Basis des trainierten Entscheidungsmodells die Liefertermineinhaltung anhand der Anwendungsumgebung der Halbleiterfertigung validiert (vgl. Abschnitt 6.3.3.1). Im zweiten Schritt folgt eine Grenzbeurteilung hinsichtlich der Ereignisse aus dem Wertschöpfungsnetzwerk, welche zuverlässig verarbeitet werden können (vgl. Abschnitt 6.3.3.2). Abschließend werden die Ergebnisse der Applikation bewertet (vgl. Abschnitt 6.3.3.3).

6.3.3.1 Liefertermineinhaltung

Für die Validierung des ARIW-Reglers wird dessen Auswirkung auf die Liefertermineinhaltung bei Ereignissen aus dem Wertschöpfungsnetzwerk analysiert. Ebenso wie im Training werden hierfür Ereignisse anhand einer Poisson-Verteilung mit dem Erwartungswert $\mu_p = 1$ und dem Zeitintervall $t_e = 28800s$ sowie $\Delta FF_{Plan} = 0,5$ generiert, die Anwendungsumgebung weicht jedoch von der Anwendungsumgebung des Trainings ab (vgl. Abschnitt 6.3.3).

Ohne den ARIW-Regler beträgt die Liefertermineinhaltung im Szenario ohne Ereignisse 84 Prozent. Im Szenario mit Ereignissen fällt die Liefertermineinhaltung anfänglich und stabilisiert sich anschließend bei 80 Prozent, womit die Differenz zwischen beiden Szenarien 4 Prozentpunkte beträgt. Bei Anwendung des ARIW-Reglers mit den Modellparametern der Trainingslinie kann die Liefertermineinhaltung trotz Ereignissen aus dem Wertschöpfungsnetzwerk um 3 Prozentpunkte auf dann 83 Prozent gesteigert werden (vgl. Abbildung 6.11). Die Konsequenzen der Ereignisse aus dem Wertschöpfungsnetzwerk auf die Liefertermineinhaltung werden für die vorliegende Anwendungsumgebung der Halbleiterfertigung entsprechend durch den ARIW-Regler um 75 Prozent reduziert. Dabei sind die Episoden der Validierung um nahezu den Faktor Zehn länger als die Episoden des Trainings, was auf eine gute Generalisierungsfähigkeit des ARIW-Reglers schließen lässt.

Erreicht wird die Verbesserung der Liefertermineinhaltung durch die Aktionen des RL-basierten ARIW-Reglers zur Prioritätsbeeinflussung von Losen. Die Analyse der ausgewählten Aktionen zeigt, dass der ARIW-Regler dabei nicht nur Aktionen auswählt, um durch Ereignisse betroffene Lose zu beeinflussen. Auch andere Lose im Umlaufbestand werden in ihrer Priorität verändert und damit beschleunigt oder verlangsamt, um Lose an einzelnen Maschinengruppen neu zu sortieren. Die Verteilung der nächsten zu bearbeitenden Lose innerhalb der Maschinengruppen auf die einzelnen Maschinen verbleibt dabei beim Dispatcher der Anwendungsumgebung (vgl. Abschnitt 6.3.1).

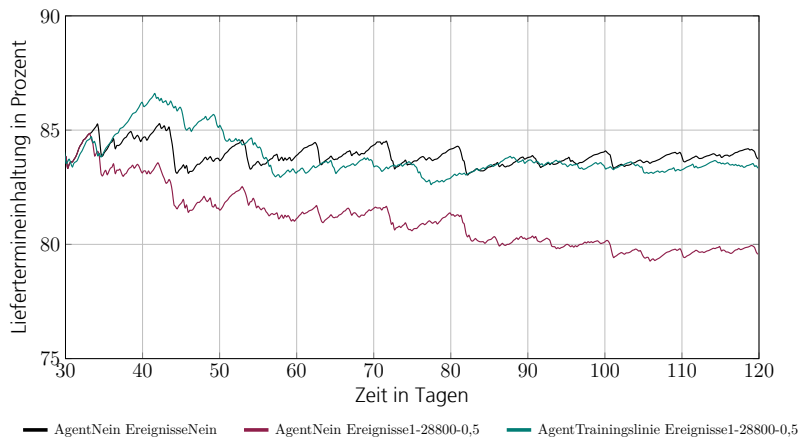


Abbildung 6.11: Liefertermineinhaltung mit Ereignissen 1-28800-0,5

6.3.3.2 Grenzbetrachtung

Im Rahmen der Grenzbetrachtung wird untersucht, welche Ereignisse aus dem Wertschöpfungsnetzwerk durch den ARIW-Regler zuverlässig verarbeitet werden können und wo dessen Grenzen sind. Zuverlässig bedeutet an dieser Stelle, dass eine Verbesserung der Liefertermineinhaltung bei Anwendung des ARIW-Reglers gegenüber dem Szenario ohne ARIW-Regler realisiert werden kann. Für die Grenzbetrachtung werden aus der Poisson-Verteilung der Ereignisse das Zeitintervall t_e und die Lieferterminänderung ΔFF_{Plan} variiert. Die Variation des Zeitintervalls ist dabei angelehnt an Pfund et al. (2006, S. 221). Eine Übersicht der Variationen für die Grenzbetrachtung ist Tabelle 6.5 zu entnehmen, eine detaillierte Darstellung des Einflusses aller Variationen dem Abschnitt A.10.

Tabelle 6.5: Untersuchte Variation der Ereignisse zur Grenzbetrachtung

Parameter	Trainingslinie	Variation
Zeitintervall t_e	28800s	86400s; 14400s; 3600s; 900s
Lieferterminänderung ΔFF_{Plan}	0,5	0,25; 0,75; 1,0; 1,25

Zeitintervall Gegenüber dem Szenario ohne Ereignisse fällt die Liefertermineinhaltung mit sinkendem Zeitintervall t_e deutlich ab (vgl. Abbildung 6.12). Der ARIW-Regler

erreicht dabei dennoch bei allen untersuchten Zeitintervallen eine Verbesserung der Liefertermineinhaltung und zeigt damit auch die Fähigkeit zur Generalisierung auf andere, nicht im Training enthaltene, Ereignisse. Prozentual nimmt die Verbesserung der Liefertermineinhaltung gegenüber dem Szenario ohne ARIW-Regler dabei sogar von 0,6 Prozent bei 1–86400–0,5 auf 6,3 Prozent bei 1–900–0,5 zu⁹. Die Konsequenzen der Ereignisse aus dem Wertschöpfungsnetzwerk auf die Liefertermineinhaltung werden durch den ARIW-Regler jedoch bei absinkendem Zeitintervall auch weniger stark reduziert. Im trainierten Fall 1–28800–0,5 werden die Konsequenzen um 75 Prozent reduziert (vgl. Abschnitt 6.3.3.1), für die Ereignisse 1–900–0,5 beträgt die Reduktion noch 19 Prozent (vgl. Abschnitt A.10).

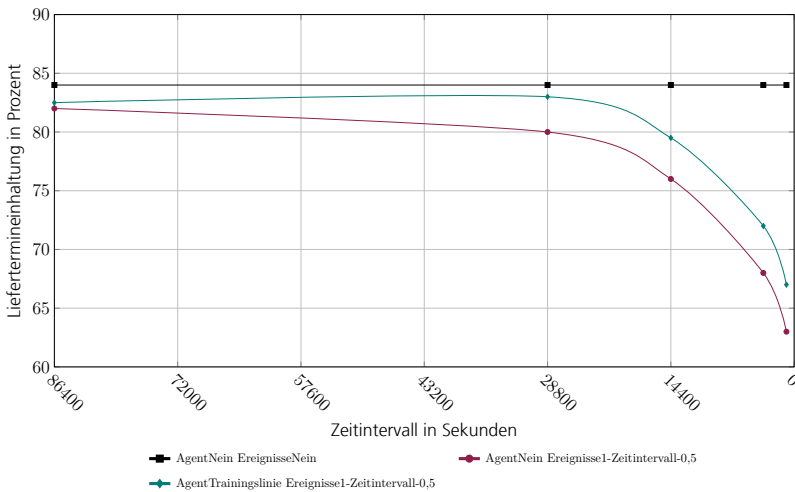


Abbildung 6.12: Einfluss des Zeitintervalls von Ereignissen auf die Liefertermineinhaltung

⁹Hiermit ist nicht die Zunahme um absolute Prozentpunkte gegenüber den beiden Szenarien gemeint, sondern die prozentuale Verbesserung der Liefertermineinhaltung. So fällt beispielsweise bei den Ereignissen 1–900–0,5 die Liefertermineinhaltung um 21 Prozentpunkte auf 63 Prozent, durch den ARIW-Regler kann sie auf 67 Prozent verbessert werden. Die prozentuale Verbesserung beträgt damit 6,3 Prozent.

Lieferterminänderung Die Zunahme der Lieferterminänderung ΔFF_{Plan} führt zu einer deutlich weniger stark abfallenden Liefertermineinhaltung als dies beim Zeitintervall der Fall ist (vgl. Abbildung 6.13). Zwar sinkt die Liefertermineinhaltung mit zunehmender Lieferterminänderung ebenfalls ab, beginnend ab den Ereignissen 1–28800–0,75 stabilisiert sie sich jedoch bei etwa 79 Prozent. Während der ARIW-Regler für 1–28800–1,0 noch eine Verbesserung der Liefertermineinhaltung erreichen kann, fällt die Liefertermineinhaltung für 1–28800–1,25 unterhalb jene des Szenarios ohne den ARIW-Regler. Der ARIW-Regler wird also instabil und ist nicht mehr in der Lage, durch die ausgewählten Aktionen eine Verbesserung der Liefertermineinhaltung zu bewirken. Bereits vorher sinkt jedoch auch die prozentuale Reduktion der Konsequenzen aus dem Wertschöpfungsnetzwerk durch den ARIW-Regler von 80 Prozent bei 1–28800–0,25 auf 20 Prozent bei 1–28800–1,0. Durch die Gleichsetzung des Agentenparameters der Flussfaktor-Anpassung ΔFF_{Steuer} mit der Lieferterminänderung ΔFF_{Plan} kann dabei auch keine Verbesserung erzielt werden (vgl. Abschnitt A.10).

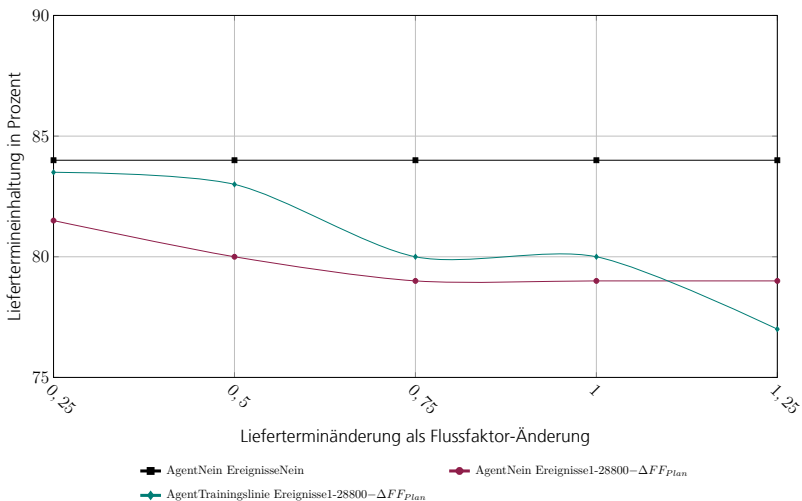


Abbildung 6.13: Einfluss der Lieferterminänderung von Ereignissen auf die Liefertermineinhaltung

Zeitintervall und Lieferterminänderung Wird sowohl das Zeitintervall t_e abgesenkt als auch gleichzeitig die Lieferterminänderung ΔF_{Plan} erhöht, sinkt die Liefertermineinhaltung mit den Ereignissen 1–14400–0,75 im Szenario ohne den ARIW-Regler um 9 Prozentpunkte auf 75 Prozent (vgl. Abbildung 6.14). Verglichen mit 1–14400–0,5 sinkt die Liefertermineinhaltung also nur um einen zusätzlichen Prozentpunkt. Mit den Ereignissen 1–14400–0,75 erreicht der ARIW-Regler dabei jedoch nur eine Verbesserung der Liefertermineinhaltung um zwei Prozentpunkte auf 77 Prozent, wohingegen für 1–14400–0,5 eine Verbesserung um 3,5 Prozentpunkte erreicht werden kann. Nichtsdestotrotz zeigt die Analyse eindeutig die Fähigkeit zur Generalisierung des ARIW-Regler hinsichtlich aus dem Training unbekannter Ereignisse. Das ist unabhängig davon, ob sich diese Ereignisse im Zeitintervall ihres Auftretens oder der Lieferterminänderung unterscheiden. Wird der ARIW-Regler mit den Modellparametern der Trainingslinie auf die Ereignisse 1–14400–0,75 trainiert, ist eine zusätzliche Verbesserung der Liefertermineinhaltung um weitere 2,5 Prozentpunkte auf dann 79,5 Prozent erreichbar. Die Belohnungskurve des Trainings ist Abschnitt A.10 zu entnehmen.

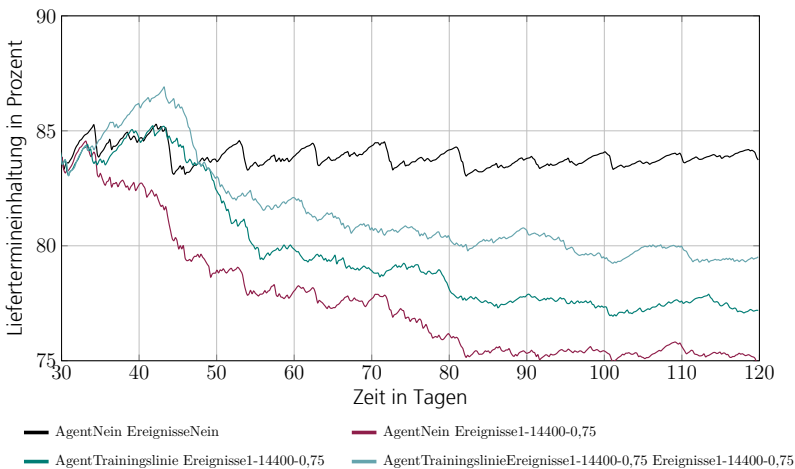


Abbildung 6.14: Liefertermineinhaltung mit Ereignissen 1-14400-0,75

6.3.3.3 Ergebnisbewertung

Im Rahmen der Applikation wurden das Potenzial des ARIW-Reglers zur Verbesserung der Liefertermineinhaltung bei Ereignissen aus dem Wertschöpfungsnetzwerk analysiert (vgl. Abschnitt 6.3.3.1) und eine Grenzbetrachtung hinsichtlich der Ereignisse aus dem Wertschöpfungsnetzwerk durchgeführt (vgl. Abschnitt 6.3.3.2). Bezüglich der Liefertermineinhaltung zeigt sich dabei, dass die Konsequenzen von Ereignissen aus dem Wertschöpfungsnetzwerk für die vorliegende Anwendungsumgebung der Halbleiterfertigung durch Aktionen des ARIW-Reglers in der Produktionssteuerung der komplexen Werkstattfertigung um bis zu 75 Prozent reduziert werden können und die Liefertermineinhaltung nachhaltig stabilisiert werden kann. Das Konzept der lernenden ereignisbasierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung hat sich in der Applikation dahingehend als zielführend erwiesen, dass durch den Aufbau mit einem Agenten, welcher über seine Aktionen die Produktionssteuerung beeinflusst, und der Fokussierung auf die Liefertermineinhaltung eines Produktionssystems gegenüber dem Wertschöpfungsnetzwerk, größere Betrachtungsräume abgebildet werden können. Gegenüber den existierenden Ansätzen aus dem Stand der Wissenschaft und Technik (vgl. Abschnitt 3.4) sind damit in der Anwendung größere Zustands- und Aktionsräume möglich.

Der trainierte ARIW-Regler weist dabei auch die Fähigkeit zur Generalisierung hinsichtlich aus dem Training unbekannten Ereignissen auf. Das ist unabhängig davon, ob sich diese Ereignisse im Zeitintervall ihres Auftretens oder der Lieferterminänderung unterscheiden. Im Falle eines absinkenden Zeitintervalls und damit häufigeren Ereignissen ist es dem ARIW-Regler in allen betrachteten Zeitintervallen möglich, die Liefertermineinhaltung gegenüber dem Szenario ohne Anwendung des ARIW-Regler zu verbessern. Die Reduktion der Konsequenzen aus den Ereignissen auf die Liefertermineinhaltung sinkt dabei jedoch von den bereits genannten 75 Prozent auf bis zu 19 Prozent ab. Für den Fall der zunehmenden Lieferterminänderungen ist ebenfalls weiterhin eine Verbesserung der Liefertermineinhaltung durch den ARIW-Regler möglich, wobei die Reduktion der Konsequenzen aus den Ereignissen auf die Liefertermineinhaltung auf bis zu 20 Prozent absinkt. Ab einem Grenzwert, der für die vorliegende Anwendungsumgebung der Halbleiterfertigung bei $\Delta FF_{Plan} > 1,0$ liegt, kann der ARIW-Regler keine Verbesserung der Liefertermineinhaltung mehr erzielen.

Zusammenfassend kann festgestellt werden, dass die lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung mittels des ARIW-Reglers in der Lage ist, für die vorliegende Anwendungsumgebung einer Front-End-Fertigung der Halbleiterindustrie bei Ereignissen aus dem Wertschöpfungsnetzwerk eine signifikante Verbesserung der Liefertermineinhaltung zu erreichen. Maßgeblichen Einfluss auf die tatsächliche Höhe der Verbesserung der Liefertermineinhaltung haben dabei das Zeitintervall des Auftretens von Ereignissen und deren Lieferterminänderung.

6.4 Zwischenfazit

Ausgehend vom Konzept der lernenden ereignisbasierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung wurde dieses im zurückliegenden Abschnitt implementiert, am Beispiel einer modellierten Halbleiterfertigung angewendet und validiert. Damit kann die vierte Teilforschungsfrage, wie eine lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung implementiert und simulativ getestet werden kann, beantwortet werden.

Die Auswertung des Trainings (vgl. Abschnitt 6.3.2.3) zeigt dabei, dass durch Hyperparameter-Optimierung und Variation der Agentenparameter gegenüber der Basislinie eine Feinabstimmung und Verbesserung des Agentenverhaltens erreicht werden kann. Die Verbesserung des Verlaufs der Belohnungskurve hinsichtlich der Dauer bis zur Konvergenz, dem erreichten Maximalwert der Belohnung und der Stabilität der Belohnungskurve wird insbesondere durch eine Anpassung der Lernrate, des Entropiefaktors und der Flussfaktor-Anpassung des Agenten erzielt. Modellparameter sind jedoch von der Anwendungsumgebung abhängig, weshalb bei anderen Anwendungsumgebungen eine erneute Variation der Modellparameter anhand des dargestellten Vorgehens zu empfehlen ist.

Die Fähigkeit des ARIW-Reglers zur Generalisierung hinsichtlich aus dem Training unbekannter Ereignisse des Wertschöpfungsnetzwerks konnte anhand der Applikation gezeigt werden (vgl. Abschnitt 6.3.3.3). Für die vorliegende Anwendungsumgebung der Halbleiterfertigung wird dabei eine signifikante Reduktion der Konsequenzen von Ereignissen aus dem Wertschöpfungsnetzwerk auf die Liefertermineinhaltung um bis zu 75 Prozent möglich. Entscheidend für die Höhe der Reduktion sind das Zeitintervall des Auftretens von

Ereignissen, folglich die Anzahl von Ereignissen, sowie deren Lieferterminänderung. Für das Zeitintervall gilt dabei, dass es dem entwickelten ARIW-Regler stets möglich ist, die Liefertermineinhaltung gegenüber dem Szenario ohne Anwendung des ARIW-Reglers zu verbessern. Die Reduktion der Konsequenzen der Ereignisse auf die Liefertermineinhaltung vermindert sich dabei jedoch mit abnehmendem Zeitintervall. Für die Lieferterminänderung aus Ereignissen gilt, dass eine Verbesserung der Liefertermineinhaltung durch den ARIW-Regler bis zu einem anwendungsabhängigen Grenzwert möglich ist. Mit zunehmender Lieferterminänderung sinkt die Reduktion der Konsequenzen aus den Ereignissen auf die Liefertermineinhaltung jedoch.

Die lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung ist damit mittels des ARIW-Reglers in der Lage, bei Ereignissen aus dem Wertschöpfungsnetzwerk eine signifikante Verbesserung der Liefertermineinhaltung zu erreichen. Hierfür baut der ARIW-Regler auf ein RL-basiertes, lernfähiges Entscheidungsmodell auf und setzt ausgewählte Aktionen automatisiert über die Produktionssteuerung um. Die Automatisierung des Entscheidungsprozesses vermeidet dabei Medienbrüche zwischen einzelnen Systemen oder Personen sowie fehleranfällige manuelle Eingriffe in die Produktionssteuerung. Mittels der Ereignisse werden Wertschöpfungsnetzwerk und Produktionssteuerung zudem direkt miteinander verknüpft, was eine unmittelbare Reaktion auf die Ereignisse erlaubt. Darüber hinaus ermöglicht das lernfähige und adaptive Entscheidungsmodell des ARIW-Reglers eine Generalisierung und damit einen Einsatz auch in bislang unbekannten Situationen. Die lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung schafft damit ein resilientes Produktionssystem, welches in der Lage ist, aktiv und flexibel mit der ausgeprägten dynamischen Komplexität eines Wertschöpfungsnetzwerks umzugehen. Gleichzeitig wird die Grundlage für ein selbstoptimierendes Wertschöpfungsnetzwerk geschaffen, das durch die effektive Mitigation von Ereignissen effizientere Prozesse und eine potenzielle Reduktion von Sicherheitsbeständen ermöglicht.

7 Reflexion

Nach der Konzeption der lernenden ereignisbasierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung in Abschnitt 5 und dem Nachweis ihrer grundlegenden Funktion anhand der Implementierung in Abschnitt 6 folgt in diesem Abschnitt die Reflexion hinsichtlich den Anforderungen und der Forschungsfrage. Im Folgenden ist deshalb zu klären, ob die in Abschnitt 4.1 definierten Anforderungen erfüllt werden (vgl. Abschnitt 7.1) und ob die zentrale Forschungsfrage aus Abschnitt 1.3 beantwortet werden kann (vgl. Abschnitt 7.2).

7.1 Bewertung der Erfüllung der Anforderungen

Mit der Unterscheidung in Anforderungen aus dem Anwendungszusammenhang, an das Entscheidungsmodell und an die Systemarchitektur wurden in Abschnitt 4.1 zehn Anforderungen definiert und strukturiert. Nachfolgend werden diese Anforderungen hinsichtlich ihrer Erfüllung mit der im Rahmen dieser Arbeit entwickelten lernenden ereignisbasierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung abgeglichen (vgl. Abschnitt 7.1.1; Abschnitt 7.1.2; Abschnitt 7.1.3). Dabei zeigt sich, dass alle Anforderungen erfolgreich umgesetzt und damit erfüllt werden konnten.

7.1.1 Anforderungen aus dem Anwendungszusammenhang

Für den Anwendungszusammenhang ist es von besonderer Relevanz, dass eine *Integration von Wertschöpfungsnetzwerk und Produktionssteuerung (A1)* erreicht wird. Umgesetzt wurde diese Integration im Konzept des ARIW, ein Regelkreis an der Schnittstelle zwischen Wertschöpfungsnetzwerk und Produktionssteuerung. Dabei bildet der ARIW-Regler die Schnittstelle, die Ereignisse aus dem Wertschöpfungsnetzwerk unmittelbar verarbeitet und Maßnahmen ableitet, welche die Liefertermineinhaltung verbessern (vgl. Abschnitt 5.3.1). Die Anforderung ist somit vollumfänglich erfüllt.

Aufgrund von ihrer besonderen Charakteristik ist darüber hinaus die *Berücksichtigung des Fertigungsprinzips der komplexen Werkstattfertigung (A2)* notwendig. Diese spiegelt sich insbesondere an zwei Stellen wider: Einerseits bei der Gestaltung des Zustandsraums. Neben den relativen Größen verschiedener Flussfaktorausprägungen erlaubt dieser mit der Maschinengruppe und dem Fertigstellungsgrad auch eine Einordnung, welche Lose in ihren wiedereintretenden Prozessflüssen des Arbeitsplans derzeit um welche Ressourcen konkurrieren. Andererseits auch bei der Auslegung des Aktionsraums, welcher die Priorisierung von Losen verändert, die anschließend vom Dispatching der bereits bestehenden Produktionssteuerung in komplexen Werkstattfertigungen verarbeitet werden kann (vgl. Abschnitt 5.3.2.1). Die Anforderung gilt deshalb als voll erfüllt.

Hinsichtlich der Abbildung der *Granularität des relevanten Betrachtungsraums (A3)* ist es notwendig, dass der Betrachtungsraum weit genug abstrahiert wird, damit nicht nur Einzelmaschinen, sondern auch das Produktionssystem als Ganzes berücksichtigt werden können. Aufgrund des übergeordneten Regelungscharakters baut der ARIW-Regler auf einen singulären Agenten auf, welcher nicht an Maschinengruppen Dispatching-Entscheidungen für Einzelmaschinen trifft, sondern die Liefertermineinhaltung des gesamten Produktionssystems bei Ereignissen aus dem Wertschöpfungsnetzwerk optimiert. Gleichzeitig ist der Aktionsraum dennoch feingranular genug, um die Priorität einzelner Lose gezielt zu beeinflussen (vgl. Abschnitt 5.3.2.1). Die Anforderung kann entsprechend als umfassend erfüllt bewertet werden.

7.1.2 Anforderungen an das Entscheidungsmodell

Resultierend aus dem Anwendungszusammenhang ist für ein Entscheidungsmodell auf Ebene der Produktionssteuerung die *Berücksichtigung externer Ereignisse (E1)* notwendig. Der ARIW wurde deshalb explizit auf die Verarbeitung von Ereignissen aus dem Wertschöpfungsnetzwerk ausgerichtet. Dabei hat der Ereignisraum einen unmittelbaren Einfluss auf den Zustandsraum, welcher wiederum den trainierten Agenten veranlasst, abhängig von der Situation eine geeignete Aktion als Korrekturmaßnahme auszuwählen (vgl. Abschnitt 5.3.2.1). Damit konnte die Anforderung in Gänze erfüllt werden.

Für die Umsetzung von Korrekturmaßnahmen zur Reaktion auf externe Ereignisse durch das Entscheidungsmodell muss der *Aktionsraum zur Einwirkung auf diskrete Aufträge (E2)* gestaltet sein. Hierfür werden einerseits im Zustandsraum die Rückführgrößen aller betrachteten Lose abgebildet, andererseits erlaubt der Aktionsraum ein Einwirken auf jedes einzelne Los mittels der Stellgröße des Steuer-Flussfaktors (vgl. Abschnitt 5.3.2.1). Demnach wird die Anforderung als durchweg erfüllt bewertet.

Aufgrund der statischen und dynamischen Komplexität in der komplexen Werkstattfertigung wie auch im Wertschöpfungsnetzwerk muss eine *Adaptivität des Entscheidungsmodells (E3)* zur Anpassung von Parametern gegeben sein. Alle Kenngrößen des Ereignis-, Zustands- und Aktionsraums sowie die Modellparameter des Agenten sind deshalb nicht auf definierte Werte festgelegt, sondern als datenbasiertes adaptives Modell gestaltet (vgl. Abschnitt 5.3.2.1; Abschnitt 5.3.2.2). Die Anforderung ist somit vollständig erfüllt.

Daran schließt die *Lernfähigkeit des Entscheidungsmodells (E4)* an, die es ermöglicht, aus Daten neue Entscheidungen zur optimierten Nutzung des Aktionsraums zu lernen und damit auf unbekannte Situationen zu generalisieren. Realisiert wurde die Lernfähigkeit durch den RL-basierten Agenten im ARIW-Regler. RL erlaubt es dabei, ohne eine vorgegebene Strategie anhand der Situation, abgebildet über den Zustandsraum, Entscheidungen zur Maximierung der Belohnung zu treffen. Die Fähigkeit zur Generalisierung zeigt sich in der Anwendung, in welcher trotz veränderter Kenngrößen der Anwendungsumgebung eine signifikante Verbesserung der Liefertermineinhaltung erreicht werden konnte (vgl. Abschnitt 5.3.2.2; Abschnitt 6.3.3). Damit ist die Anforderung voll erfüllt.

Zur Reaktion auf Ereignisse aus dem Wertschöpfungsnetzwerk auf der Ebene der Produktionssteuerung muss zudem eine hohe *Geschwindigkeit des Entscheidungsmodells (E5)* erreicht werden. Die erforderliche Geschwindigkeit bezieht sich dabei nicht auf die vor dem Einsatz des Entscheidungsmodells notwendige Trainingsdauer, sondern auf die Geschwindigkeit der Entscheidungsfindung nach Auftreten eines Ereignisses. Der Algorithmus A3C des ARIW-Reglers basiert auf KNN. Diese zeichnen sich dadurch aus, dass das Training, insbesondere bei komplexen Problemstellungen und Datensätzen, zwar sehr zeitaufwändig sein kann, trainierte KNN jedoch sehr schnell ausgeführt werden können. Der Kommunikationsaufwand mit weiteren IT-Systemen benötigt dabei typischerweise deutlich mehr Zeit als die Anwendung des trainierten KNN auf einen Datensatz, konkret

einen Zustand zur Entscheidung über eine Aktion (vgl. Abschnitt 6.3.2; Abschnitt 6.3.3). Die Anforderung ist somit umfassend erfüllt.

7.1.3 Anforderungen an die Systemarchitektur

Ein hoher *Automatisierungsgrad des Systems (S1)* trägt einerseits zur Geschwindigkeit des Systems zur lernenden ereignisbasierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung bei, andererseits auch zur Reduktion von Fehlern in den Entscheidungen des Entscheidungsmodells. Der hohe Automatisierungsgrad wird dadurch erreicht, dass sowohl das Training als auch die Anwendung des ARIW automatisiert ablaufen und Entscheidungen mittels eines datenbasierten Modells durch den Algorithmus erlernt werden. Daten werden über maschinenlesbare Schnittstellen ohne menschlichen Handlungsbedarf ausgetauscht (vgl. Abschnitt 5.3.2.2; Abschnitt 5.3.2.3; Abschnitt 6.2). Aus diesen Gründen kann die Anforderung als vollständig erfüllt bewertet werden.

Durch das Agieren an der Schnittstelle zwischen Wertschöpfungsnetzwerk und Produktionssteuerung ist es notwendig, eine *Integration in die bestehende Produktions-IT-Landschaft (S2)* eines Unternehmens vorzusehen. Hierfür wurde die Architektur zur Einbettung des ARIW in die Produktions-IT-Landschaft erarbeitet, welche modulare und erweiterbare Schnittstellen vorsieht (vgl. Abschnitt 5.3.3). Damit kann auch diese Anforderung gänzlich erfüllt werden.

7.2 Beantwortung der Forschungsfrage

Die Formulierung von Forschungsfragen ist eine zentrale Leistung im Forschungsprozess. Für die vorliegende Arbeit wurde in Abschnitt 1.3 die nachfolgende Forschungsfrage formuliert:

Wie kann eine **lernende**, auf **Ereignissen aus dem Wertschöpfungsnetzwerk basierte** Optimierung der **Produktionssteuerung** für die **komplexe Werkstattfertigung** gestaltet werden?

Ausgehend von dieser zentralen Forschungsfrage wurden vier Teilforschungsfragen abgeleitet. Die Teilforschungsfragen wurden dabei in den vorangegangenen Abschnitten bereits eingehend beantwortet und sollen nachfolgend nochmals zusammengefasst werden.

1. Was sind die relevanten Bestandteile der komplexen Werkstattfertigung und ihres Wertschöpfungsnetzwerks, um die Produktionssteuerung basierend auf Ereignissen aus dem Wertschöpfungsnetzwerk zu optimieren?

In Abschnitt 2 wurden relevante Grundlagen einer lernenden ereignisbasierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung erarbeitet. Die relevanten Bestandteile sind dabei die komplexe Werkstattfertigung und ihr Wertschöpfungsnetzwerk mit den einhergehenden Ereignissen und möglicher IT-Unterstützung, die PPS mit ihren Aufgaben, Zielen, Messgrößen und möglicher IT-Unterstützung sowie lernende Systeme, insbesondere durch ML und Regelkreise.

2. Welche Anforderungen werden an eine lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung gestellt?

In Abschnitt 4 wurden zehn Anforderungen an eine lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung definiert. Diese Anforderungen gliedern sich in Anforderungen aus dem Anwendungszusammenhang, an das Entscheidungsmodell und an die Systemarchitektur.

3. Wie kann das Konzept einer lernenden ereignisbasierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung aufgebaut werden?

In Abschnitt 5 wurde die lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung konzipiert. Die hierfür erforderlichen Artefakte sind der Regelkreis, die Methode und die Architektur für die lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung. Diese drei Artefakte wirken zusammen, um die Liefertermineinhaltung bei Ereignissen aus dem Wertschöpfungsnetzwerk zu verbessern und hierfür Entscheidungen hinsichtlich Korrekturmaßnahmen zur Priorisierung von Losen eines Produktionssystems zu treffen.

4. Wie kann eine lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung implementiert und simulativ getestet werden?

In Abschnitt 6 wurde die lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung implementiert und am Beispiel einer

Halbleiterfertigung simulativ getestet. Eine Grenzbetrachtung erlaubt dabei eine Einschätzung der weiterführenden Anwendungsmöglichkeiten.

In Summe ergibt sich aus den Antworten auf die Teilforschungsfragen die Antwort auf die zentrale Forschungsfrage, welche damit vollumfänglich beantwortet werden kann. Damit wird das Ziel der Arbeit, die Optimierung der Liefertermineinhaltung bei Ereignissen aus dem Wertschöpfungsnetzwerk in der Produktionssteuerung der komplexen Werkstattfertigung, durch die Beantwortung der zentralen Forschungsfrage erreicht. Dies stellt gleichzeitig auch den Abschluss des iterativen Forschungsprozesses dieser Arbeit dar und bildet die Grundlage dafür, dass die Erkenntnisse aus dem Forschungsprozess anhand des Information System Research Frameworks einerseits in die Wissensbasis und andererseits in die Praxis einfließen können.

8 Zusammenfassung und Ausblick

8.1 Zusammenfassung

Produzierende Unternehmen bewegen sich heute in einem Umfeld, in der der Kundennutzen und damit auch die Liefertermineinhaltung im Fokus der unternehmerischen Aktivitäten steht. Die Komplexität des Umfelds und der Märkte sowie innerhalb des Unternehmens steigt gleichzeitig kontinuierlich an, weshalb der Leistungserstellungsprozess zunehmend in Wertschöpfungsnetzwerken organisiert wird. Trotz Fortschritten in der IT wird das Potenzial der Produktionssteuerung hinsichtlich der Erreichung logistischer Zielgrößen und damit auch der Kompensation von Ereignissen aus dem Wertschöpfungsnetzwerk jedoch noch nicht ausgeschöpft. Eine besondere Herausforderung ist dies für komplexe Werkstattfertigungen mit ihrer immanenten statischen und dynamischen Komplexität.

Es war deshalb das **Ziel**, im Rahmen dieser Arbeit eine lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung zu entwickeln. Diese optimiert die Liefertermineinhaltung bei Ereignissen aus dem Wertschöpfungsnetzwerk durch Korrekturmaßnahmen auf der Ebene der Produktionssteuerung in komplexen Werkstattfertigungen. Hierfür ist ein lernfähiges Entscheidungsmodell notwendig, um eine kontinuierliche Anpassung über die Zeit zu ermöglichen. Gleichzeitig werden bislang bei Ereignissen aus dem Wertschöpfungsnetzwerk übliche manuelle Eingriffe in die Produktionssteuerung vermieden und die Resilienz des Produktionssystems gesteigert.

Der **Forschungsprozess** und diese Arbeit wurden anhand des Information Systems Research Framework nach Hevner et al. (2004) und den Phasen der angewandten Forschung nach Ulrich (1981) strukturiert. Im ersten Schritt wurden hierfür Grundlagen der komplexen Werkstattfertigung und deren Wertschöpfungsnetzwerke, der PPS sowie von lernenden Systemen erarbeitet (vgl. Abschnitt 2). Darauf aufbauend wurde der Stand der Wissenschaft und Technik hinsichtlich existierender Ansätze zur Reaktion auf Ereignisse aus dem Wertschöpfungsnetzwerk, zur Ereignisverarbeitung in der Produktionssteuerung und zur Nutzung von bestärkendem Lernen in der Produktionssteuerung der Werkstattfertigung

aufbereitet (vgl. Abschnitt 3). Der Stand der Wissenschaft und Technik wurde anschließend anhand von zehn definierten Anforderungen, welche sich in Anforderungen aus dem Anwendungszusammenhang, an das Entscheidungsmodell und an die Systemarchitektur gliedern, eingeordnet und bewertet (vgl. Abschnitt 4). Ausgehend von den Anforderungen wurde die lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung konzeptioniert und die Artefakte des Regelkreises, der Methode und der Architektur entwickelt (vgl. Abschnitt 5). Anschließend wurde diese als Softwareartefakt implementiert, trainiert und angewendet. Damit konnte gezeigt werden, dass die lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung eine signifikante Verbesserung der Liefertermineinhaltung bei Ereignissen aus dem Wertschöpfungsnetzwerk ermöglicht (vgl. Abschnitt 6). Mit der Bewertung anhand der Anforderungen und den vorliegenden Ergebnissen konnte die Forschungsfrage nach der Gestaltung einer lernenden, auf Ereignissen aus dem Wertschöpfungsnetzwerk basierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung zum Abschluss des Forschungsprozesses beantwortet werden (vgl. Abschnitt 7).

Ergebnis des Forschungsprozesses und der vorliegenden Arbeit sind die Artefakte der lernenden ereignisbasierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung in Form des Regelkreises, konkret des ARIW, der Methode und der Architektur sowie der Implementierung. Damit wird ein resilientes Produktionssystem geschaffen, welches in der Lage ist, aktiv und flexibel mit der ausgeprägten dynamischen Komplexität eines Wertschöpfungsnetzwerks umzugehen. Gleichzeitig wird die Grundlage für ein selbstoptimierendes Wertschöpfungsnetzwerk gebildet.

Für den Aufbau und die Anwendung der lernenden ereignisbasierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung müssen in einem produzierenden Unternehmen diverse **Voraussetzungen** erfüllt werden. So ist für das Training des Entscheidungsmodells ein Simulationsmodell des Produktionssystems notwendig. Des Weiteren müssen Mechanismen und IT-Systeme vorhanden sein, um Ereignisse aus dem Wertschöpfungsnetzwerk automatisiert zu erkennen und zu verarbeiten. Darüber hinaus werden maschinenlesbare Schnittstellen benötigt, um den Ereignis- und Zustandsraum des ARIW-Reglers mit relevanten Daten zu befüllen sowie dessen Entscheidungen über Aktionen mittels der Produktionssteuerung umzusetzen.

Es sind jedoch auch die **Grenzen** der lernenden ereignisbasierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung zu berücksichtigen. Das in der vorliegenden Arbeit gestaltete Konzept ist durch die Nutzung der vermeintlichen Nachteile der komplexen Werkstattfertigung als Handlungsspielraum für den ARIW-Regler auf den Einsatz in diesem Fertigungsprinzip beschränkt. Des Weiteren können nur Ereignisse verarbeitet werden, welche durch kurzfristige, automatisierbare Korrekturmaßnahmen in Form von Prioritätsveränderungen mitigiert werden können. Auch ist die Anzahl der Ereignisse in einem Zeitabschnitt sowie die Stärke der Auswirkungen der Ereignisse, welche durch die lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung kompensiert werden kann, begrenzt. Die derzeit verfügbare Hardware limitiert darüber hinaus die Größe des Betrachtungsraums hinsichtlich der Anzahl der Maschinengruppen und der Lose. Diese Grenzen werden auch mit den Ansatzpunkten für weitere Forschungsarbeiten im folgenden Abschnitt adressiert.

Unter Berücksichtigung der Voraussetzungen und Grenzen bietet die lernende ereignisbasierte Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung jedoch vielfältige **Potenziale**. Infolge von Korrekturmaßnahmen in der Produktionssteuerung wird deren Fähigkeit zur Sicherstellung der Erreichung logistischer Zielgrößen weiter ausgeschöpft, wodurch die Liefertermineinhaltung verbessert und mehr Kundenzufriedenheit geschaffen werden kann. Gleichzeitig wird die Resilienz bei Ereignissen aus dem Wertschöpfungsnetzwerk erhöht, was perspektivisch einen Abbau von Sicherheitsbeständen im Wertschöpfungsnetzwerk ermöglicht. Der im Rahmen dieser Arbeit entwickelte Ansatz der lernenden ereignisbasierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung bietet damit ein enormes Zukunftspotenzial für produzierende Unternehmen in einem komplexen Umfeld mit einer Vielzahl von Ereignissen aus dem Wertschöpfungsnetzwerk.

8.2 Ausblick

Die Beantwortung der zentralen Forschungsfrage nach der Gestaltung einer lernenden, auf Ereignissen aus dem Wertschöpfungsnetzwerk basierten Optimierung der Produktionssteuerung für die komplexe Werkstattfertigung bringt zahlreiche relevante Erkenntnisse

hervor, welche in die Wissensbasis anderer Forscherinnen und Forscher einfließen. Damit eröffnet die vorliegende Arbeit auch Ansatzpunkte für weitere Forschungsarbeiten.

So ist die Anwendung der lernenden ereignisbasierten Optimierung der Produktionssteuerung derzeit auf das Fertigungsprinzip der komplexen Werkstattfertigung beschränkt. Eine Untersuchung der Übertragbarkeit auf weitere Fertigungsprinzipien erlaubt potenziell den Transfer auf zusätzliche Anwendungen. Hierfür ist insbesondere zu analysieren, welche Handlungsmöglichkeiten das jeweilige Fertigungsprinzip bietet und wie resultierend daraus der Aktionsraum gestaltet sein muss.

Ausgedehnt werden können die Anwendungsmöglichkeiten des Ansatzes auch durch die Vergrößerung des Betrachtungsraums über die Produktionssteuerung hinaus. Wird die Auftragsfreigabe mit einbezogen, können zusätzlich Maßnahmen zur Anpassung von Losstarts und damit des Bestands des Produktionssystems durch den ARIW-Regler ausgelöst werden. Dabei ist jedoch zu beachten, dass mögliche Implementierungen durch die derzeit verfügbare Hardware und durch die Größe des Aktionsraums limitiert werden. Einen Ansatzpunkt zum Umgang hiermit bildet das hierarchische bestärkende Lernen (Sutton & Barto 2018, S. 461–463; Nachum et al. 2018, S. 2). Potenziell könnten damit im ersten Schritt Maschinengruppen und erst im zweiten Schritt einzelne Lose zur Beeinflussung ausgewählt werden. Ebenso wird es damit potenziell möglich, dass neben diskreten Aktionen auch kontinuierliche Parameter, beispielsweise die Steuer-Flussfaktoranpassung, verändert werden könnten. Hierfür werden diskrete und kontinuierliche Aktionen im Aktionsraum kombiniert (Hu & Kaneko 2021, S. 1–2; Hausknecht & Stone 2016, S. 2). Vor der Realisierung ist jedoch noch einige Forschung an geeigneten Algorithmen für hierarchisches bestärkendes Lernen notwendig.

Ein zusätzlicher Ansatz für weitere Forschungsarbeiten besteht in der Weiterentwicklung hin zu einem Multiagentensystem. Einerseits ist hierfür die Kombination mit existierenden Ansätzen des bestärkenden Lernens, welche auf die Optimierung interner Zielgrößen in der Produktionssteuerung der komplexen Werkstattfertigung abzielen, möglich. Andererseits ist auch eine Weiterentwicklung dahingehend vielversprechend, dass unterschiedliche Partner eines Wertschöpfungsnetzwerks jeweils eine lernende ereignisbasierte Optimierung der Produktionssteuerung einsetzen, deren Agenten dann in einem Multiagentensystem mit dem

Ziel der Optimierung der Liefertermineinhaltung des gesamten Wertschöpfungsnetzwerks kooperieren.

Literatur

Abadi et al. 2016

Abadi, Martin; Barham, Paul; Chen, Jianmin; Chen, Zhifeng; Davis, Andy; Dean, Jeffrey; Devin, Matthieu; Ghemawat, Sanjay; Irving, Geoffrey; Isard, Michael; Kudlur, Manjunath; Levenberg, Josh; Monga, Rajat; Moore, Sherry; Murray, Derek G.; Steiner, Benoit; Tucker, Paul; Vasudevan, Vijay; Warden, Pete; Wicke, Martin; Yu, Yuan; Zheng, Xiaoqiang, 2016.
TensorFlow: A System for Large-Scale Machine Learning.
In: *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation*.
12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16).
Savannah, GA, USA, 02.11.2016–04.11.2016, S. 265–283.
ISBN: 978-1-931971-33-1

Abele & Reinhart 2011

Abele, Eberhard; Reinhart, Gunther, 2011.
Zukunft der Produktion: Herausforderungen, Forschungsfelder, Chancen. München: Carl Hanser.
ISBN: 978-3-446-42595-8

adesso 2021

adesso, 2021.
Zum Status von KI in D-A-CH. Dortmund: adesso.
URL: <https://ki.adesso.de/ki-studie>
Zugriff am: 23.11.2021

Adhitya et al. 2007a

Adhitya, Arief; Srinivasan, Rajagopalan; Karimi, I. A., 2007.
A model-based rescheduling framework for managing abnormal supply chain events.
Computers & Chemical Engineering **31** (5-6), S. 496–518.
DOI: 10.1016/j.compchemeng.2006.07.002

Adhitya et al. 2007b

Adhitya, Arief; Srinivasan, Rajagopalan; Karimi, I. A., 2007.
Heuristic rescheduling of crude oil operations to manage abnormal supply chain events.
AIChE Journal **53** (2), S. 397–422.
DOI: 10.1002/aic.11069

Aelker et al. 2013

Aelker, Judith; Bauernhansl, Thomas; Ehm, Hans, 2013.
Managing Complexity in Supply Chains: A Discussion of Current Approaches on the Example of the Semiconductor Industry.
Procedia CIRP **7**, S. 79–84.
DOI: 10.1016/j.procir.2013.05.014

- Alpaydin 2020** Alpaydin, Ethem, 2020.
Introduction to machine learning. 4th ed. Cambridge, MA, USA: MIT Press. Adaptive Computation and Machine Learning. ISBN: 978-0-262-04379-3
- Altenmüller et al. 2020** Altenmüller, Thomas; Stüker, Tillmann; Waschneck, Bernd; Kuhnle, Andreas; Lanza, Gisela, 2020.
Reinforcement learning for an intelligent and autonomous production control of complex job-shops under time constraints.
Production Engineering **14** (3), S. 319–328.
DOI: 10.1007/s11740-020-00967-8
- Álvarez & Díaz 2011** Álvarez, Esther; Díaz, Fernando, 2011.
A web-based approach for exceptions management in the supply chain.
Robotics and Computer-Integrated Manufacturing **27** (4), S. 681–686.
DOI: 10.1016/j.rcim.2010.12.004
- Anderson 1995** Anderson, John R., 1995.
Learning and memory: An integrated approach. New York, NY, USA: Wiley.
ISBN: 978-0471115960
- Anderson & Krathwohl 2001** Anderson, Lorin W.; Krathwohl, David R., 2001.
A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives. New York, NY, USA: Addison Wesley Longman.
ISBN: 0-8013-1903-X
- Arping 1977** Arping, Hermann, 1977.
Beitrag zur rechnergestützten Fertigungsregelung mit automatisierter Entscheidungshilfe im Störfall bei Einzel- und Kleinserienfertigung.
Aachen, RWTH, Diss., 1977
- Arulkumaran et al. 2017** Arulkumaran, Kai; Deisenroth, Marc Peter; Brundage, Miles; Bharath, Anil Anthony, 2017.
Deep Reinforcement Learning: A Brief Survey.
IEEE Signal Processing Magazine **34** (6), S. 26–38.
DOI: 10.1109/MSP.2017.2743240
- Arviv et al. 2016** Arviv, Kfir; Stern, Helman; Edan, Yael, 2016.
Collaborative reinforcement learning for a two-robot job transfer flow-shop scheduling problem.
International Journal of Production Research **54** (4), S. 1196–1209.
DOI: 10.1080/00207543.2015.1057297

- Baader & Montanus 2008** Baader, Andreas; Montanus, Sven, 2008. Transparency in Global Supply Chain Networks – Methods and Tools for Integrated Supply Chain Event Management. In: Ijoui, Raschid; Emmerich, Heike; Cey, Michael (Hrsg.): *Strategies and tactics in supply chain event management*. Berlin: Springer, S. 3–11. ISBN: 978-3-540-73765-0
- Bangsow 2011** Bangsow, Steffen, 2011. *Praxishandbuch Plant Simulation und SimTalk: Anwendung und Programmierung in über 150 Beispiel-Modellen*. München: Carl Hanser. ISBN: 978-3-446-42782-2
- Bangsow 2015** Bangsow, Steffen, 2015. *Tecnomatix plant simulation: Modeling and programming by means of examples*. Cham: Springer. ISBN: 978-3-319-19502-5
- Bauer et al. 2017** Bauer, Dennis; Stock, Daniel; Bauernhansl, Thomas, 2017. Movement Towards Service-orientation and App-orientation in Manufacturing IT. *Procedia CIRP* **62**, S. 199–204. DOI: 10.1016/j.procir.2016.06.079
- Bauer et al. 2019a** Bauer, Dennis; Schlereth, Andreas; Maier, Florian; Ponsignon, Thomas; Waschneck, Bernd, 2019. Automated Framework Based on Supply Chain Events. In: De Luca, Cristina (Hrsg.): *A Semi40 Project Prospective*. Villach, Österreich: Semi40 Consortium, S. 36–41
- Bauer et al. 2019b** Bauer, Dennis; Bauernhansl, Thomas; Sauer, Alexander, 2019. Enhanced Classification of Events for Manufacturing Companies in Supply Networks. *Procedia CIRP* **81**, S. 87–92. DOI: 10.1016/j.procir.2019.03.016
- Bauer et al. 2019c** Bauer, Dennis; Schlereth, Andreas; Bauernhansl, Thomas, 2019. Systemdynamik-Ansätze zur Optimierung der PPS: Nutzen und Anwendung in der Halbleiterindustrie. *wt Werkstattstechnik* **109** (4), S. 225–229. DOI: 10.5281/zenodo.2653050
- Bauer et al. 2020a** Bauer, Dennis; Bauernhansl, Thomas; Sauer, Alexander, 2020. Approach for an Adaptive Control Loop between Supply Network and Manufacturing. *Procedia CIRP* **93**, S. 1552–1557. DOI: 10.1016/j.procir.2020.04.058
- Bauer et al. 2020b** Bauer, Dennis; Bauernhansl, Thomas; Sauer, Alexander, 2020. Wertschöpfungsnetzwerk und Produktion integrieren: Ein adaptiver, regelkreisbasierter Ansatz. *wt Werkstattstechnik* **110** (3), S. 92–97

- Bauer et al. 2021a** Bauer, Dennis; Bauernhansl, Thomas; Sauer, Alexander, 2021. Improvement of Delivery Reliability by an Intelligent Control Loop between Supply Network and Manufacturing. *Applied Sciences* **11** (5), S. 2205. DOI: 10.3390/app11052205
- Bauer et al. 2021b** Bauer, Dennis; Böhm, Markus; Bauernhansl, Thomas; Sauer, Alexander, 2021. Increased resilience for manufacturing systems in supply networks through data-based turbulence mitigation. *Production Engineering* **15** (3-4), S. 385–395. DOI: 10.1007/s11740-021-01036-4
- Bauernhansl et al. 2014** Bauernhansl, Thomas; Schatz, Anja; Jäger, Jens, 2014. Komplexität bewirtschaften – Industrie 4.0 und die Folgen. *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb* **109** (5), S. 347–350. DOI: 10.3139/104.111140
- Bauernhansl 2016** Bauernhansl, Thomas, 2016. Mehr Chancen als Risiken. *Automationspraxis* (1-2), S. 22–23
- Bauernhansl 2017a** Bauernhansl, Thomas, 2017. Die Vierte Industrielle Revolution. In: Vogel-Heuser, Birgit; Bauernhansl, Thomas; ten Hompel, Michael (Hrsg.): *Handbuch Industrie 4.0*. Berlin: Springer Vieweg, S. 1–31. ISBN: 978-3-662-53253-9
- Bauernhansl 2017b** Bauernhansl, Thomas, 2017. Mass Personalization - der nächste Schritt. *wt Werkstattstechnik* **107** (3), S. 107
- Bauernhansl et al. 2018** Bauernhansl, Thomas; Hörcher, Günter; Bressner, Markus; Röhm, Maren, 2018. *MANUFUTURE-DE: Ermittlung prioritärer Forschungsthemen für die nachhaltige Ausgestaltung von europäischen Forschungsprogrammen für die produzierende Industrie bis 2030*. Stuttgart: Fraunhofer IPA
- Bauernhansl & Miehe 2020** Bauernhansl, Thomas; Miehe, Robert, 2020. Industrielle Produktion – Historie, Treiber und Ausblick. In: Bauernhansl, Thomas (Hrsg.): *Fabrikbetriebslehre 1*. Berlin: Springer, S. 1–33. ISBN: 978-3-662-44538-9

- Bearzotti et al. 2008** Bearzotti, Lorena; Salomone, Enrique; Chiotti, Omar J., 2008. An autonomous multi-agent approach to supply chain event management. In: IEEE (Hrsg.): *Proceedings of the 2008 IEEE International Conference on Service Operations and Logistics, and Informatics*. 2008 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI 2008). Peking, China, 12.10.2008–15.10.2008, S. 524–529. ISBN: 978-1-4244-2012-4
- Bearzotti et al. 2012** Bearzotti, Lorena A.; Salomone, Enrique; Chiotti, Omar J., 2012. An autonomous multi-agent approach to supply chain event management. *International Journal of Production Economics* **135** (1), S. 468–478. DOI: 10.1016/j.ijpe.2011.08.023
- Bechte 1980** Bechte, Wolfgang, 1980. *Steuerung der Durchlaufzeit durch belastungsorientierte Auftragsfreigabe bei Werkstattfertigung*. Hannover, Univ., Diss., 1980
- Begemann 2005** Begemann, Carsten, 2005. *Terminorientierte Kapazitätssteuerung in der Fertigung*. Hannover: PZH Produktionstechn. Zentrum. Berichte aus dem IFA 02/2005. Hannover, Univ., Diss., 2005. ISBN: 3-939026-01-8
- Behdani 2013** Behdani, Behzad, 2013. *Handling disruptions in supply chains: An integrated framework and an agent-based model*. Delft: Next Generation Infrastructures Foundation. NGInfra PhD thesis series on infrastructures 54. Delft, Niederlande, Univ., Diss., 2013. ISBN: 9789079787432
- Bishop 2006** Bishop, Christopher M., 2006. *Pattern recognition and machine learning*. New York, NY, USA: Springer. Information science and statistics. ISBN: 978-0387-31073-2
- Blackstone 2013** Blackstone, John H., 2013. *APICS dictionary*. Fourteenth edition. Chicago: APICS. ISBN: 978-0-9882146-1-3
- Bode & Wagner 2015** Bode, Christoph; Wagner, Stephan M., 2015. Structural drivers of upstream supply chain complexity and the frequency of supply chain disruptions. *Journal of Operations Management* **36** (1), S. 215–228. DOI: 10.1016/j.jom.2014.12.004

- Bodendorf 2006** Bodendorf, Freimut, 2006.
Daten- und Wissensmanagement. 2., aktualisierte und erweiterte Auflage. Berlin: Springer. Springer-Lehrbuch. ISBN: 978-3-540-28743-8
- Brambring 2017** Brambring, Felix, 2017.
Steigerung der Datenintegrität in der Produktionssteuerung. Aachen: Apprimus-Verlag. Ergebnisse aus der Produktionstechnik 25.
Aachen, RWTH, Diss., 2017.
ISBN: 978-3-86359-537-1
- Braun 2016** Braun, Anja-Tatjana, 2016.
Optimierungsmodell zur Planung der direkten Wiederverwendung bei der Vermietung mobiler und langlebiger Investitionsgüter in Closed-Loop Supply Chains. Stuttgart: Fraunhofer Verlag. Stuttgarter Beiträge zur Produktionsforschung 55.
Stuttgart, Univ., Diss., 2016.
ISBN: 978-3-8396-1052-7
- Breithaupt 2001** Breithaupt, Jan-Wilhelm, 2001.
Rückstandsorientierte Produktionsregelung von Fertigungsbereichen: Grundlagen und Anwendung. Düsseldorf: VDI-Verlag. Fortschritt-Berichte VDI Reihe 2, Fertigungstechnik 571.
Hannover, Univ., Diss., 2000.
ISBN: 9783183571024
- Brockman et al. 2016** Brockman, Greg; Cheung, Vicki; Pettersson, Ludwig; Schneider, Jonas; Schulman, John; Tang, Jie; Zaremba, Wojciech, 2016.
OpenAI Gym.
Computing Research Repository, S. 1–4.
arXiv: 1606.01540
- Bruns & Dunkel 2010** Bruns, Ralf; Dunkel, Jürgen, 2010.
Event-driven architecture: Softwarearchitektur für ereignisgesteuerte Geschäftsprozesse. Berlin: Springer. ISBN: 978-3-642-02439-9
- Busch 1987** Busch, U., 1987.
Bestandsregelte Durchfluß-Steuerung.
CIM-Management (1), S. 18–26
- Cardon et al. 2018** Cardon, Dominique; Cointet, Jean-Philippe; Mazières, Antoine, 2018.
Neurons Spike Back: The Invention of Inductive Machines and the Artificial Intelligence Controversy.
Réseaux **211** (5), S. 173–220.
DOI: 10.3917/res.211.0173

- Caspi et al. 2017** Caspi, Itai; Leibovich, Gal; Endrawis, Shadi; Novik, Gal, 2017. *Reinforcement Learning Coach: An open source research framework for training and evaluating reinforcement learning agents*. DOI: 10.5281/zenodo.1134898
- Chen et al. 2020** Chen, Ronghua; Yang, Bo; Li, Shi; Wang, Shilong, 2020. A self-learning genetic algorithm based on reinforcement learning for flexible job-shop scheduling problem. *Computers & Industrial Engineering* **149**, S. 106778. DOI: 10.1016/j.cie.2020.106778
- Chien et al. 2008** Chien, Chen-Fu; Dauzère-Pérès, Stéphane; Ehm, Hans; Fowler, John W.; Jiang, Zhibin; Krishnaswamy, Shekar; Mönch, Lars; Uzsoy, Reha, 2008. Modeling and analysis of semiconductor manufacturing in a shrinking world: Challenges and successes. In: Mason, Scott J.; Hill, Ray R; Mönch, Lars; Rose, Oliver; Jefferson, Thomas; Fowler, John W. (Hrsg.): *Proceedings of the 2008 Winter Simulation Conference*. 2008 Winter Simulation Conference (WSC 2008). Miami, FL, USA, 07.12.2008–10.12.2008, S. 2093–2099. ISBN: 978-1-4244-2707-9
- Chien et al. 2020** Chien, Chen-Fu; Lin, Yun-Siang; Lin, Sheng-Kai, 2020. Deep reinforcement learning for selecting demand forecast models to empower Industry 3.5 and an empirical study for a semiconductor component distributor. *International Journal of Production Research* **58** (9), S. 2784–2804. DOI: 10.1080/00207543.2020.1733125
- Chmielewicz 1994** Chmielewicz, Klaus, 1994. *Forschungskonzeptionen der Wirtschaftswissenschaft*. 3., unveränd. Aufl. Stuttgart: Schäffer-Poeschel. Sammlung Poeschel 92. ISBN: 3-7910-9197-2
- Chopra 2018** Chopra, Sunil, 2018. *Supply Chain Management: Strategy, Planning, and Operation*. 7th ed. Boston, MA, USA: Pearson. ISBN: 978-0134731889
- Christopher 2011** Christopher, Martin, 2011. *Logistics & supply chain management*. 4. ed. Harlow, England: Financial Times Prentice Hall. ISBN: 978-0-273-73112-2

- Corsten & Gössinger 2016** Corsten, Hans; Gössinger, Ralf, 2016.
Produktionswirtschaft: Einführung in das industrielle Produktionsmanagement. 14., überarbeitete und erweiterte Auflage. Berlin: De Gruyter Oldenbourg. Lehr- und Handbücher der Betriebswirtschaftslehre.
ISBN: 978-3-11-045277-8
- DIN 66001** DIN 66001:1983.
Sinnbilder und ihre Anwendung
- DIN 8580** DIN 8580:2003.
Fertigungsverfahren: Begriffe, Einteilung
- DIN EN 60027-6** DIN EN 60027-6:2008.
Formelzeichen für die Elektrotechnik: Teil 6: Steuerungs- und Regelungstechnik
- DIN IEC 60050-351** DIN IEC 60050-351:2013.
Internationales Elektrotechnisches Wörterbuch: Teil 351: Leittechnik
- DIN SPEC 91329** DIN SPEC 91329:2016.
Erweiterung des EPCIS-Ereignismodells um aggregierte Produktionsergebnisse zur Verwendung in betrieblichen Informationssystemen
- Dittrich & Fohlmeister 2020** Dittrich, Marc-André; Fohlmeister, Silas, 2020.
Cooperative multi-agent system for production control using reinforcement learning.
CIRP Annals - Manufacturing Technology **69** (1), S. 389–392.
DOI: 10.1016/j.cirp.2020.04.005
- Döbel et al. 2018** Döbel, Inga; Leis, Miriam; Vogelsang, Manuel Molina; Neustroev, Dmitry; Petzka, Henning; Riemer, Annamaria; Rüping, Stefan; Voss, Angelika; Wegele, Martin; Welz, Juliane, 2018.
Maschinelles Lernen: Eine Analyse zu Kompetenzen, Forschung und Anwendung. München: Fraunhofer-Gesellschaft
- Dyckhoff 2006** Dyckhoff, Harald, 2006.
Produktionstheorie: Grundzüge industrieller Produktionswirtschaft : mit 20 Tabellen. 5, überarb. Aufl. Berlin: Springer. Springer-Lehrbuch.
ISBN: 978-3-540-32600-7
- Echslar Minguillon & Lanza 2017** Echslar Minguillon, Fabio; Lanza, Gisela, 2017.
Maschinelles Lernen in der PPS: Konzept für den Einsatz von maschinellem Lernen für eine Matrix-Produktion.
wt Werkstattstechnik **107** (9), S. 630–634

- Eirinakis et al. 2021** Eirinakis, Pavlos; Kasapidis, Gregory; Mourtos, Ioannis; Repoussis, Panagiotis; Zampou, Eleni, 2021. Situation-aware manufacturing systems for capturing and handling disruptions. *Journal of Manufacturing Systems* **58**, S. 365–383. DOI: 10.1016/j.jmsy.2020.12.014
- Eiselt & Sandblom 2004** Eiselt, Horst A.; Sandblom, Carl-Louis, 2004. *Decision analysis, location models, and scheduling problems*. Berlin: Springer. ISBN: 978-3-540-40338-8
- ElMaraghy et al. 2012** ElMaraghy, Waguih; ElMaraghy, Hoda; Tomiyama, Tetsuo; Monostori, Laszlo, 2012. Complexity in engineering design and manufacturing. *CIRP Annals - Manufacturing Technology* **61** (2), S. 793–814. DOI: 10.1016/j.cirp.2012.05.001
- Espeholt et al. 2018** Espeholt, Lasse; Soyer, Hubert; Munos, Remi; Simonyan, Karen; Mnih, Volodymyr; Ward, Tom; Doron, Yotam; Firoiu, Vlad; Harley, Tim; Dunning, Iain; Legg, Shane; Kavukcuoglu, Koray, 2018. IMPALA: Scalable Distributed Deep-RL with Importance Weighted Actor-Learner Architectures. In: Dy, Jennifer; Krause, Andreas (Hrsg.): *Proceedings of the 35th International Conference on Machine Learning*. 35th International Conference on Machine Learning (ICML 2018). Stockholm, Schweden, 10.07.2018–15.07.2018, S. 1407–1416
- Etzion & Niblett 2011** Etzion, Opher; Niblett, Peter, 2011. *Event processing in action*. Greenwich, CT, USA: Manning. ISBN: 9781935182214
- Eversheim 1989** Eversheim, Walter, 1989. *Organisation in der Produktionstechnik: Band 4: Fertigung und Montage*. 2., neubearb. u. erw. Aufl. Berlin: Springer. VDI-Buch. ISBN: 978-3-642-64800-7
- Feigin et al. 1994** Feigin, Gerry; Fowler, John W.; Leachman, Robert, 1994. *Semiconductor Wafer Manufacturing Data Format Specification*. Austin, TX, USA: SEMATECH
- Fernández et al. 2016** Fernández, Erica; Bogado, Verónica; Salomone, Enrique; Chiotti, Omar J., 2016. Framework for modelling and simulating the supply process monitoring to detect and predict disruptive events. *Computers in Industry* **80**, S. 30–42. DOI: 10.1016/j.compind.2016.04.002

- Fertier et al. 2021** Fertier, Audrey; Martin, Guillaume; Barthe-Delanoë, Anne-Marie; Lesbegueries, Julien; Montarnal, Aurélie; Truptil, Sébastien; Bénaben, Frédéric; Salatgé, Nicolas, 2021. Managing events to improve situation awareness and resilience in a supply chain. *Computers in Industry* **132**, S. 103488. DOI: 10.1016/j.compind.2021.103488
- Fishman 2001** Fishman, George S., 2001. *Discrete-event simulation: Modeling, programming, and analysis*. New York, NY, USA: Springer. Springer series in operations research. ISBN: 978-0-387-95160-7
- Fleisch et al. 2005** Fleisch, Elgar; Christ, Oliver; Dierkes, Markus, 2005. Die betriebswirtschaftliche Vision des Internets der Dinge. In: Fleisch, Elgar; Mattern, Friedemann (Hrsg.): *Das Internet der Dinge*. Berlin: Springer, S. 3–37. ISBN: 978-3-540-24003-7
- Fleischmann et al. 2015** Fleischmann, Bernhard; Meyr, Herbert; Wagner, Michael, 2015. Advanced Planning. In: Stadler, Hartmut; Kilger, Christoph; Meyr, Herbert (Hrsg.): *Supply chain management and advanced planning*. Berlin: Springer, S. 71–95. ISBN: 978-3-642-55308-0
- Föllinger et al. 2016** Föllinger, Otto; Konigorski, Ulrich; Lohmann, Boris; Roppenecker, Günter; Trächtler, Ansgar, 2016. *Regelungstechnik: Einführung in die Methoden und ihre Anwendung*. 12., überarbeitete Auflage. Berlin: VDE Verlag. ISBN: 978-3-8007-4201-1
- Fowler & Robinson 1995** Fowler, John W.; Robinson, Jennifer, 1995. *Measurement and Improvement of Manufacturing Capacity (MIMAC): Final SEMATECH-Report 95062861A-TR*. Austin, TX, USA: SEMATECH
- Fowler et al. 2015** Fowler, John W.; Mönch, Lars; Ponsignon, Thomas, 2015. Discrete-event simulation for semiconductor wafer fabrication facilities: A tutorial. *International Journal of Industrial Engineering: Theory Applications and Practice* **22** (5), S. 661–682
- François-Lavet et al. 2018** François-Lavet, Vincent; Henderson, Peter; Islam, Riashat; Bellemare, Marc G.; Pineau, Joelle, 2018. An Introduction to Deep Reinforcement Learning. *FNT in Machine Learning (Foundations and Trends in Machine Learning)* **11** (3-4), S. 219–354. DOI: 10.1561/22000000071

- Friedli & Schuh 2012** Friedli, Thomas; Schuh, Günther, 2012.
Wettbewerbsfähigkeit der Produktion an Hochlohnstandorten.
 2. Aufl. Berlin: Springer.
 ISBN: 978-3-642-30275-6
- Gabel & Riedmiller 2007** Gabel, Thomas; Riedmiller, Martin, 2007.
 Adaptive Reactive Job-Shop Scheduling with Reinforcement
 Learning Agents.
*International Journal of Information Technology and Intelligent
 Computing* **4** (24), S. 1–30
- Gabel 2009** Gabel, Thomas, 2009.
*Multi-Agent Reinforcement Learning Approaches for
 Distributed Job-Shop Scheduling Problems*.
 Osnabrück, Univ., Diss., 2009.
 urn:nbn:de:gbv:700-2009081216
- Gabel & Riedmiller 2012** Gabel, Thomas; Riedmiller, Martin, 2012.
 Distributed policy search reinforcement learning for job-shop
 scheduling tasks.
International Journal of Production Research **50** (1), S. 41–61.
 DOI: 10.1080/00207543.2011.571443
- Genc et al. 2014** Genc, E.; Duffie, Neil; Reinhart, Gunther, 2014.
 Event-based Supply Chain Early Warning System for an
 Adaptive Production Control.
Procedia CIRP **19**, S. 39–44.
 DOI: 10.1016/j.procir.2014.04.076
- Genc & Lafortune 2009** Genc, Sahika; Lafortune, Stéphane, 2009.
 Predictability of event occurrences in partially-observed
 discrete-event systems.
Automatica **45** (2), S. 301–311.
 DOI: 10.1016/j.automatica.2008.06.022
- GHX LLC 2020** GHX LLC, 2020.
Supply chain event management.
 Patentschrift US 10,592,848 B2
- Giannakis & Louis 2016** Giannakis, Mihalís; Louis, Michalis, 2016.
 A multi-agent based system with big data processing for
 enhanced supply chain agility.
Journal of Enterprise Information Management **29** (5),
 S. 706–727.
 DOI: 10.1108/JEIM-06-2015-0050
- Goodfellow et al. 2016** Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron, 2016.
Deep learning. Cambridge, MA, USA: MIT Press.
 ISBN: 978-0262035613

Graesser & Keng 2019

Graesser, Laura Harding; Keng, Wah Loon, 2019.
Foundations of deep reinforcement learning: Theory and practice in Python. Boston, MA, USA: Addison-Wesley. Pearson
addison-wesley data & analytics series.
ISBN: 978-0-13-517238-4

Groche et al. 2018

Groche, Peter; Hoppe, Florian; Kessler, Thomas;
Kleemann, Annemie, 2018.
Industrial Working Environment 2025.
In: Liewald, Mathias (Hrsg.): *New developments in sheet metal forming - Conference proceedings of the International Conference on 'New Developments in Sheet Metal Forming'*.
International Conference on New Developments in Sheet Metal Forming.
Fellbach, 15.05.2018–16.05.2018, S. 125–136

Groover 2020

Groover, Mikell P., 2020.
Fundamentals of modern manufacturing: Materials, processes, and systems. Seventh edition. Hoboken, NJ, USA: John Wiley & Sons.
ISBN: 978-1-119-72201-4

Gross 2015

Gross, Richard, 2015.
Psychology: The Science of Mind and Behaviour. 7th ed.
London, England: Hodder Education.
ISBN: 978-1471-82973-4

Guarnaschelli et al. 2013

Guarnaschelli, Armando; Chiotti, Omar J.; Salomone, Hector E., 2013.
An approach based on constraint satisfaction problems to disruptive event management in supply chains.
International Journal of Production Economics **144** (1), S. 223–242.
DOI: 10.1016/j.ijpe.2013.02.007

Günther & Tempelmeier 2012

Günther, Hans-Otto; Tempelmeier, Horst, 2012.
Produktion und Logistik. 9., aktualisierte und erweiterte Auflage. Berlin: Springer.
ISBN: 9783642251641

Günthner 2008

Günthner, Willibald A., 2008.
Ereignisorientierte Logistik.
In: Nyhuis, Peter (Hrsg.): *Beiträge zu einer Theorie der Logistik*.
Berlin: Springer, S. 373–390.
ISBN: 978-3-540-75641-5

Gupta & Sivakumar 2006

Gupta, Amit Kumar; Sivakumar, Appa Iyer, 2006.
Job shop scheduling techniques in semiconductor manufacturing.
The International Journal of Advanced Manufacturing Technology **27** (11-12), S. 1163–1169.
DOI: 10.1007/s00170-004-2296-z

- Hackstein 1989** Hackstein, Rolf, 1989.
Produktionsplanung und -steuerung (PPS): Ein Handbuch für die Betriebspraxis. 2., überarb. Aufl. Düsseldorf: VDI-Verlag.
ISBN: 3-18-400924-6
- Hansmann 2014** Hansmann, Karl-Werner, 2014.
Industrielles Management. 8., völlig überarbeitete und erweiterte Aufl. München: De Gruyter Oldenbourg.
ISBN: 978-3-486-58058-7
- Hausknecht & Stone 2016** Hausknecht, Matthew; Stone, Peter, 2016.
Deep Reinforcement Learning in Parameterized Action Space.
In: Bengio, Yoshua; LeCun, Yann (Hrsg.): *Proceedings of the 4th International Conference on Learning Representations.*
4th International Conference on Learning Representations (ICLR 2016).
San Juan, Puerto Rico, 02.05.2016–04.05.2016
- Hedtstück 2017** Hedtstück, Ulrich, 2017.
Complex event processing: Verarbeitung von Ereignismustern in Datenströmen. Berlin: Springer Vieweg.
ISBN: 978-3-662-53451-9
- Heinecke 2013** Heinecke, Georg, 2013.
Resilient Automotive Production in Vulnerable Supply Networks: A Supply Chain Event Management System.
Zürich, Schweiz, ETH, Diss., 2013.
DOI: 10.3929/ethz-a-010049348
- Henderson et al. 2018** Henderson, Peter; Islam, Riashat; Bachman, Philip; Pineau, Joelle; Precup, Doina; Meger, David, 2018.
Deep Reinforcement Learning that Matters.
In: McIlraith, Sheila A.; Weinberger, Kilian Q. (Hrsg.): *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, the Thirtieth Innovative Applications of Artificial Intelligence Conference, and the Eighth Symposium on Educational Advances in Artificial Intelligence.*
New Orleans, LA, USA, 02.02.2018–07.02.2018,
S. 3207–3214.
ISBN: 978-1-57735-800-8

- Hessel et al. 2018**
Hessel, Matteo; Modayil, Joseph; van Hasselt, Hado; Schaul, Tom; Ostrovski, Georg; Dabney, Will; Horgan, Dan; Piot, Bilal; Azar, Mohammad; Silver, David, 2018.
Rainbow: Combining Improvements in Deep Reinforcement Learning.
In: McIlraith, Sheila A.; Weinberger, Kilian Q. (Hrsg.): *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, the Thirtieth Innovative Applications of Artificial Intelligence Conference, and the Eighth Symposium on Educational Advances in Artificial Intelligence*.
New Orleans, LA, USA, 02.02.2018–07.02.2018, S. 3215–3222.
ISBN: 978-1-57735-800-8
- Heusler et al. 2006**
Heusler, Klaus Felix; Stölzle, Wolfgang; Bachmann, Harald, 2006.
Supply Chain Event Management.
Wirtschaftswissenschaftliches Studium **35** (1), S. 19–24.
DOI: 10.15358/0340-1650-2006-1-19
- Hevner et al. 2004**
Hevner, Alan R; March, Salvatore T; Park, Jinsoo; Ram, Sudha, 2004.
Design Science in Information Systems Research.
MIS Quarterly **28** (1), S. 75–106
- Hevner 2007**
Hevner, Alan R, 2007.
A Three Cycle View of Design Science Research.
Scandinavian Journal of Information Systems **19** (2), S. 87–92
- Hintz et al. 2011**
Hintz, Greg; Kaza, Sri; Kwon, O. Sung; Leopoldseeder, Markus; Weig, Florian, 2011.
From source to drain: Fixing the supply chain.
McKinsey on Semiconductors (Autumn 2011), S. 42–49
- Hochreiter & Schmidhuber 1997**
Hochreiter, Sepp; Schmidhuber, Jürgen, 1997.
Long short-term memory.
Neural computation **9** (8), S. 1735–1780.
DOI: 10.1162/neco.1997.9.8.1735
- Hofmann et al. 2020**
Hofmann, Constantin; Krahe, Carmen; Stricker, Nicole; Lanza, Gisela, 2020.
Autonomous production control for matrix production based on deep Q-learning.
Procedia CIRP **88**, S. 25–30.
DOI: 10.1016/j.procir.2020.05.005
- Hopp 2008**
Hopp, Wallace J., 2008.
Supply chain science. Long Grove, IL, USA: Waveland Press.
ISBN: 978-1577667384

- Horgan et al. 2018** Horgan, Dan; Quan, John; Budden, David; Barth-Marón, Gabriel; Hessel, Matteo; van Hasselt, Hado; Silver, David, 2018.
Distributed Prioritized Experience Replay.
In: *Proceedings of the 6th International Conference on Learning Representations*.
6th International Conference on Learning Representations (ICLR 2018).
Vancouver, BC, Kanada, 30.04.2018–03.05.2018, S. 1–19
- Hu 2013** Hu, S. Jack, 2013.
Evolving Paradigms of Manufacturing: From Mass Production to Mass Customization and Personalization.
Procedia CIRP **7**, S. 3–8.
DOI: 10.1016/j.procir.2013.05.002
- Hu & Kaneko 2021** Hu, Zhejie; Kaneko, Tomoyuki, 2021.
Hierarchical Advantage for Reinforcement Learning in Parameterized Action Space.
In: *Proceedings of the 3rd IEEE Conference on Games*.
3rd IEEE Conference on Games.
Kopenhagen, Dänemark, 17.08.2021–20.08.2021
- Huber et al. 2021** Huber, Marco F.; Nagel, Tobias; Lamprecht, Raphael; Eiling, Florian, 2021.
Potenziale von Reinforcement Learning für die Produktion.
Industrie 4.0 Management **2021** (2), S. 25–29.
DOI: 10.30844/I40M_21-2_S25-29
- Ivanov & Dolgui 2020** Ivanov, Dmitry; Dolgui, Alexandre, 2020.
Viability of intertwined supply networks: extending the supply chain resilience angles towards survivability. A position paper motivated by COVID-19 outbreak.
International Journal of Production Research **58** (10), S. 2904–2915.
DOI: 10.1080/00207543.2020.1750727
- Jacod & Protter 2004** Jacod, Jean; Protter, Philip, 2004.
Probability Essentials. Second Edition. Berlin: Springer.
ISBN: 978-3-642-55682-1
- Jankowska et al. 2007** Jankowska, Anna Maria; Kurbel, Karl; Schreiber, Denny, 2007.
An architecture for agent-based mobile Supply Chain Event Management.
International Journal of Mobile Communications **5** (3), S. 243–258.
DOI: 10.1504/ijmc.2007.012393
- Jordan & Mitchell 2015** Jordan, Michael I.; Mitchell, Tom M., 2015.
Machine learning: Trends, perspectives, and prospects.
Science **349** (6245), S. 255–260.
DOI: 10.1126/science.aaa8415

- Kalinowski et al. 2014** Kalinowski, Krzysztof; Grabowik, Cezary; Kempa, Wojciech; Paprocka, Iwona, 2014.
The Procedure of Reaction to Unexpected Events in Scheduling of Manufacturing Systems with Discrete Production Flow.
Academy of Management Review **1036**, S. 840–845.
DOI: 10.4028/www.scientific.net/AMR.1036.840
- Kallrath & Maindl 2006** Kallrath, Josef; Maindl, Thomas I., 2006.
Real Optimization with SAP APO. Berlin: Springer.
ISBN: 978-3-540-22561-4
- Kardos et al. 2020** Kardos, Csaba; Laflamme, Catherine; Gallina, Viola; Sihn, Wilfried, 2020.
Dynamic scheduling in a job-shop production system with reinforcement learning.
Procedia CIRP **97**, S. 104–109.
DOI: 10.1016/j.procir.2020.05.210
- Kasakow et al. 2016** Kasakow, Georg; Menck, Nicole; Aurich, Jan C., 2016.
Event-driven Production Planning and Control Based on Individual Customer Orders.
Procedia CIRP **57**, S. 434–438.
DOI: 10.1016/j.procir.2016.11.075
- Kienzle 2012** Kienzle, Florian, 2012.
Fertigungssteuerung in der Musterfertigung von Systemlieferanten. Wissenschaftliche Schriftenreihe des Institutes für Betriebswissenschaften und Fabriksysteme 93.
Chemnitz, Techn. Univ., Diss., 2011.
urn:nbn:de:bsz:ch1-qucosa-81248
- Kiesewetter 1991** Kiesewetter, Stephen A., 1991.
Entwicklung einer dynamischen adaptierbaren Produktionsregelung: Dezentrale, hierarchische Regelung operativer Produktionsabläufe.
Aachen, RWTH, Diss., 1991
- Kletti 2015** Kletti, Jürgen (Hrsg.), 2015.
MES - Manufacturing Execution System: Moderne Informationstechnologie unterstützt die Wertschöpfung. 2. Auflage. Berlin: Springer Vieweg.
ISBN: 978-3-662-46901-9
- Knickle & Kemmeter 2002** Knickle, Kimberly; Kemmeter, J, 2002.
Supply chain event management in the field: Success with visibility. Boston, MA, USA: AMR Research
- Kopp et al. 2020** Kopp, Denny; Hassoun, Michael; Kalir, Adar; Monch, Lars, 2020.
SMT2020—A Semiconductor Manufacturing Testbed.
IEEE Transactions on Semiconductor Manufacturing **33** (4), S. 522–531.
DOI: 10.1109/TSM.2020.3001933

- Krcmar 2015** Krcmar, Helmut, 2015.
Informationsmanagement. 6., überarb. Aufl. Wiesbaden: Springer Gabler.
ISBN: 978-3-662-45863-1
- Kremers 2011** Kremers, Luc, 2011.
2010 State of Supply Chain performance in semiconductor industry. Singapur: iCognitive
- Kropp & Schuh 2014** Kropp, Sebastian; Schuh, Guenther, 2014.
Event-Driven Production Control based on Sensor Events.
In: IEEE (Hrsg.): *Proceedings of the 2014 European Conference on Smart Objects, Systems and Technologies*.
2014 European Conference on Smart Objects, Systems and Technologies (Smart SysTech 2014).
Dortmund, 01.07.2014–02.07.2014, S. 1–9.
ISBN: 978-3-8007-3626-3
- Kropp 2016** Kropp, Sebastian Kurt, 2016.
Entwicklung eines Ereignismodells als Grundlage der Produktionsregelung. Aachen: Apprimus-Verlag. Schriftenreihe Rationalisierung 137.
Aachen, RWTH, Diss., 2015.
ISBN: 978-3-86359-399-5
- Kubicek 1977** Kubicek, Herbert, 1977.
Heuristische Bezugsrahmen und heuristisch angelegte Forschungsdesigns als Elemente einer Konstruktionsstrategie empirischer Forschung.
In: Köhler, Richard (Hrsg.): *Empirische und handlungstheoretische Forschungskonzeptionen in der Betriebswirtschaftslehre*.
Stuttgart: Poeschel, S. 3–36.
ISBN: 3-7910-0214-7
- Kuhn & Hellingrath 2002** Kuhn, Axel; Hellingrath, Bernd, 2002.
Supply Chain Management: Optimierte Zusammenarbeit in der Wertschöpfungskette. Berlin: Springer.
ISBN: 978-3-540-65423-0
- Kuhn 2017** Kuhn, Thomas, 2017.
Digitaler Zwilling.
Informatik-Spektrum **40** (5), S. 440–444.
DOI: 10.1007/s00287-017-1061-2
- Kuhnle et al. 2019** Kuhnle, Andreas; Röhrig, Nicole; Lanza, Gisela, 2019.
Autonomous order dispatching in the semiconductor industry using reinforcement learning.
Procedia CIRP **79**, S. 391–396.
DOI: 10.1016/j.procir.2019.02.101

- Kuhnle 2020** Kuhnle, Andreas, 2020.
Adaptive Order Dispatching based on Reinforcement Learning: Application in a Complex Job Shop in the Semiconductor Industry. Düren: Shaker. Forschungsberichte aus dem wbk, Institut für Produktionstechnik Universität Karlsruhe 241. Karlsruhe, Karlsruher Institut für Technologie, Diss., 2020. ISBN: 978-3-8440-7762-9
- Kuhnle et al. 2021** Kuhnle, Andreas; Kaiser, Jan-Philipp; Theiß, Felix; Stricker, Nicole; Lanza, Gisela, 2021.
Designing an adaptive production control system using reinforcement learning.
Journal of Intelligent Manufacturing (32), S. 855–876.
DOI: 10.1007/s10845-020-01612-y
- Kurbel et al. 2009** Kurbel, Earl; Nowakowski, Kamil; Schreiber, Danny, 2009.
Enhancing Supply Chain Reliability through Agent-Based Supply Chain Event Management.
In: Kendall, Kenneth E.; Varshney, Upkar (Hrsg.): *Proceedings of the 15th Americas Conference on Information Systems*. 15th Americas Conference on Information Systems (AMCIS 2009). San Francisco, CA, USA, 06.08.2009–09.08.2009, S. 1–9. ISBN: 978-1-61567-581-4
- Lachman 1997** Lachman, Sheldon J., 1997.
Learning is a Process: Toward an Improved Definition of Learning.
The Journal of Psychology **131** (5), S. 477–480.
DOI: 10.1080/00223989709603535
- Lamarter et al. 2011** Lamarter, Steffen; Legat, Christoph; Lepratti, Raffaello; Scharnagl, Joachim; Jordan, Lars, 2011.
Event-based Reactive Production Order Scheduling for Manufacturing Execution Systems.
IFAC Proceedings **44** (1), S. 2722–2730.
DOI: 10.3182/20110828-6-IT-1002.02054
- Lamprecht et al. 2021** Lamprecht, Raphael; Wurst, Ferdinand; Huber, Marco F., 2021.
Reinforcement Learning based Condition-oriented Maintenance Scheduling for Flow Line Systems.
In: *Proceedings of the 2021 IEEE 19th International Conference on Industrial Informatics*. IEEE 19th International Conference on Industrial Informatics (INDIN 2021). Palma de Mallorca, Spanien, 21.07.2021–23.07.2021, S. 1–7. ISBN: 978-1-7281-4395-8

- Le Roux et al. 2012** Le Roux, Nicolas; Bengio, Yoshua; Fitzgibbon, Andrew, 2012. Improving First and Second-Order Methods by Modeling Uncertainty.
In: Sra, Suvrit; Nowozin, Sebastian; Wright, Stephen J. (Hrsg.): *Optimization for machine learning*. Cambridge, MA, USA: MIT Press, S. 403–430. ISBN: 9780262016469
- LeCun et al. 2015** LeCun, Yann; Bengio, Yoshua; Hinton, Geoffrey, 2015. Deep learning.
Nature **521** (7553), S. 436–444.
DOI: 10.1038/nature14539
- Lee 2004** Lee, Hau L., 2004.
The Triple-A Supply Chain.
Harvard Business Review **82** (10), S. 102–113
- Lee et al. 2019** Lee, Won-Jun; Kim, Byung-Hee; Ko, Keyhoon; Shin, Hayong, 2019.
Simulation Based Multi-Objective Fab Scheduling by Using Reinforcement Learning.
In: IEEE (Hrsg.): *Proceedings of the 2019 Winter Simulation Conference*.
2019 Winter Simulation Conference (WSC 2019). National Harbor, MD, USA, 08.12.2019–11.12.2019, S. 2236–2247.
ISBN: 978-1-7281-3283-9
- Lepratti et al. 2012** Lepratti, Raffaello; Lamparter, Steffen; Heinecke, Georg; Scharnagl, Joachim; Hansen, Ralf, 2012.
In Echtzeit: RFID-gestütztes Produktionsassistenzsystem: Robuste Produktion durch bessere Lieferkettentransparenz.
atp edition (11), S. 36–44
- Liang et al. 2018** Liang, Eric; Liaw, Richard; Nishihara, Robert; Moritz, Philipp; Fox, Roy; Goldberg, Ken; Gonzalez, Joseph; Jordan, Michael; Stoica, Ion, 2018.
RLlib: Abstractions for Distributed Reinforcement Learning.
Proceedings of Machine Learning Research (80), S. 1–10
- Little 1961** Little, John D. C., 1961.
A Proof for the Queuing Formula.
Operations Research **9** (3), S. 383–387
- Liu et al. 2020** Liu, Chien-Liang; Chang, Chuan-Chin; Tseng, Chun-Jan, 2020.
Actor-Critic Deep Reinforcement Learning for Solving Job Shop Scheduling Problems.
IEEE Access **8**, S. 71752–71762.
DOI: 10.1109/ACCESS.2020.2987820

Lödding 2001

Lödding, Hermann, 2001.
Dezentrale bestandsorientierte Fertigungsregelung. Düsseldorf: VDI-Verlag. Fortschritt-Berichte VDI Reihe 2, Fertigungstechnik 587.
Hannover, Univ., Diss., 2001.
ISBN: 3-18-358702-5

Lödding 2016

Lödding, Hermann, 2016.
Verfahren der Fertigungssteuerung: Grundlagen, Beschreibung, Konfiguration. 3. Auflage. Berlin: Springer Vieweg. VDI-Buch.
ISBN: 978-3-662-48458-6

Lödding 2020

Lödding, Hermann (Hrsg.), 2020.
PPS-Report 2019: Studienergebnisse. Garbsen: TEWISS.
ISBN: 978-3-95900-402-2

Lohmer & Lasch 2021

Lohmer, Jacob; Lasch, Rainer, 2021.
Production planning and scheduling in multi-factory production networks: a systematic literature review.
International Journal of Production Research **59** (7), S. 2028–2054.
DOI: 10.1080/00207543.2020.1797207

Loos 1999

Loos, Peter, 1999.
Grunddatenverwaltung und Betriebsdatenerfassung als Basis der Produktionsplanung und -steuerung.
In: Corsten, Hans (Hrsg.): *Einführung in das Produktionscontrolling*. München: Vahlen, S. 227–252.
ISBN: 3800623226

Luckham 2010

Luckham, David C., 2010.
The power of events: An introduction to complex event processing in distributed enterprise systems. 6th edition. Boston, MA, USA: Addison-Wesley.
ISBN: 9780201727890

Luczak & Eversheim 1999

Luczak, Holger; Eversheim, Walter (Hrsg.), 1999.
Produktionsplanung und -steuerung: Grundlagen, Gestaltung und Konzepte. 2., korrigierte Auflage 1999. Berlin: Springer.
ISBN: 978-3-662-09472-3

Luhmann 2015

Luhmann, Niklas, 2015.
Soziale Systeme: Grundriß einer allgemeinen Theorie. 16. Auflage. Frankfurt am Main: Suhrkamp.
Suhrkamp-Taschenbuch Wissenschaft 666.
ISBN: 3-518-28266-2

Luo 2020

Luo, Shu, 2020.
Dynamic scheduling for flexible job shop with new job insertions by deep reinforcement learning.
Applied Soft Computing **91**, S. 106208.
DOI: 10.1016/j.asoc.2020.106208

- Macchion et al. 2020** Macchion, Laura; Moretto, Antonella; Caniato, Federico; Danese, Pamela; Vinelli, Andrea, 2020. Static supply chain complexity and sustainability practices: A multitier examination. *Corporate Social Responsibility and Environmental Management* **27** (6), S. 2679–2691. DOI: 10.1002/csr.1992
- Mack & Khare 2016** Mack, Oliver; Khare, Anshuman, 2016. Perspectives on a VUCA World. In: Mack, Oliver; Khare, Anshuman; Krämer, Andreas; Burgartz, Thomas (Hrsg.): *Managing in a VUCA World*. Cham: Springer, S. 3–19. ISBN: 978-3-319-16888-3
- Malone & Crowston 1994** Malone, Thomas W.; Crowston, Kevin, 1994. The interdisciplinary study of coordination. *ACM Computing Surveys* **26** (1), S. 87–119. DOI: 10.1145/174666.174668
- Mason et al. 2002** Mason, Scott J.; Fowler, John W.; Matthew Carlyle, W., 2002. A modified shifting bottleneck heuristic for minimizing total weighted tardiness in complex job shops. *Journal of Scheduling* **5** (3), S. 247–262. DOI: 10.1002/jos.102
- May et al. 2021** May, Marvin Carl; Kiefer, Lars; Kuhnle, Andreas; Stricker, Nicole; Lanza, Gisela, 2021. Decentralized Multi-Agent Production Control through Economic Model Bidding for Matrix Production Systems. *Procedia CIRP* **96**, S. 3–8. DOI: 10.1016/j.procir.2021.01.043
- Metz 2014** Metz, Daniel, 2014. *The Concept of a Real-Time Enterprise in Manufacturing*. Wiesbaden: Springer Gabler. Siegen, Univ., Diss., 2013. ISBN: 978-3-658-03749-9
- Minsky 1961** Minsky, Marvin, 1961. Steps toward Artificial Intelligence. *Proceedings of the IRE* **49** (1), S. 8–30. DOI: 10.1109/JRPROC.1961.287775
- Mnih et al. 2013** Mnih, Volodymyr; Kavukcuoglu, Koray; Silver, David; Graves, Alex; Antonoglou, Ioannis; Wierstra, Daan; Riedmiller, Martin, 2013. Playing Atari with Deep Reinforcement Learning. *Computing Research Repository*, S. 1–9

- Mnih et al. 2016** Mnih, Volodymyr; Badia, Adria Puigdomenech; Mirza, Mehdi; Graves, Alex; Lillicrap, Timothy P.; Harley, Tim; Silver, David; Kavukcuoglu, Koray, 2016.
Asynchronous Methods for Deep Reinforcement Learning.
In: Balcan, Maria Florina; Weinberger, Kilian Q. (Hrsg.): *Proceedings of The 33rd International Conference on Machine Learning*.
33rd International Conference on Machine Learning (ICML 2016).
New York, NY, USA, 20.06.2016–22.06.2016, S. 1928–1937.
ISBN: 978-1-5108-2900-8
- Möller 2011** Möller, Klaus, 2011.
Innovationscontrolling. Stuttgart: Schäffer-Poeschel.
ISBN: 978-3-7910-3089-0
- Mönch et al. 2003** Mönch, Lars; Rose, Oliver; Sturm, Roland, 2003.
A Simulation Framework for the Performance Assessment of Shop-Floor Control Systems.
SIMULATION **79** (3), S. 163–170.
DOI: 10.1177/0037549703256039
- Mönch 2006** Mönch, Lars, 2006.
Agentenbasierte Produktionssteuerung komplexer Produktionssysteme. Wiesbaden: Deutscher Universitätsverlag.
Ilmenau, Techn. Univ., Diss., 2005.
ISBN: 978-3-8350-0249-4
- Mönch et al. 2011** Mönch, Lars; Fowler, John W.; Dauzère-Pérès, Stéphane; Mason, Scott J.; Rose, Oliver, 2011.
A survey of problems, solution techniques, and future challenges in scheduling semiconductor manufacturing operations.
Journal of Scheduling **14** (6), S. 583–599.
DOI: 10.1007/s10951-010-0222-9
- Mönch et al. 2013** Mönch, Lars; Fowler, John W.; Mason, Scott J., 2013.
Production planning and control for semiconductor wafer fabrication facilities: Modeling, analysis, and systems. New York, NY, USA: Springer. Operations Research / Computer Science Interfaces Series 52.
ISBN: 9781461444718
- Mönch et al. 2018** Mönch, Lars; Uzsoy, Reha; Fowler, John W., 2018.
A survey of semiconductor supply chain models part I: semiconductor supply chains, strategic network design, and supply chain simulation.
International Journal of Production Research **56** (13), S. 4524–4545.
DOI: 10.1080/00207543.2017.1401233

- Müller et al. 2020** Müller, Klaus; Vignaux, Tony; Scherfke, Stefan; Lünsdorf, Ontje, 2020.
SimPy - Discrete event simulation for Python.
 URL: <https://simpy.readthedocs.io/en/4.0.1/>
 Zugriff am: 26.08.2021
- Nachum et al. 2018** Nachum, Ofir; Gu, Shixiang; Lee, Honglak; Levine, Sergey, 2018.
 Data-Efficient Hierarchical Reinforcement Learning.
 In: Bengio, Samy; Wallach, Hanna M.; Larochelle, Hugo; Grauman, Kirsten; Cesa-Bianchi, Nicolò; Garnett, Roman (Hrsg.): *Advances in Neural Information Processing Systems 31*.
 Conference on Neural Information Processing Systems (NeurIPS 2018).
 Montréal, QC, Kanada, 02.12.2018–08.12.2018, S. 1–11
- Naik et al. 2019** Naik, Abhishek; Shariff, Roshan; Yasuri, Niko; Sutton, Richard S., 2019.
 Discounted Reinforcement Learning is Not an Optimization Problem.
 In: Wallach, Hanna M.; Larochelle, Hugo; Beygelzimer, Alina; d'Alché-Buc, Florence; Fox, Emily B.; Garnett, Roman (Hrsg.): *Advances in Neural Information Processing Systems 32*.
 Conference on Neural Information Processing Systems (NeurIPS 2019).
 Vancouver, Kanada, 08.12.2019–14.12.2019, S. 1–7
- Nakagawa et al. 2011** Nakagawa, Elisa Yumi; Oliveira Antonino, Pablo; Becker, Martin, 2011.
 Reference Architecture and Product Line Architecture: A Subtle But Critical Difference.
 In: Crnkovic, Ivica; Gruhn, Volker; Book, Matthias (Hrsg.): *Software Architecture*.
 Berlin: Springer, S. 207–211.
 ISBN: 978-3-642-23797-3
- Negahban & Smith 2014** Negahban, Ashkan; Smith, Jeffrey S., 2014.
 Simulation for manufacturing system design and operation: Literature review and analysis.
Journal of Manufacturing Systems **33** (2), S. 241–261.
 DOI: 10.1016/j.jmsy.2013.12.007
- Neugebauer 2018** Neugebauer, Reimund (Hrsg.), 2018.
Digitalisierung: Schlüsseltechnologien für Wirtschaft und Gesellschaft. Berlin: Springer Vieweg.
 ISBN: 978-3-662-55889-8

- Niehues 2017** Niehues, Michael Richard, 2017.
Adaptive Produktionssteuerung für Werkstattfertigungssysteme durch fertigungsbegleitende Reihenfolgebildung. München: utzverlag. Forschungsberichte iwb 329.
München, Techn. Univ., Diss., 2016.
ISBN: 978-3-8316-4650-0
- Niehues et al. 2018** Niehues, Michael Richard; Blum, Matthias;
Teschemacher, Ulrich; Reinhart, Gunther, 2018.
Adaptive job shop control based on permanent order sequencing.
Production Engineering **12** (1), S. 65–71.
DOI: 10.1007/s11740-017-0780-1
- Nissen 2002** Nissen, Volker, 2002.
Supply Chain Event Management.
Wirtschaftsinformatik **44** (5), S. 477–480.
DOI: 10.1007/BF03250869
- Nyhuis et al. 2009** Nyhuis, Peter; Hartmann, Wiebke; Kennemann, Marco;
Münzberg, Ben, 2009.
Produktionsregelung mit logistischen Kennlinien:
Fertigungssteuerung auf Basis einer kaskadierten Regelung.
wt Werkstattstechnik **99** (4), S. 239–243
- Nyhuis & Wiendahl 2012** Nyhuis, Peter; Wiendahl, Hans-Peter, 2012.
Logistische Kennlinien: Grundlagen, Werkzeuge und Anwendungen. 3. Aufl. Berlin: Springer. VDI-Buch.
ISBN: 978-3-540-92838-6
- Nyhuis et al. 2016** Nyhuis, Peter; Mayer, Jonas; Pielmeier, Julia; Berger, Christoph;
Engehausen, Friederike; Hempel, Thomas; Hünnekes, Philipp,
2016.
Aktuellen Herausforderungen der Produktionsplanung und -steuerung mittels Industrie 4.0 begegnen: Studienergebnisse.
Garbsen: PZH Verlag.
ISBN: 978-3-95900-104-5
- OECD 2002** OECD, 2002.
Frascati Manual 2002: Proposed Standard Practice for Surveys on Research and Experimental Development. Paris, Frankreich: OECD Publishing.
ISBN: 9789264199033

- Oren et al. 2021** Oren, Joel; Ross, Chana; Lefarov, Maksym; Richter, Felix; Taitler, Ayal; Feldman, Zohar; Di Castro, Dotan; Daniel, Christian, 2021.
SOLO: Search Online, Learn Offline for Combinatorial Optimization Problems.
In: *Proceedings of the Fourteenth International Symposium on Combinatorial Search*.
14th International Symposium on Combinatorial Search. Guangzhou, China, 26.07.2021–30.07.2021, S. 97–105.
ISBN: 978-1-57735-870-1
- Ota et al. 2021** Ota, Kei; Jha, Devesh K.; Kanezaki, Asako, 2021.
Training Larger Networks for Deep Reinforcement Learning.
Computing Research Repository, S. 1–16.
arXiv: 2102.07920v1
- Otto 2003** Otto, Andreas, 2003.
Supply Chain Event Management: Three Perspectives.
The International Journal of Logistics Management **14** (2), S. 1–13.
DOI: 10.1108/09574090310806567
- Palma et al. 2019a** Palma, Fanny; Saucedo, Jania A.; Marmolejo, José A., 2019.
Recovery Method of Supply Chain Under Ripple Effect: Supply Chain Event Management (SCEM) Application.
In: Vasant, Pandian; Zelinka, Ivan; Weber, Gerhard-Wilhelm (Hrsg.): *Intelligent Computing & Optimization*.
Cham: Springer International Publishing, S. 455–465.
ISBN: 978-3-030-00978-6
- Palma et al. 2019b** Palma, Fanny; Saucedo, Jania A.; Marmolejo-Saucedo, J. A.; Rodriguez-Aguilar, Roman, 2019.
The supply chain event management application: a case study.
IFAC-PapersOnLine **52** (13), S. 2698–2703.
DOI: 10.1016/j.ifacol.2019.11.615
- Panten 2019** Panten, Niklas, 2019.
Deep Reinforcement Learning zur Betriebsoptimierung hybrider industrieller Energienetze. Aachen: Shaker. Schriftenreihe des PTW: Innovation Fertigungstechnik.
Darmstadt, Techn. Univ., Diss., 2019.
ISBN: 978-3-8440-7036-1
- Papula 2016** Papula, Lothar, 2016.
Mathematik für Ingenieure und Naturwissenschaftler: Band 3 : Vektoranalysis, Wahrscheinlichkeitsrechnung, Mathematische Statistik, Fehler- und Ausgleichsrechnung. 7. Auflage.
Wiesbaden: Springer Vieweg.
ISBN: 978-3-658-11923-2

- Park et al. 2020** Park, In-Beom; Huh, Jaeseok; Kim, Joongkyun; Park, Jonghun, 2020.
A Reinforcement Learning Approach to Robust Scheduling of Semiconductor Manufacturing Facilities.
IEEE Transactions on Automation Science and Engineering **17** (3), S. 1420–1431.
DOI: 10.1109/TASE.2019.2956762
- Petermann 1996** Petermann, Dirk, 1996.
Modellbasierte Produktionsregelung. Düsseldorf: VDI-Verlag. Fortschritt-Berichte VDI Reihe 20, Rechnerunterstützte Verfahren 193.
Hannover, Univ., Diss., 1995.
ISBN: 3183193205
- Pfitzer et al. 2018** Pfitzer, Florian; Provost, Julien; Mieth, Carina; Liertz, Wolfgang, 2018.
Event-Driven Production Rescheduling in Job Shop Environments.
In: IEEE (Hrsg.): *Proceedings of the 14th International Conference on Automation Science and Engineering*.
14th IEEE International Conference on Automation Science and Engineering (CASE 2018).
München, 20.08.2018–24.08.2018, S. 939–944.
ISBN: 978-1-5386-3593-3
- Pfund et al. 2006** Pfund, Michele E; Scott, Mason J; Fowler, John W., 2006.
Semiconductor Manufacturing and Scheduling.
In: Herrmann, Jeffrey W. (Hrsg.): *Handbook of Production Scheduling*.
Boston, MA, USA: Springer, S. 213–241.
ISBN: 9780387331157
- Pfund & Fowler 2017** Pfund, Michele E.; Fowler, John W., 2017.
Extending the boundaries between scheduling and dispatching: hedging and rescheduling techniques.
International Journal of Production Research **55** (11), S. 3294–3307.
DOI: 10.1080/00207543.2017.1306133
- Pielmeier et al. 2017** Pielmeier, Julia; Braunreuther, Stefan; Reinhart, Gunther, 2017.
Modeling Approach for Situational Event-handling within Production Planning and Control Based on Complex Event Processing.
Procedia CIRP **63**, S. 271–276.
DOI: 10.1016/j.procir.2017.03.158

- Pielmeier et al. 2019** Pielmeier, Julia; Theumer, Philipp; Schutte, Corné S.L.; Snyman, Stephan; Bessdo, Olaf; Braunreuther, Stefan; Reinhart, Gunther, 2019.
Method for event-based production control.
Procedia CIRP **79**, S. 373–378.
DOI: 10.1016/j.procir.2019.02.094
- Pielmeier 2020** Pielmeier, Julia, 2020.
System zur ereignisorientierten Produktionssteuerung.
München: utzverlag. Forschungsberichte iwb 356.
München, Techn. Univ., Diss., 2019.
ISBN: 978-3-8316-4856-6
- Pinedo 2016** Pinedo, Michael L., 2016.
Scheduling: Theory, Algorithms, and Systems. 5th ed. Cham: Springer.
ISBN: 978-3-319-26578-0
- Ponomarov & Holcomb 2009** Ponomarov, Serhiy Y.; Holcomb, Mary C., 2009.
Understanding the concept of supply chain resilience.
The International Journal of Logistics Management **20** (1), S. 124–143.
DOI: 10.1108/09574090910954873
- Poole & Mackworth 2017** Poole, David L.; Mackworth, Alan K., 2017.
Artificial intelligence: Foundations of computational agents.
2nd edition. Cambridge, MA, USA: Cambridge University Press.
ISBN: 9781107195394
- Porter 1985** Porter, Michael E., 1985.
Competitive advantage: Creating and sustaining superior performance. New York, NY, USA: Free Press.
ISBN: 0-684-84146-0
- Pritschow & Wiendahl 1995** Pritschow, Günter; Wiendahl, Hans-Peter, 1995.
Application of Control Theory for Production Logistics – Results of a Joint Project.
CIRP Annals - Manufacturing Technology **44** (1), S. 421–424.
DOI: 10.1016/S0007-8506(07)62355-5
- Rammer 2020** Rammer, Christian, 2020.
Auf Künstliche Intelligenz kommt es an: Beitrag von KI zur Innovationsleistung und Performance der deutschen Wirtschaft.
Berlin: Bundesministerium für Wirtschaft und Energie.
URL: https://www.de.digital/DIGITAL/Redaktion/DE/Digitalisierungsindex/Publikationen/publikation-download-kuenstliche-intelligenz-2.pdf?__blob=publicationFile&v=3
Zugriff am: 26.05.2021

Raschka et al. 2020

Raschka, Sebastian; Patterson, Joshua; Nolet, Corey, 2020. Machine Learning in Python: Main Developments and Technology Trends in Data Science, Machine Learning, and Artificial Intelligence. *Information* **11** (4), S. 1–44. DOI: 10.3390/info11040193

Reinhart & Gyger 2008

Reinhart, Gunther; Gyger, Tobias, 2008. Identification of implicit strategies in production control. In: IEEE (Hrsg.): *Proceedings of the 2008 IEEE International Conference on Industrial Engineering and Engineering Management*. 2008 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM 2008). Singapur, 08.12.2008–11.12.2008, S. 302–306. ISBN: 978-1-4244-2629-4

Reinhart et al. 2013

Reinhart, Gunther; Niehues, Michael Richard; Batzner, Joachim; Welter, E., 2013. Werkstattsteuerung mittels Ressourcenkonten: Hochvariante Produkte mittels bestandsregelndem Verfahren effizient und engpassarm fertigen. *wt Werkstattstechnik* **103** (4), S. 331–335

Reinhart 2017

Reinhart, Gunther (Hrsg.), 2017. *Handbuch Industrie 4.0: Geschäftsmodelle, Prozesse, Technik*. München: Carl Hanser. ISBN: 978-3-446-44642-7

Reinhart & Zühlke 2017

Reinhart, Gunther; Zühlke, Detlef, 2017. Von CIM zu Industrie 4.0. In: Reinhart, Gunther (Hrsg.): *Handbuch Industrie 4.0*. München: Carl Hanser, S. XXXI–XL. ISBN: 978-3-446-44642-7

Richter 2019

Richter, Stefan, 2019. *Statistisches und maschinelles Lernen: Gängige Verfahren im Überblick*. Berlin: Springer Spektrum. ISBN: 978-3-662-59353-0

Riegelmayr 2020

Riegelmayr, Wolfgang P., 2020. *Industrie 4.0 - Vernetzungen für die digitale Fabrik: Leitungstechnik, Schnittstellen, Leistungsmerkmale, Gestaltungs- und Auslegungsprinzipien*. München: Carl Hanser. ISBN: 978-3-446-46147-5

- Rinciog et al. 2020** Rinciog, Alexandru; Mieth, Carina; Scheikl, Paul Maria; Meyer, Anney, 2020.
Sheet-Metal Production Scheduling Using AlphaGo Zero.
In: Nyhuis, Peter; Herberger, David; Hübner, Marco (Hrsg.):
Proceedings of the 1st Conference on Production Systems and Logistics.
1st Conference on Production Systems and Logistics (CPSL 2020).
Stellenbosch, Südafrika, 17.03.2020–20.03.2020, S. 342–352.
DOI: 10.15488/9676
- Roberts 2018** Roberts, Mike, 2018.
Designing even-driven systems: Concepts and patterns for streaming services with Apache Kafka. Sebastopol, CA, USA: O'Reilly Media.
ISBN: 9781492038245
- Robinson 2004** Robinson, Stewart, 2004.
Simulation: The practice of model development and use. Chichester, England: John Wiley & Sons.
ISBN: 0-470-84772-7
- Ropohl 2009** Ropohl, Günter, 2009.
Signatures der technischen Welt: Neue Beiträge zur Technikphilosophie. Berlin: Lit. Technikphilosophie 20.
ISBN: 978-3-643-10465-6
- Rösch 2021** Rösch, Martin Walter, 2021.
System zur energieorientierten und kostenbasierten Produktionssteuerung mittels Reinforcement Learning. München, Techn. Univ., Diss., 2021.
urn:nbn:de:bvb:91-diss-20211004-1597405-1-3
- Russell & Norvig 2021** Russell, Stuart J.; Norvig, Peter, 2021.
Artificial intelligence: A modern approach. 4th edition. Hoboken, NJ, USA: Pearson.
ISBN: 978-0-13-461099-3
- Rüttgers et al. 2018** Rüttgers, Jürgen; Bonefeld-Dahl, Cecilia; Herlitschka, Sabine; van Saarloos, Wim; Stigenberg, Anna Hultin; Nolan, Alistair; Rosser, Susan; San Sebastian, Inaki; Ségol, Bernadette; Stejskalová, Eva; Thienpont, Hugo; Tolio, Tullio; Vasara, Antti; Vazquez, Adiari, 2018.
Re-finding Industry: Defining Innovation. Brüssel, Belgien: Europäische Kommission.
ISBN: 978-92-79-85272-5
- Sarin et al. 2011** Sarin, Subhash C.; Varadarajan, Amrusha; Wang, Lixin, 2011.
A survey of dispatching rules for operational control in wafer fabrication.
Production Planning & Control **22** (1), S. 4–24.
DOI: 10.1080/09537287.2010.490014

- Sauer 2018** Sauer, Alexander, 2018.
Energieeffizient und flexibel: Die Fabrik der Zukunft.
Stuttgart: Effizienz-Gipfel 2018, 14.05.2018.
URL: <https://www.eep.uni-stuttgart.de/dokumente/Gipfel-2018/Sauer-Vortrag-2018-Gipfel.pdf>
Zugriff am: 10.11.2021
- Schanz 1987** Schanz, Günther, 1987.
Wissenschaftstheoretische Grundfragen der
Führungsforschung.
In: Kieser, Alfred; Reber, Gerhard; Wunderer, Rolf (Hrsg.):
Handwörterbuch der Führung.
Stuttgart: Schäffer-Poeschel, S. 2039–2047.
ISBN: 9783791080284
- Schanz 1988** Schanz, Günther, 1988.
Methodologie für Betriebswirte. 2., überarb. u. erw. Aufl.
Stuttgart: Poeschel. Sammlung Poeschel P 132.
ISBN: 3-7910-9161-1
- Schel et al. 2018** Schel, Daniel; Henkel, Christian; Stock, Daniel; Meyer, Olga;
Rauhöft, Greg; Einberger, Peter; Stöhr, Matthias;
Daxer, Marc Andre; Seidelmann, Joachim, 2018.
Manufacturing Service Bus: An Implementation.
Procedia CIRP **67**, S. 179–184.
DOI: 10.1016/j.procir.2017.12.196
- Schenk et al. 2012** Schenk, Michael; Schürmeyer, Maik; Bauhoff, Fabian, 2012.
Koordination interner Produktionsnetzwerke.
In: Schuh, Günther; Stich, Volker (Hrsg.): *Produktionsplanung
und -steuerung 1*.
Berlin: Springer Vieweg, S. 426–470.
ISBN: 978-3-642-25422-2
- Schiffer et al. 2020** Schiffer, Martina; Wiendahl, Hans-Hermann; Saretz, Benedikt;
Lickefett, Michael; Pietrzak, Georg; Forstmann, Bodo, 2020.
SCM2040 – Wie verändert sich die Logistik in der Zukunft?
Stuttgart: Fraunhofer IPA.
URL: [https://www.ipa.fraunhofer.de/de/Publikationen/studien/
supply_chain_management_2040.html](https://www.ipa.fraunhofer.de/de/Publikationen/studien/supply_chain_management_2040.html)
Zugriff am: 23.11.2021
- Scholz-Reiter et al. 2011** Scholz-Reiter, Bernd; Windt, Katja; Liu, Huaxin, 2011.
A multiple-logistic-objective-optimized manufacturing planning
and control system.
*Proceedings of the Institution of Mechanical Engineers, Part B:
Journal of Engineering Manufacture* **225** (4), S. 599–610.
DOI: 10.1177/2041297510394108
- Schönsleben 2016** Schönsleben, Paul, 2016.
Integrales Logistikmanagement. Berlin: Springer.
ISBN: 978-3-662-48333-6

- Schrittwieser et al. 2020** Schrittwieser, Julian; Antonoglou, Ioannis; Hubert, Thomas; Simonyan, Karen; Sifre, Laurent; Schmitt, Simon; Guez, Arthur; Lockhart, Edward; Hassabis, Demis; Graepel, Thore; Lillicrap, Timothy P.; Silver, David, 2020. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature* **588** (7839), S. 604–609. DOI: 10.1038/s41586-020-03051-4
- Schuh et al. 2011a** Schuh, Günther; Lödding, Hermann; Stich, Volker; Reuter, Christian; Schmidt, Oliver; Potente, Till; Franzkoch, Bastian; Brosze, Tobias; Thomas, Christina; Wesch-Potente, Cathrin, 2011. High Resolution Production Management. In: Brecher, Christian; Klocke, Fritz (Hrsg.): *Wettbewerbsfaktor Produktionstechnik*. Aachen: Shaker, S. 61–80. ISBN: 978-3-8440-0087-0
- Schuh et al. 2011b** Schuh, Günther; Kampker, Achim; Rittstieg, Moritz, 2011. Vernetzte Wertschöpfung und Kooperationsmanagement. In: Schuh, Günther; Kampker, Achim (Hrsg.): *Strategie und Management produzierender Unternehmen*. Berlin, Heidelberg: Springer, S. 463–535. ISBN: 978-3-642-14501-8
- Schuh et al. 2012a** Schuh, Günther; Brosze, Tobias; Brandenburg, Ulrich, 2012. Aachener PPS-Modell. In: Schuh, Günther; Stich, Volker (Hrsg.): *Produktionsplanung und -steuerung 1*. Berlin: Springer Vieweg, S. 11–28. ISBN: 978-3-642-25422-2
- Schuh et al. 2012b** Schuh, Günther; Brandenburg, Ulrich; Cuber, Stefan, 2012. Aufgaben. In: Schuh, Günther; Stich, Volker (Hrsg.): *Produktionsplanung und -steuerung 1*. Berlin: Springer Vieweg, S. 29–81. ISBN: 978-3-642-25422-2
- Schuh et al. 2012c** Schuh, Günther; Schmidt, Carsten; Helmig, Jan, 2012. Prozesse. In: Schuh, Günther; Stich, Volker (Hrsg.): *Produktionsplanung und -steuerung 1*. Berlin: Springer Vieweg, S. 109–194. ISBN: 978-3-642-25422-2

- Schuh et al. 2013** Schuh, Günther; Potente, Till; Thomas, Christina; Hauptvogel, Annika, 2013. Cyber-Physical Production Management. In: Prabhu, Vittal (Hrsg.): *Advances in Production Management Systems. Sustainable Production and Service Supply Chains*. Berlin: Springer, S. 477–484. ISBN: 978-3-642-41262-2
- Schuh & Schmidt 2014** Schuh, Günther; Schmidt, Carsten, 2014. *Produktionsmanagement: Handbuch Produktion und Management 5*. 2. Aufl. Berlin: Springer Vieweg. VDI-Buch. ISBN: 978-3-642-54287-9
- Schuh et al. 2015** Schuh, Günther; Stich, Volker; Hocken, Christian; Schenk, Michael, 2015. Design of an Integrated Model for the Real-Time Disturbance Management in Transportation Supply Networks. In: Umeda, Shigeki; Nakano, Masaru; Mizuyama, Hajime; Hibino, Nironori; Kiritsis, Dimitris; Cieminski, Gregor von (Hrsg.): *Advances in Production Management Systems: Innovative Production Management Towards Sustainable Growth*. Cham: Springer International Publishing, S. 144–151. ISBN: 978-3-319-22755-9
- Schuh & Riesener 2017** Schuh, Günther; Riesener, Michael, 2017. *Produktkomplexität managen: Strategien - Methoden - Tools*. 3., vollständig überarbeitete Auflage. München: Carl Hanser. ISBN: 978-3-446-45225-1
- Schulman et al. 2016** Schulman, John; Moritz, Philipp; Levine, Sergey; Jordan, Michael I.; Abbeel, Pieter, 2016. High-Dimensional Continuous Control Using Generalized Advantage Estimation. In: Bengio, Yoshua; LeCun, Yann (Hrsg.): *Proceedings of the 4th International Conference on Learning Representations*. 4th International Conference on Learning Representations (ICLR 2016). San Juan, Puerto Rico, 02.05.2016–04.05.2016
- Schulman et al. 2017** Schulman, John; Wolski, Filip; Dhariwal, Prafulla; Radford, Alec; Klimov, Oleg, 2017. Proximal Policy Optimization Algorithms. *Computing Research Repository*, S. 1–12. arXiv: 1707.06347

- Senior et al. 2020** Senior, Andrew W.; Evans, Richard; Jumper, John; Kirkpatrick, James; Sifre, Laurent; Green, Tim; Qin, Chongli; Židek, Augustin; Nelson, Alexander W. R.; Bridgland, Alex; Penedones, Hugo; Petersen, Stig; Simonyan, Karen; Crossan, Steve; Kohli, Pushmeet; Jones, David T.; Silver, David; Kavukcuoglu, Koray; Hassabis, Demis, 2020. Improved protein structure prediction using potentials from deep learning. *Nature* **577** (7792), S. 706–710. DOI: 10.1038/s41586-019-1923-7
- Serdarasan 2013** Serdarasan, Seyda, 2013. A review of supply chain complexity drivers. *Computers & Industrial Engineering* **66** (3), S. 533–540. DOI: 10.1016/j.cie.2012.12.008
- Sewak 2019** Sewak, Mohit, 2019. *Deep Reinforcement Learning*. Singapur: Springer. ISBN: 978-981-13-8284-0
- Shah et al. 2010** Shah, Purvin; Gosavi, Abhijit; Nagi, Rakesh, 2010. A machine learning approach to optimise the usage of recycled material in a remanufacturing environment. *International Journal of Production Research* **48** (4), S. 933–955. DOI: 10.1080/00207540802452157
- Sherstinsky 2020** Sherstinsky, Alex, 2020. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. *Physica D: Nonlinear Phenomena* **404**, S. 132306. DOI: 10.1016/j.physd.2019.132306
- Siegert et al. 2017** Siegert, Jörg; Schlegel, Thilo; Groß, Erwin; Bauernhansl, Thomas, 2017. Standardized Coordinate System for Factory and Production Planning. *Procedia Manufacturing* **9**, S. 127–134. DOI: 10.1016/j.promfg.2017.04.032
- Siegert et al. 2018** Siegert, Jörg; Schlegel, Thilo; Bauernhansl, Thomas, 2018. Matrix Fusion Factory. *Procedia Manufacturing* (23), S. 177–182. DOI: 10.1016/j.promfg.2018.04.013

- Silver et al. 2018** Silver, David; Hubert, Thomas; Schrittwieser, Julian; Antonoglou, Ioannis; Lai, Matthew; Guez, Arthur; Lanctot, Marc; Sifre, Laurent; Kumaran, Dharshan; Graepel, Thore; Lillicrap, Timothy P.; Simonyan, Karen; Hassabis, Demis, 2018.
A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play.
Science **362** (6419), S. 1140–1144.
DOI: 10.1126/science.aar6404
- Simon 1952** Simon, Herbert A., 1952.
On the Application of Servomechanism Theory in the Study of Production Control.
Econometrica **20** (2), S. 247–268.
DOI: 10.2307/1907849
- Simon 1996** Simon, Herbert A., 1996.
The sciences of the artificial. 3. ed. Cambridge, MA, USA: MIT Press.
ISBN: 9780262193740
- Stachowiak 1973** Stachowiak, Herbert, 1973.
Allgemeine Modelltheorie. Wien: Springer.
ISBN: 3-211-81106-0
- Stadtler 2015** Stadtler, Hartmut, 2015.
Supply Chain Management: An Overview.
In: Stadtler, Hartmut; Kilger, Christoph; Meyr, Herbert (Hrsg.): *Supply chain management and advanced planning*.
Berlin: Springer, S. 3–28.
ISBN: 978-3-642-55308-0
- Steenwinckel et al. 2021** Steenwinckel, Bram; de Paepe, Dieter; Vanden Haute, Sander; Heyvaert, Pieter; Bentefrit, Mohamed; Moens, Pieter; Dimou, Anastasia; van den Bossche, Bruno; de Turck, Filip; van Hoecke, Sofie; Ongenaes, Femke, 2021.
FLAGS: A methodology for adaptive anomaly detection and root cause analysis on sensor data streams by fusing expert knowledge with machine learning.
Future Generation Computer Systems **116**, S. 30–48.
DOI: 10.1016/j.future.2020.10.015
- Stricker et al. 2018** Stricker, Nicole; Kuhnle, Andreas; Sturm, Roland; Friess, Simon, 2018.
Reinforcement learning for adaptive order dispatching in the semiconductor industry.
CIRP Annals - Manufacturing Technology **67** (1), S. 511–514.
DOI: 10.1016/j.cirp.2018.04.041

- Stubbe 2010** Stubbe, Kilian, 2010.
Development and Simulation Assessment of Semiconductor Production System Enhancements for Fast Cycle Times.
Dresden, Techn. Univ., Diss., 2010.
urn:nbn:de:bsz:14-qucosa-27343
- Styr et al. 2020** Styr, Angelika; Schumacher, Simon; Bauer, Dennis; Lucke, Dominik; Bauernhansl, Thomas, 2020.
Description Model for the Assessment of Autonomous Production Stages.
Procedia CIRP **93**, S. 353–358.
DOI: 10.1016/j.procir.2020.03.048
- Sutton & Barto 2018** Sutton, Richard S.; Barto, Andrew G., 2018.
Reinforcement Learning: An Introduction. 2nd edition.
Cambridge, MA, USA: MIT Press. Adaptive Computation and Machine Learning.
ISBN: 978026203924
- Szepesvári 2019** Szepesvári, Csaba, 2019.
Algorithms for Reinforcement Learning. Online Edition March 2019. Morgan & Claypool Publishers
- Thomas et al. 2018** Thomas, Tara Elizabeth; Koo, Jinkyu; Chaterji, Somali; Bagchi, Saurabh, 2018.
Minerva: A reinforcement learning-based technique for optimal scheduling and bottleneck detection in distributed factory operations.
In: IEEE (Hrsg.): *Proceedings of the 10th International Conference on Communication Systems & Networks*.
10th International Conference on Communication Systems & Networks (COMSNETS 2018).
Bangalore, Indien, 03.01.2018–07.01.2018, S. 129–136.
ISBN: 978-1-5386-1182-1
- Tian et al. 2019** Tian, Songling; Wang, Taiyong; Zhang, Lei; Wu, Xiaoqiang, 2019.
Real-time shop floor scheduling method based on virtual queue adaptive control: Algorithm and experimental results.
Measurement **147**, S. 106689.
DOI: 10.1016/j.measurement.2019.05.080
- Tomczak 1992** Tomczak, Torsten, 1992.
Forschungsmethoden in der Marketingwissenschaft: Ein Plädoyer für den qualitativen Forschungsansatz.
Marketing ZFP **14** (2), S. 77–87.
DOI: 10.15358/0344-1369-1992-2-77
- Toro et al. 2021** Toro, Carlos; Wang, Wei; Akhtar, Humza, 2021.
Implementing Industry 4.0. Cham: Springer.
ISBN: 978-3-030-67269-0

- Turing 1950** Turing, Alan M., 1950.
Computing Machinery and Intelligence.
Mind **LIX** (236), S. 433–460.
DOI: 10.1093/mind/LIX.236.433
- Ulrich 1981** Ulrich, Hans, 1981.
Die Betriebswirtschaftslehre als anwendungsorientierte Sozialwissenschaft.
In: Geist, Manfred; Köhler, Richard (Hrsg.): *Die Führung des Betriebes*.
Stuttgart: Poeschel, S. 1–25.
ISBN: 3-7910-0308-9
- Ulrich & Probst 1995** Ulrich, Hans; Probst, Gilbert J. B., 1995.
Anleitung zum ganzheitlichen Denken und Handeln: Ein Brevier für Führungskräfte. 4., unveränd. Aufl. Bern: Haupt.
ISBN: 3-258-05182-8
- Ulrich & Hill 1976** Ulrich, Peter; Hill, Wilhelm, 1976.
Wissenschaftstheoretische Grundlagen der Betriebswirtschaftslehre (Teil I).
Wirtschaftswissenschaftliches Studium **5** (7), S. 304–309
- Uzsoy et al. 1992** Uzsoy, Reha; Lee, Chung-Yee; Martin-Vega, Louis A., 1992.
A Review of Production Planning and Scheduling Models in the Semiconductor Industry: Part I: System Characteristics, Performance Evaluation and Production Planning.
IIE Transactions **24** (4), S. 47–60.
DOI: 10.1080/07408179208964233
- Uzsoy et al. 1994** Uzsoy, Reha; Lee, Chung-Yee; Martin-Vega, Louis A., 1994.
A Review of Production Planning and Scheduling Models in the Semiconductor Industry: Part II: Shop-Floor Control.
IIE Transactions **26** (5), S. 44–55.
DOI: 10.1080/07408179408966627
- Vaishnavi & Kuechler 2015** Vaishnavi, Vijay; Kuechler, William, 2015.
Design science research methods and patterns: Innovating information and communication technology. Second edition.
Boca Raton, FL, USA: CRC Press.
ISBN: 9781498715263
- Vakharia & Yenipazarli 2008** Vakharia, Asoo J.; Yenipazarli, Arda, 2008.
Managing Supply Chain Disruptions.
Foundations and Trends in Technology, Information and Operations Management **2** (4), S. 243–325.
DOI: 10.1561/02000000012
- van Houdt et al. 2020** van Houdt, Greg; Mosquera, Carlos; Nápoles, Gonzalo, 2020.
A review on the long short-term memory model.
Artificial Intelligence Review **53** (8), S. 5929–5955.
DOI: 10.1007/s10462-020-09838-1

- van Otterlo & Wiering 2012** van Otterlo, Martijn; Wiering, Marco, 2012.
Reinforcement Learning and Markov Decision Processes.
In: Wiering, Marco; van Otterlo, Martijn (Hrsg.): *Reinforcement Learning*.
Berlin: Springer, S. 3–42.
ISBN: 978-3-642-27644-6
- van Rees 2003** van Rees, Reinout, 2003.
Clarity in the usage of the terms ontology, taxonomy and classification.
CIB REPORT 284 (432), S. 1–8
- VDI 1992** VDI, Gesellschaft Produktionstechnik (ADB) (Hrsg.), 1992.
Lexikon der Produktionsplanung und -steuerung: Begriffszusammenhänge und Begriffsdefinitionen. 4. Aufl.
Düsseldorf: VDI-Verlag.
ISBN: 3-18-401006-6
- VDI 3633** VDI 3633:2018.
Simulation von Logistik-, Materialfluss- und Produktionssystemen: Begriffe
- VDI 5600-1** VDI 5600-1:2016.
Fertigungsmanagementsysteme: Manufacturing Execution Systems – MES
- Vinyals et al. 2019** Vinyals, Oriol; Babuschkin, Igor; Czarnecki, Wojciech M.; Mathieu, Michaël; Dudzik, Andrew; Chung, Junyoung; Choi, David H.; Powell, Richard; Ewalds, Timo; Georgiev, Petko; Oh, Junhyuk; Horgan, Dan; Kroiss, Manuel; Danihelka, Ivo; Huang, Aja; Sifre, Laurent; Cai, Trevor; Agapiou, John P.; Jaderberg, Max; Vezhnevets, Alexander S.; Leblond, Rémi; Pohlen, Tobias; Dalibard, Valentin; Budden, David; Sulsky, Yury; Molloy, James; Paine, Tom L.; Gulcehre, Caglar; Wang, Ziyu; Pfaff, Tobias; Wu, Yuhuai; Ring, Roman; Yogatama, Dani; Wünsch, Dario; McKinney, Katrina; Smith, Oliver; Schaul, Tom; Lillicrap, Timothy P.; Kavukcuoglu, Koray; Hassabis, Demis; Apps, Chris; Silver, David, 2019.
Grandmaster level in StarCraft II using multi-agent reinforcement learning.
Nature **575** (7782), S. 350–354.
DOI: 10.1038/s41586-019-1724-z
- Vlachakis & Apostolou 2014** Vlachakis, Georgios; Apostolou, Dimitris, 2014.
An event-based framework for supply chain management.
In: IEEE (Hrsg.): *Proceedings of the 5th International Conference on Information, Intelligence, Systems and Applications*.
5th International Conference on Information, Intelligence, Systems and Applications (IISA 2014).
Chania, Griechenland, 07.07.2014–09.07.2014, S. 440–445.
ISBN: 978-1-4799-6171-9

- Vlahakis et al. 2018** Vlahakis, George; Apostolou, Dimitris; Kopanaki, Evi, 2018. Enabling situation awareness with supply chain event management. *Expert Systems with Applications* **93**, S. 86–103. DOI: 10.1016/j.eswa.2017.10.013
- Vogel-Heuser et al. 2017a** Vogel-Heuser, Birgit; Bauernhansl, Thomas; ten Hompel, Michael (Hrsg.), 2017. *Handbuch Industrie 4.0: Bd. 1: Produktion*. 2., erweiterte und bearbeitete Auflage. Berlin: Springer Vieweg. ISBN: 978-3-662-45278-3
- Vogel-Heuser et al. 2017b** Vogel-Heuser, Birgit; Bauernhansl, Thomas; ten Hompel, Michael (Hrsg.), 2017. *Handbuch Industrie 4.0: Bd. 2: Automatisierung*. 2., erweiterte und bearbeitete Auflage. Berlin: Springer Vieweg. ISBN: 978-3-662-53247-8
- Walzer et al. 2008** Walzer, Karen; Rode, Jochen; Wunsch, Daniela; Groch, Matthias, 2008. Event-driven manufacturing: Unified management of primitive and complex events for manufacturing monitoring and control. In: IEEE (Hrsg.): *Proceedings of the 2008 IEEE International Workshop on Factory Communication Systems*. 2008 IEEE International Workshop on Factory Communication Systems (WFCS 2008). Dresden, 21.05.2008–23.05.2008, S. 383–391. ISBN: 978-1-4244-2349-1
- Wang et al. 2020** Wang, Dongyuan; Qiao, Fei; Wang, Junkai; Liu, Juan; Kong, Weichang, 2020. Human-Machine Cooperation Based Adaptive Scheduling for a Smart Shop Floor. In: IEEE (Hrsg.): *Proceedings of the 2020 IEEE International Conference on Systems, Man, and Cybernetics*. 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC 2020). Toronto, ON, Canada, 11.10.2020–14.10.2020, S. 788–793. ISBN: 978-1-7281-8526-2
- Wang et al. 2016** Wang, Xiao; Wang, Hongwei; Qi, Chao, 2016. Multi-agent reinforcement learning based maintenance policy for a resource constrained flow line system. *Journal of Intelligent Manufacturing* **27** (2), S. 325–333. DOI: 10.1007/s10845-013-0864-5
- Warnecke 1989** Warnecke, Hans-Jürgen, 1989. Die Produktion als Regelkreis, Überlegungen zu seiner Gestaltung. *Automatisierungstechnische Praxis atp* **31** (3), S. 110–115

Waschneck et al. 2016

Waschneck, Bernd; Bauernhansl, Thomas; Altenmüller, Thomas; Kyek, Andreas, 2016. Production Scheduling in Complex Job Shops from an Industry 4.0 Perspective: A Review and Challenges in the Semiconductor Industry.
In: *Proceedings of the 2016 International Conference on Knowledge Technologies and Data-driven Business*. International Conference on Knowledge Technologies and Data-driven Business (i-KNOW 2016). Graz, Österreich, 18.10.2016–19.10.2016. DOI: 10.5281/zenodo.495155

Waschneck et al. 2017

Waschneck, Bernd; Brian, Lee Wei Fong; Benny, Koh Chey Woon; Rippler, Christoph; Schmid, Gottfried, 2017. Unified Frontend and Backend Industrie 4.0 Roadmap for Semiconductor Manufacturing.
In: *Proceedings of the 2017 International Conference on Knowledge Technologies and Data-driven Business*. International Conference on Knowledge Technologies and Data-driven Business (i-KNOW 2017). Graz, Österreich, 11.10.2017–12.10.2017

Waschneck et al. 2018a

Waschneck, Bernd; Reichstaller, Andre; Belzner, Lenz; Altenmüller, Thomas; Bauernhansl, Thomas; Knapp, Alexander; Kyek, Andreas, 2018. Deep reinforcement learning for semiconductor production scheduling.
In: *Proceedings of the 29th Annual SEMI Advanced Semiconductor Manufacturing Conference*. 29th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC 2018). Saratoga Springs, NY, USA, 30.04.2018–03.05.2018, S. 301–306. ISBN: 978-1-5386-3748-7

Waschneck et al. 2018b

Waschneck, Bernd; Reichstaller, André; Belzner, Lenz; Altenmüller, Thomas; Bauernhansl, Thomas; Knapp, Alexander; Kyek, Andreas, 2018. Optimization of global production scheduling with deep reinforcement learning.
Procedia CIRP **72**, S. 1264–1269. DOI: 10.1016/j.procir.2018.03.212

Waschneck 2020

Waschneck, Bernd, 2020. *Autonome Entscheidungsfindung in der Produktionssteuerung komplexer Werkstattfertigungen*. Stuttgart: Fraunhofer Verlag. Stuttgarter Beiträge zur Produktionsforschung 108. Stuttgart, Univ., Diss., 2020. ISBN: 978-3-8396-1623-9

- Westkämper et al. 2000** Westkämper, Engelbert; Wiendahl, Hans-Hermann; Pritschow, Günter; Remp, Burkhard; Schanz, Michael, 2000. Turbulenz in der PPS – eine Analogie. *wt Werkstattstechnik* **90** (5), S. 203–207
- Westkämper 2006** Westkämper, Engelbert, 2006. *Einführung in die Organisation der Produktion*. Berlin: Springer. Springer-Lehrbuch. ISBN: 978-3-540-26039-4
- Westkämper & Löffler 2016** Westkämper, Engelbert; Löffler, Carina, 2016. *Strategien der Produktion: Technologien, Konzepte und Wege in die Praxis*. Berlin: Springer. ISBN: 978-3-662-48913-0
- Wiendahl et al. 2000** Wiendahl, Hans-Hermann; Westkämper, Engelbert; Remp, Pritschow, Günter, 2000. PPC in a turbulent environment - fundamentals and approaches. In: *Proceedings of the 33rd CIRP International Seminar on Manufacturing Systems*. 33rd CIRP International Seminar on Manufacturing Systems. Stockholm, Schweden, 05.06.2000–07.06.2000, S. 320–325
- Wiendahl 2012** Wiendahl, Hans-Hermann, 2012. *Auftragsmanagement der industriellen Produktion: Grundlagen, Konfiguration, Einführung*. 2011. Aufl. Berlin: Springer. VDI-Buch. ISBN: 978-3-642-19148-0
- Wiendahl et al. 1993a** Wiendahl, Hans-Peter; Pritschow, Günter; Milberg, Joachim, 1993. Produktionsregelung - interdisziplinäre Zusammenarbeit führt zu neuen Ansätzen: Teil 1: Ebenenkonzept als Basis einer kaskadierten Regelung. *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb* **88** (6), S. 265–268
- Wiendahl et al. 1993b** Wiendahl, Hans-Peter; Pritschow, Günter; Milberg, Joachim, 1993. Produktionsregelung - interdisziplinäre Zusammenarbeit führt zu neuen Ansätzen: Teil 2: Modellierung auf Basis der kontinuierlichen Regelungstheorie und der Fuzzy Logic. *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb* **88** (7-8), S. 352–354
- Wiendahl & Scholtissek 1994** Wiendahl, Hans-Peter; Scholtissek, P., 1994. Management and Control of Complexity in Manufacturing. *CIRP Annals - Manufacturing Technology* **43** (2), S. 533–540. DOI: 10.1016/S0007-8506(07)60499-5

- Wiendahl 1997** Wiendahl, Hans-Peter, 1997.
Fertigungsregelung: Logistische Beherrschung von Fertigungsabläufen auf Basis des Trichtermodells. München: Carl Hanser.
ISBN: 3-446-19084-8
- Wiendahl & Wiendahl 2019** Wiendahl, Hans-Peter; Wiendahl, Hans-Hermann, 2019.
Betriebsorganisation für Ingenieure. 9., vollständig überarbeitete Auflage. München: Carl Hanser.
ISBN: 978-3-446-46061-4
- Wiener 1961** Wiener, Norbert, 1961.
Cybernetics or control and communication in the animal and the machine. 2. ed. Cambridge, MA, USA: MIT Press.
ISBN: 0-262-73009-X
- Wilding 1998** Wilding, Richard, 1998.
The supply chain complexity triangle: Uncertainty generation in the supply chain.
International Journal of Physical Distribution & Logistics Management **28** (8), S. 599–616.
DOI: 10.1108/09600039810247524
- Yao et al. 2018** Yao, Xifan; Zhang, Jianming; Li, Yongxiang; Zhang, Cunji, 2018.
Towards flexible RFID event-driven integrated manufacturing for make-to-order production.
International Journal of Computer Integrated Manufacturing **31** (3), S. 228–242.
DOI: 10.1080/0951192X.2017.1407455
- Zäh et al. 2010** Zäh, Michael F.; Reinhart, Gunther; Ostgathe, Martin; Geiger, Florian; Lau, Christian, 2010.
A holistic approach for the cognitive control of production systems.
Advanced Engineering Informatics **24** (3), S. 300–307.
DOI: 10.1016/j.aei.2010.05.014
- Zäpfel 2001** Zäpfel, Günther, 2001.
Grundzüge des Produktions- und Logistikmanagement. 2., unwesentlich veränd. Aufl. München: Oldenbourg.
Internationale Standardlehrbücher der Wirtschafts- und Sozialwissenschaften.
ISBN: 3-486-25618-1
- Zhang et al. 2019** Zhang, Jian; Ding, Guofu; Zou, Yisheng; Qin, Shengfeng; Fu, Jianlin, 2019.
Review of job shop scheduling research and its new perspectives under Industry 4.0.
Journal of Intelligent Manufacturing **30** (4), S. 1809–1830.
DOI: 10.1007/s10845-017-1350-2

Zhang et al. 2017

Zhang, Tao; Xie, Shufang; Rose, Oliver, 2017.
Real-time job shop scheduling based on simulation and Markov decision processes.
In: IEEE (Hrsg.): *Proceedings of the 2017 Winter Simulation Conference*.
2017 Winter Simulation Conference (WSC 2017).
Las Vegas, NV, USA, 03.12.2017–06.12.2017, S. 3899–3907.
ISBN: 978-1-5386-3428-8

Zhang et al. 2009

Zhang, Y. H.; Dai, Q. Y.; Zhong, R. Y., 2009.
An Extensible Event-Driven Manufacturing Management with Complex Event Processing Approach.
International Journal of Control and Automation **2** (3), S. 13–24

Zhao et al. 2019

Zhao, Meng; Li, Xinyu; Gao, Liang; Wang, Ling; Xiao, Mi, 2019.
An improved Q-learning based rescheduling method for flexible job-shops with machine failures.
In: IEEE (Hrsg.): *Proceedings of the 15th International Conference on Automation Science and Engineering*.
15th IEEE International Conference on Automation Science and Engineering (CASE 2019).
Vancouver, BC, Kanada, 22.08.2019–26.08.2019, S. 331–337.
ISBN: 978-1-7281-0356-3

Zhao et al. 2021

Zhao, Yejian; Wang, Yanhong; Tan, Yuanyuan; Zhang, Jun; Yu, Hongxia, 2021.
Dynamic Jobshop Scheduling Algorithm Based on Deep Q Network.
IEEE Access **9**, S. 122995–123011.
DOI: 10.1109/ACCESS.2021.3110242

Zhou et al. 2021a

Zhou, Tong; Tang, Dunbing; Zhu, Haihua; Zhang, Zequn, 2021.
Multi-agent reinforcement learning for online scheduling in smart factories.
Robotics and Computer-Integrated Manufacturing **72**, S. 102202.
DOI: 10.1016/j.rcim.2021.102202

Zhou et al. 2021b

Zhou, Tong; Tang, Dunbing; Zhu, Haihua; Wang, Liping, 2021.
Reinforcement Learning With Composite Rewards for Production Scheduling in a Smart Factory.
IEEE Access **9**, S. 752–766.
DOI: 10.1109/ACCESS.2020.3046784

Zhu et al. 2018

Zhu, Haihua; Wang, Yingcong; Chen, Ming, 2018.
Research on Mechanism of Real-Time MAS based Dynamic Intelligent Manufacturing Systems.
Mechanics **24** (1), S. 121–127.
DOI: 10.5755/j01.mech.24.1.18631

Zimmermann et al. 2006

Zimmermann, Roland; Winkler, Stefan; Bodendorf, Freimut, 2006.
Agent-Based Supply Chain Event Management — Concept and Assessment.
In: IEEE (Hrsg.): *Proceedings of the 39th Annual Hawaii International Conference on System Sciences*.
39th Annual Hawaii International Conference on System Sciences (HICSS 2006).
Kauia, HI, USA, 04.01.2006–07.01.2006.
ISBN: 0-7695-2507-5

A Anhang

A.1 Klassifikation von Ereignissen

Klassifikation beschreibt eine Gruppierung von Entitäten nach bestimmten Kriterien. Sie sorgt also für Struktur, indem sie Daten etikettiert. Der Zweck einer Klassifikation ist es, eine Person oder Software beim Strukturieren oder Finden von Informationen zu unterstützen (van Rees 2003, S. 1–4). Für die Klassifikation von Ereignissen existieren in der Literatur unterschiedliche Ansätze. Nach Genc & Lafortune (2009, S. 301) können Ereignisse in unmittelbar beobachtbare und mittelbar beobachtbare Ereignisse gegliedert werden. Mittelbar beobachtbare Ereignisse sind dabei meist nur anhand ihrer Symptome in einem System identifizierbar (Etzion & Niblett 2011, S. 6). Des Weiteren können Ereignisse nach Etzion & Niblett (2011, S. 6) in erwartete Ereignisse, sogenannte Routineereignisse, und unerwartete Ereignisse unterschieden werden. Unerwartete Ereignisse sind dabei oft negativ konnotiert und werden als Störungen interpretiert, können jedoch ebenso positiv sein. Vakharia & Yenipazarli (2008, S. 250–253) klassifizieren Ereignisse, welche negative Auswirkungen auf ein Wertschöpfungsnetzwerk haben, nach deren Ursache, dem Lebenszyklusabschnitt des Wertschöpfungsnetzwerks, in welchem sie auftreten, nach dem Typ des Ereignisses sowie nach der Entscheidungsebene des Managements. Auf produzierende Unternehmen bezogen, unterscheiden Westkämper et al. (2000, S. 205) und Wiendahl et al. (2000, S. 322) Ereignisse, die als Turbulenzkeime bezeichnet werden, nach internen und externen Ursachen sowie danach, ob Entscheidungen der Produktionsplanung oder -steuerung betroffen sind. Die resultierende Turbulenz kann darüber hinaus objektiv messbar sein oder subjektiv empfunden werden (Westkämper et al. 2000, S. 203). Wiendahl (2012, S. 211) baut darauf auf und klassifiziert die ebenfalls als Turbulenzkeime bezeichneten Ereignisse auch nach der betroffenen Entscheidung, zusätzlich jedoch nach dem betroffenen Prozess eines Wertschöpfungsnetzwerks und feingranularer nach der Ursache. Die Strukturierung der Planungs- und Steuerungsprozesse in einem Wertschöpfungsnetzwerk ähnelt dabei der Definition nach Fleischmann et al. (2015, S. 77) in Abschnitt 2.1.2.

Gezielt auf produzierende Unternehmen in Wertschöpfungsnetzwerken ausgerichtet, können Ereignisse nach ihrem Ursprung, ihrer Ursache, dem Lebenszyklusabschnitt des Wertschöpfungsnetzwerks, der Vorhersagbarkeit ihres Eintretens und ihres Inhalts, ihrem tatsächlichen Inhalt sowie dem notwendigen Entscheidungshorizont klassifiziert werden (Bauer et al. 2019b, S. 89–90). Diese Klassifikation ist in Tabelle A.1 dargestellt.

Tabelle A.1: Klassifikation von Ereignissen für produzierende Unternehmen in Wertschöpfungsnetzwerken (eigene Darstellung in Anlehnung an Bauer et al. 2019b, S. 90)

Eigenschaft	Ausprägung A	Ausprägung B
Ursprung	Intern	Extern
Ursache	Naturereignis	Menschliche Handlung
Lebenszyklusabschnitt des Wertschöpfungsnetzwerks	Design	Betrieb
Vorhersagbarkeit des Eintretens	Stochastisch	Deterministisch
Vorhersagbarkeit des Inhalts	Keine	Bekannte Menge
Inhalt	Mengenänderung	Zeitänderung
Entscheidungshorizont	Kurzfristig	Langfristig

A.2 Daten für Supply Chain Management und Produktionsplanung und -steuerung

Daten für das SCM und die PPS können in Stamm- und Bewegungsdaten gegliedert werden (Schuh et al. 2012b, S. 72). *Stammdaten*, die immer auftragsneutral sind, weisen dabei eine lange Lebensdauer auf und beanspruchen einen umfassenden Pflegeaufwand, da im Planungsprozess permanent auf sie zugegriffen wird (Wiendahl & Wiendahl 2019, S. 169; Schuh et al. 2012b, S. 73; Loos 1999, S. 228). Aufgrund der Auftragsneutralität sind Stammdaten in der Regel wenig volatil. Bei deren Anlage ist jedoch noch nicht abzusehen, wann Änderungen notwendig sein werden und wie lange die Stammdaten gültig sind (Loos 1999, S. 228). Zu den Stammdaten zählen Material- und Ressourcenstammdaten, Arbeitspläne sowie Kunden- und Lieferantenstammdaten (Schuh et al. 2012b, S. 73).

Bewegungsdaten hingegen zeichnen sich durch eine begrenzte Lebensdauer aus. Charakteristisch für Bewegungsdaten ist deshalb der Zeitbezug, beispielsweise ein Lagerzustand bezogen auf einen Zeitpunkt, und die Beschreibung von Statuszuständen, beispielsweise ob ein Produktionsauftrag in Bearbeitung oder abgeschlossen ist. Gleichzeitig beziehen sich Bewegungsdaten immer auf gespeicherte Stammdaten (Schuh et al. 2012b, S. 78). Zu den Bewegungsdaten zählen Lagerbestände, Bedarfe, Fertigungsaufträge und Betriebsdaten, wobei für die PPS die Betriebsdaten von besonderer Bedeutung sind (Loos 1999, S. 242). Betriebsdaten bezeichnen jene zeitlichen, mengenmäßigen, kapazitäts- und auftragsbezogenen Daten, die das Verhalten und den Zustand der Produktion beschreiben (Schuh et al. 2012b, S. 78–79; VDI 1992, S. 125). Während die Verwaltung von Stammdaten typischerweise durch die lokale PPS geschieht, können Bewegungsdaten auch durch das Wertschöpfungsnetzwerk und dessen IT-Systeme beeinflusst werden (Schönsleben 2016, S. 385).

Neben der Unterscheidung in Stamm- und Bewegungsdaten können Daten auch in *Ist- und Soll-Daten* unterteilt werden. Dabei beschreiben Ist-Daten den tatsächlichen Zustand, während Soll-Daten die geplanten Zielgrößen beschreiben. Für die Steuerung von Wertschöpfungsnetzwerk und Produktion sind vor allem die Differenzen zwischen Soll- und Ist-Daten relevant, die zur Überwachung und gegebenenfalls zur Auslösung von Korrekturmaßnahmen dienen (Zäpfel 2001, S. 62). Um der Produktionssteuerung und insbesondere der Auftragsüberwachung einen kontinuierlichen Abgleich zwischen Soll- und Ist-Daten zu ermöglichen, ist eine rechtzeitige Rückmeldung der Betriebsdaten notwendig. Dies gilt besonders in der komplexen Werkstattfertigung, da die Anzahl der Freiheitsgrade und der unvorhergesehenen Störungen eine schnelle Reaktion und steuernde Eingriffe erfordern. In Branchen wie der Halbleiterindustrie ist eine automatische Echtzeit-Datenerfassung deshalb schon seit den 1980er Jahren Standard (Hansmann 2014, S. 375; Mönch et al. 2011, S. 584; VDI 1992, S. 179).

A.3 IT-Unterstützung in der Produktionsplanung und -steuerung

Typischerweise wird die IT-Unterstützung in der PPS anhand der fünf nachfolgend beschriebenen IT-Systeme realisiert (vgl. Abbildung 2.11).

Mit *ERP* werden IT-Systeme in Unternehmen bezeichnet, welche die Planung, Steuerung und Durchführung der Geschäftsprozesse unterstützen. Sie umfassen üblicherweise verschiedene Module, beispielsweise zur Buchführung, zum Controlling, zum Personalwesen aber auch zur PPS (Günther & Tempelmeier 2012, S. 331). Mit den Modulen werden sämtliche Bereiche integriert, welche an der Wertschöpfung eines Unternehmens beteiligt sind, um übergreifend zu planen und zu steuern (Wiendahl & Wiendahl 2019, S. 280). Der Vorteil dieses integrierten Ansatzes ist eine einheitliche Datenbasis, um auch in kurzer Zeit aktuelle Kennzahlen, beispielsweise zur Auftragslage, abfragen zu können (Mönch et al. 2013, S. 264; Kuhn & Hellingrath 2002, S. 129). ERP werden von nahezu allen produzierenden Unternehmen eingesetzt (Lödding 2020, S. 25). Aufgrund von Einschränkungen bei der Planung, beispielsweise durch die Planung mit unbegrenzter Kapazität, eignen sich typische PPS-Module nur begrenzt für die Planung in der komplexen Werkstattfertigung (Mönch et al. 2013, S. 264). Außerdem unterstützt dieser traditionelle Ansatz meist keine schnelle Reaktion auf plötzlich auftretende Ereignisse wie Störungen oder Planänderungen. ERP eignet sich deshalb für die übergeordnete Planung im Unternehmen sowie für die Grobplanung der Produktion, für die Feinplanung werden jedoch weitere IT-Systeme benötigt (Kletti 2015, S. 15).

Ein *Advanced Planning and Scheduling System (APS)*, welches nur in wenigen Unternehmen vorhanden ist, erweitert die groben Planungsmöglichkeiten im ERP durch feinere Planungsmöglichkeiten, vor allem um eine kurzfristige Planung auf Produktionslevel und berücksichtigt dabei auch begrenzte Kapazitäten (Lödding 2020, S. 25; Mönch et al. 2013, S. 28, 264; Kuhn & Hellingrath 2002, S. 129). ERP ist in solch einer Konstellation weiterhin für Transaktionen und deren Ausführung zuständig, APS für die Feinplanung der Produktion (Stadtler 2015, S. 12; Mönch et al. 2013, S. 264). Die wichtigste Feinplanungsaufgabe im APS ist die Ressourceneinsatzplanung, also die Zuordnung von Aufträgen zu Ressourcen und die Planung einer Auftragsreihenfolge (Günther & Tempelmeier 2012, S. 364). Hierfür werden exakte mathematische Optimierungen und einfache Heuristiken genutzt, um einen möglichst breiten Anwendungsbereich abzudecken (Schuh et al. 2013, S. 478; Günther & Tempelmeier 2012, S. 365).

Ein *MES* bietet eine sinnvolle funktionale Ergänzung der übergeordneten Systeme ERP und APS, um aktuelle Fertigungsprozesse noch feingranularer zu planen und zu steu-

ern sowie die Prozesstransparenz zu gewährleisten. Für viele Ereignisse wie Start- und Stopp-Meldungen von Maschinen bieten diese übergeordneten Systeme keine passenden Datencontainer und keine Verarbeitungsmechaniken (VDI 5600-1, S. 2). Etwa die Hälfte der produzierenden Unternehmen setzt deshalb ein MES ein (Lödding 2020, S. 25). Um die Fertigung ideal zu unterstützen, muss ein MES drei Aspekte abdecken: Einen prognostischen Aspekt zur Produktionsplanung, einen aktuellen Aspekt zur Produktionssteuerung und einen historischen Aspekt zur Auswertung und Bewertung von Fertigungsprozessen (VDI 5600-1, S. 6). Für alle drei Aspekte ist es notwendig, dass alle relevanten Daten zu Ressourcen, Prozessen und Prozessflüssen möglichst lückenlos erfasst und in einer Produktionsdatenbank integriert werden, um eine größtmögliche Transparenz der Fertigung zu schaffen (Mönch et al. 2013, S. 9, 252; Kletti 2015, S. 201–202). Aufgrund der vielen Unsicherheiten benötigt die Werkstattfertigung und insbesondere die komplexe Werkstattfertigung gegenüber anderen Fertigungsprinzipien mehr MES-Funktionalitäten, um die Lücke zwischen übergeordneten Systemen und der Steuerung auf Fertigungsebene zu schließen (Kletti 2015, S. 6). Mit der integrierten Datenbasis bilden MES die Grundlage für die nachfolgenden IT-Systeme Scheduling und Dispatching (Mönch et al. 2013, S. 9). Je nach Gestaltung können diese Funktionalitäten auch vollständig oder teilweise in das MES integriert sein (Mönch et al. 2013, S. 252).

Scheduling ist ein regelmäßiger Entscheidungsprozess, der sich mit der Zuordnung von Ressourcen zu Aufträgen über gegebene Zeitperioden befasst. Das Ziel dabei ist die Optimierung einer oder mehrerer Zielgrößen (Pinedo 2016, S. 1; Mönch et al. 2013, S. 27). In der komplexen Werkstattfertigung bezieht sich die Zuordnung auf die Zuweisung von Aufträgen zu Maschinen mit dem Ziel, verfügbare Ressourcen effektiv und effizient zu nutzen (Waschneck et al. 2016, S. 2). Eine Übersicht der Verfahren und Algorithmen zum Scheduling in der komplexen Werkstattfertigung ist Gupta & Sivakumar (2006, S. 1163) zu entnehmen. Die Aufgabe von Scheduling-Systemen entspricht damit der Ressourcenfeinplanung der allgemeinen PPS, wird jedoch in der komplexen Werkstattfertigung erst nach der Auftragsfreigabe ausgeführt. Scheduling-Systeme verarbeiten deshalb nur Fertigungsaufträge, welche bereits freigegeben sind (Mönch et al. 2013, S. 27). Dabei werden idealerweise auch unvorhergesehene Ereignisse wie Maschinenausfälle oder unerwartet längere Prozesszeiten berücksichtigt, da diese einen massiven Einfluss auf die weitere Planung haben

können (Pinedo 2016, S. 4–5). Ergebnis des Scheduling ist ein Plan der Zuordnung von Fertigungsaufträgen zu Zeitintervallen und Ressourcen, üblicherweise mit einem Horizont von einer Schicht oder einem Tag (Mönch et al. 2013, S. 27). Dabei können Scheduling-Systeme verschieden ausgeprägt sein und unterschiedliche Ebenen abbilden. Dies reicht von Einzelmaschinen über Arbeitsgruppen und Arbeitsbereiche bis zu ganzen Fabriken (Mönch et al. 2011, S. 589). Auf Fabrikebene sind Scheduling-Systeme in der Praxis derzeit jedoch nicht im Einsatz, da diese eine zu lange Rechendauer aufweisen und oftmals auch zu kostenintensiv sind (Mönch et al. 2011, S. 584, 594). Für Einzelmaschinen, Arbeitsgruppen und teilweise auch Arbeitsbereiche hingegen werden Scheduling-Systeme in der Industrie eingesetzt, insbesondere an Engpass-Maschinen (Mönch et al. 2013, S. 255).

Dispatching bezeichnet die Zuweisung des nächsten Auftrags zu einer Ressource, sobald diese frei wird (Mönch et al. 2013, S. 27). Dispatching-Systeme, die regelbasiert arbeiten, also den nächsten Auftrag basierend auf Prioritäten der wartenden Aufträge auswählen, sind in der komplexen Werkstattfertigung die populärsten Verfahren zur Produktionssteuerung. Dies ist insbesondere in deren schneller Implementierung, der echtzeitfähigen Reaktion auf Ereignisse, der Nachvollziehbarkeit durch den Menschen und den geringen Rechenaufwand begründet (Waschneck et al. 2016, S. 3; Mönch et al. 2013, S. 27; Sarin et al. 2011, S. 5). Dispatching-Systeme können deshalb gegenüber Scheduling-Systemen auch als Just-in-Time-Prozesse bezeichnet werden (Waschneck et al. 2016, S. 2). Nachteile der regelbasierten Dispatching-Systeme sind, dass diese nicht adaptiv, lernend oder flexibel agieren. Dennoch können sie als Basis für erweiterte Lösungen dienen, welche mit neuen Technologien aufgebaut werden (Waschneck et al. 2016, S. 4). Für eine ausführliche Darstellung populärer Dispatch-Regeln, beispielsweise FIFO, EDD und Shortest Processing Time (SPT), sei auf die Literatur verwiesen (Schuh & Schmidt 2014, S. 215–219; Gupta & Sivakumar 2006, S. 1164).

A.4 Klassifikation von Ansätzen der Fertigungsregelung

Existierender Ansätze der Fertigungsregelung lassen sich nach Niehues (2017) anhand ihres Reglertyps in Bestandsregelung, Kapazitätsregelung, Belastungsregelung, wissensbasierte

Regelung und Regelung durch Neu-/Umplanung einteilen. Die Reglertypen unterscheiden sich dabei in ihren Führungsgrößen, Regelgrößen und Stellgrößen.

Ansätze der *Bestandsregelung* begrenzen den Auftragszugang in die Fertigung in Abhängigkeit von deren Bestandshöhe. Dem liegt der Zusammenhang des Fehlerkreises der Produktionssteuerung zugrunde (vgl. Abbildung 1.2), nach welchem hohe Bestände längere Durchlaufzeiten und demzufolge eine weniger präzise Vorhersagbarkeit der Fertigstellungstermine verursachen. Die Bestandsregelung ermöglicht eine Stabilisierung der Durchlaufzeiten, erlaubt jedoch keine gezielte Beeinflussung einzelner Aufträge (Niehues 2017, S. 22–23). Bekannte Ansätze der Bestandsregelung sind die Belastungsorientierte Auftragsfreigabe (Bechte 1980) und die darauf aufbauende Belastungsorientierte Fertigungsregelung (Wiendahl 1997), welche Bestand und Durchlaufzeiten regeln, sowie die Dezentrale bestandsorientierte Fertigungsregelung (Lödding 2001), welche den Auftragszugang einzelner Arbeitssysteme reguliert.

Bei der *Kapazitätsregelung* wird die Kapazität an die aktuelle Belastung der Fertigung angepasst. Aufgrund der damit einhergehenden Regulierung von Engpässen ist mit einer höheren Liefertermineinhaltung als bei der Bestandsregelung zu rechnen. Jedoch ist hierfür eine Kapazitätsflexibilität in der Fertigung notwendig, welche in der Praxis nicht immer gegeben ist (Niehues 2017, S. 23). Die Dezentrale Rückstandsregelung (Breithaupt 2001), welche auf der Modellbasierten Produktionsregelung (Petermann 1996) aufbaut, realisiert dies durch eine gezielte Regelung des Durchflusses an einzelnen Arbeitssystemen. Die Terminorientierte Kapazitätssteuerung (Begemann 2005) hingegen richtet die Regelung an der Terminerreichung der Aufträge aus.

Werden mehrere Stell- und Regelgrößen berücksichtigt, liegt eine *Belastungsregelung* vor, welche darauf abzielt, die Belastung der Fertigung konstant zu halten. Die möglichen Steuerungsoptionen sind deshalb vielzählig, es erfolgt jedoch auch hier keine exakte Terminierung einzelner Aufträge (Niehues 2017, S. 23–24). Reinhart et al. (2013) und Kienzle (2012) bauen auf der Dezentralen bestandsorientierten Fertigungsregelung auf und erweitern diese um zusätzliche Stellgrößen. Die Bestandsgeregelte Durchflusssteuerung (Busch 1987) baut auf der Kapazitätsregelung auf, um mit lokalen Maßnahmen den Auftragszugang zu Engpasssystemen zu regulieren. Eine Kombination diverser zentraler und dezentraler Bestandsregler beschreiben die Adaptierbare Produktionsregelung (Kiesewetter 1991) und

die Logistische Multizielgrößenoptimierung (Scholz-Reiter et al. 2011). Nyhuis et al. (2009) verbinden die Produktionsregelung, bestehend aus Bestands- und Kapazitätsregelung, mit den logistischen Zielgrößen der PPS (vgl. Abschnitt 2.2.2).

Die *Wissensbasierte Regelung* leitet anhand eines vollständig terminierten Produktionsplans regelbasiert lokale Maßnahmen ab, um diesen Produktionsplan zu modifizieren. Häufig ist dies als Expertensystem gestaltet, also als offener Regelkreis, in welchem der Mensch die letztliche Entscheidung trifft und der Regler als Entscheidungsunterstützung agiert (Niehues 2017, S. 24–25). Beispiele für Expertensysteme sind Mönch (2006, S. 30) zu entnehmen. Aufgebaut werden können wissensbasierte Regelungen durch fest vorgegebene Regeln oder als lernende Systeme, welche aus bekannten Maßnahmen auswählen. Ansätze für lernende Systeme wurden bereits in den 1990er Jahren entwickelt (Wiendahl et al. 1993a; Wiendahl et al. 1993b).

Bei der *Regelung durch Neu-/Umplanung* wird der Produktionsdurchlauf durch Anpassung oder Neuerstellung des Produktionsplans geregelt. Hierunter fallen auch Verfahren, die auf Prioritätsregeln zur Anpassung der Auftragsreihenfolge basieren (Niehues 2017, S. 25). Insbesondere in der komplexen Werkstattfertigung ist dieses Dispatching durch Prioritätsregeln weit verbreitet (vgl. Abschnitt 2.2.4). Prioritätsregeln können dabei auch mit Ansätzen des ML kombiniert werden (Echsler Minguillon & Lanza 2017, S. 631). Auch die adaptive Werkstattsteuerung, welche die Auftragsreihenfolge zwischen zwei Planungszeitpunkten anhand einer Kostenfunktion optimiert, kann in die Klasse der Regelung durch Neu-/Umplanung eingeordnet werden (Niehues et al. 2018; Niehues 2017).

A.5 Kurzfristige, operative Maßnahmen der Produktionssteuerung

Ausgehend vom Modell der Fertigungssteuerung nach Lödging (2016) formuliert Pielmeier (2020, S. 122–127) verschiedene Maßnahmen für die Auftragsfreigabe, die Kapazitätssteuerung und die Reihenfolgebildung. Für die Produktionssteuerung sind dabei insbesondere die Maßnahmen der Kapazitätssteuerung und der Reihenfolgebildung relevant. Tabelle A.2 zeigt die Einordnung der Maßnahmen der Kapazitätssteuerung hinsichtlich der Charakteristik der komplexen Werkstattfertigung, Tabelle A.3 jene der Maßnahmen der Reihenfolgebildung.

Tabelle A.2: Einordnung der Maßnahmen der Kapazitätssteuerung anhand der Charakteristik der komplexen Werkstattfertigung

Maßnahme nach Pielmeier (2020)	Einordnung anhand der Charakteristik der komplexen Werkstattfertigung
Anpassung der Bearbeitungsgeschwindigkeit	Prozessparameter, die auch als Rezept bezeichnet werden, sind abhängig von der Kundenbranche in der Regel durch die Kunden freigegeben und bedürfen bei Änderungen eine neue Freigabe. Kurzfristig ist dies nicht umsetzbar. Innerhalb freigegebener Bänder könnten die Rezepte zwar angepasst werden, die Prozesse der komplexen Werkstattfertigung sind jedoch auf Effizienz und damit auf einen Betriebspunkt ausgelegt.
Einsatz von Springern	Die Kapazitätsbegrenzung der komplexen Werkstattfertigung entsteht in der Regel durch die Maschinenkapazität, nicht die Personalkapazität. Entsprechend kann so wenig bis keine zusätzliche Kapazität geschaffen werden.
Mehrmaschinenbedienung	Durch den hohen Automatisierungsgrad ist die Mehrmaschinenbedienung Standard in der komplexen Werkstattfertigung.
Substitution von Maschinen	Durch Parallelmaschinen ist die Substitution von einzelnen Maschinen in der komplexen Werkstattfertigung inhärent möglich, gegebenenfalls eingeschränkt für Engpassmaschinen. Da die Auswahl des nächsten zu bearbeitenden Loses für die Maschinen beim Dispatcher verbleibt, wird die Maßnahme in dieser Arbeit nicht weiter berücksichtigt.
Überlappende Fertigung	Die Splittung von Losen ist zwar möglich, wird aufgrund der Auslegung von Prozessen auf einen effizienten Betriebspunkt in der komplexen Werkstattfertigung jedoch selten angewendet. Ausnahmen sind Prototypenlose oder Nacharbeit.

Tabelle A.3: Einordnung der Maßnahmen der Reihenfolgebildung anhand der Charakteristik der komplexen Werkstattfertigung

Maßnahme nach Pielmeier (2020)	Einordnung anhand der Charakteristik der komplexen Werkstattfertigung
Zusammenfassung von Arbeitsvorgängen	Eine weitere Parallelisierung zur Ausnutzung von Stillstandszeiten an Anlagen ist in der Regel in der komplexen Werkstattfertigung nicht möglich, da die Auslastungsoptimierung bereits durch die Produktionssteuerung angestrebt wird.
Rüstoptimierung	Die Rüstzeitoptimierung wird im Hinblick auf einen effizienten Betriebspunkt der Prozesse in der komplexen Werkstattfertigung bereits durch das Dispatching umfasst. Hierfür können eigene Prioritätsregeln definiert werden.
Vorziehen von Arbeitsvorgängen	Ein Vorziehen von Aufträgen zur Ausnutzung von Stillstandszeiten an Anlagen ist in der Regel in der komplexen Werkstattfertigung nicht möglich, da die Auslastungsoptimierung bereits durch die Produktionssteuerung angestrebt wird. Etwas breiter formuliert kann dadurch jedoch die Priorität von Aufträgen angepasst werden, um diese früher zu bearbeiten, was in der komplexen Werkstattfertigung möglich und auch Usus ist. Zu beachten sind bei einer signifikanten Prioritätserhöhung dabei die möglichen negativen Auswirkungen von Eillosen.
Rightshifting und Jumping	Eine Verschiebung von Aufträgen in die Zukunft kann ebenfalls als Prioritätsanpassung verstanden werden, in diesem Fall zur späteren Bearbeitung eines Auftrags. Auch dies ist in der komplexen Werkstattfertigung möglich und Usus.

A.6 Modellparameter

Nachfolgend sind die Modellparameter gegliedert in Hyperparameter (vgl. Tabelle A.4), Agentenparameter (vgl. Tabelle A.5) und Trainingsparameter (vgl. Tabelle A.6) dargestellt. Die Hyperparameter sind dabei vom ausgewählten Algorithmus abhängig, in dieser Arbeit A3C (vgl. Abschnitt 6.2.3; Abschnitt A.7). Statische Modellparameter werden über alle Episoden des Trainings konstant gehalten und nur gegebenenfalls nach der Modellevaluation und Abschluss einer Epoche verändert, wohingegen dynamische Modellparameter während des Trainings bei jedem Schritt verändert werden können.

Tabelle A.4: Hyperparameter

Parameter	Typ	Erläuterung
Lernrate α	dynamisch	Die Lernrate α beschreibt die Anpassung gelernter Werte und kann entweder konstant gehalten oder dynamisch abgesenkt werden, wobei eine absinkende Lernrate die Konvergenz unterstützen kann (van Otterlo & Wiering 2012, S. 30). Festzulegen sind für die Lernrate die anfängliche Lernrate α_0 zum Zeitpunkt t_0 , die finale Lernrate α_{Final} und die Anzahl der Schritte n_α , über welche α abgesenkt wird. Die Anzahl der Schritte bezieht sich dabei auf die Epoche. Für eine konstante Lernrate ist $\alpha_0 = \alpha_{Final}$.
Entropiefaktor β	statisch	Mit dem Entropiefaktor β wird die Regularisierung des Entropieterms (vgl. Code 3) kontrolliert. Damit kann Einfluss auf die Exploration des Agenten genommen werden (Mnih et al. 2016, S. 1931).
Diskontierungsfaktor γ	statisch	Der Diskontierungsfaktor γ gewichtet kurz- und langfristige Belohnungen. Je weiter sich γ an 1 nähert, desto weitsichtiger agiert der Agent (Sutton & Barto 2018, S. 55).
Struktur des KNN D_{KNN}	statisch	Die Eingangsschicht des KNN ist in ihrer Größe durch die Größe des Zustandsraums S definiert, die der Ausgangsschicht durch die Größe des Aktionsraums A . Veränderbar sind die versteckten Schichten des KNN. Neben der Anzahl der versteckten Schichten ist auch eine Variation von deren Größe, konkret der Anzahl der Neuronen in den versteckten Schichten, möglich.

Tabelle A.5: Agentenparameter

Parameter	Typ	Erläuterung
Modellgewichte θ	dynamisch	Die Modellgewichte θ , in der englischen Literatur oftmals allgemeiner als Parameter bezeichnet, beschreiben für ein KNN die Gewichte der Neuronen, welche durch den Algorithmus angepasst werden, um abhängig vom Algorithmus die Strategie π , die Wertfunktion $V^\pi(S_t)$, die Qualitätsfunktion $Q^\pi(S_t, A_t)$, die Advantage-Funktion $A^\pi(S_t, A_t)$ oder eine Kombination daraus zu lernen. Ebenfalls durch den Algorithmus wird bestimmt, ob und wie Modellgewichte θ bei einer Aktoranzahl n_{Akt} größer eins zwischen den Aktoren synchronisiert werden.
Flussfaktor-Anpassung ΔFF_{Steuer}	statisch	Wird durch den Agent eine Aktion zur Prioritätserhöhung $a_{\uparrow,i}$ oder zur Prioritätsverringering $a_{\downarrow,i}$ ausgewählt, wird der Steuer-Flussfaktor FF_{Steuer} des entsprechenden Loses angepasst mit $FF_{Steuer} = FF_{Steuer} + \Delta FF_{Steuer}$ (vgl. Abschnitt 5.3.2.1). Für Aktionen zur Prioritätserhöhung $a_{\uparrow,i}$ ist ΔFF_{Steuer} dabei negativ, für Aktionen zur Prioritätsverringering $a_{\downarrow,i}$ positiv.
Schrittweite Δt	statisch	Die Schrittweite Δt beschreibt den zeitlichen Abstand zwischen den Zeitpunkten t und $t+1$. Zu jedem Zeitpunkt t hat der Agent dabei die Möglichkeit, eine Aktion auszuwählen und umzusetzen. Damit entspricht die Schrittweite der Abtast- bzw. Regelperiode des ARIW-Reglers.

Tabelle A.6: Trainingsparameter

Parameter	Typ	Erläuterung
Abbruchbedingung Episode	statisch	Als Abbruchbedingung für eine Episode wird eine Anzahl von Schritten bis zum End-Zeitpunkt T festgelegt. Aus dem Produkt der Schrittweite Δt und der Anzahl der Schritte bis zum End-Zeitpunkt T ergibt sich die Länge einer Episode.
Abbruchbedingung Epoche	statisch	Die Abbruchbedingung für eine Epoche ergibt sich aus der Anzahl von Schritten bis zum End-Zeitpunkt T_{Epoche} . Dabei ist T_{Epoche} ein Vielfaches von T .
Aktoranzahl n_{Akt}	statisch	Mit der Anzahl an Aktoren n_{Akt} wird festgelegt, wie viele Aktoren während des Trainings parallel mit jeweils einer separaten Umgebung interagieren, um die Modellgewichte θ zu lernen.

A.7 Auswahl eines Algorithmus

Zur Auswahl eines Algorithmus für den Agenten des ARIW-Reglers wird anhand der Anforderungen aus Schritt 7 in Abschnitt 5.3.2.2 vorgegangen. Es werden in der Literatur gut dokumentierte und etablierte Algorithmen betrachtet, welche die Mindestanforderungen erfüllen. Dies sind die wertbasierten Algorithmen DQN und Ape-X, jeweils mit den Rainbow-Erweiterungen, sowie die wert- und strategiebasierten Algorithmen A3C, IMPALA, PPO und APPO. Alle Algorithmen werden mit den jeweiligen Standardparametern aus der Literatur anhand einer verkleinerten Umgebung basierend auf der Anwendungsumgebung (vgl. Abschnitt 6.3.1) bis $T_{\text{Epöche}} = 5 \cdot 10^6$ trainiert.

Bei der Betrachtung der wertbasierten Algorithmen zeigt sich, dass sich diese sehr volatil verhalten und nicht robust konvergieren (vgl. Abbildung A.1; Abbildung A.2). Gegenüber dem Ausgangszustand des untrainierten Agenten wird die Belohnung nur marginal verbessert. Einzig DQN Rainbow erreicht zwischenzeitlich eine deutliche Steigerung der Belohnung, diese bleibt jedoch nicht stabil und bricht, wie durch die Kurve DQN Rainbow in Abbildung A.1 dargestellt, bei etwa $2,8 \cdot 10^6$ und $5 \cdot 10^6$ Schritten ein.

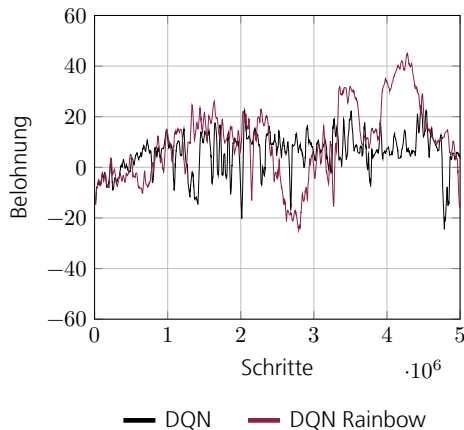


Abbildung A.1: Verlauf der Belohnung für die Algorithmen DQN und DQN Rainbow

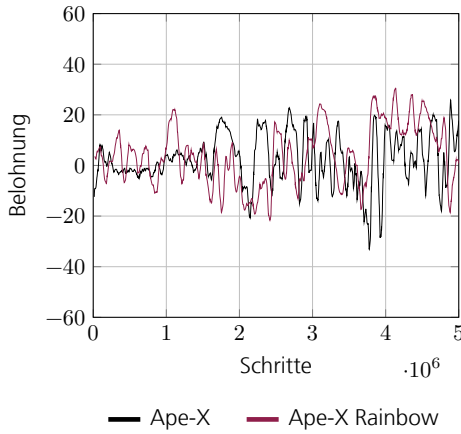


Abbildung A.2: Verlauf der Belohnung für die Algorithmen Ape-X und Ape-X Rainbow

Die wert- und strategiebasierten Algorithmen A3C, IMPALA, PPO und APPO zeigen alle ein konvergierendes Verhalten (vgl. Abbildung A.3; Abbildung A.4). Die verteilten Architekturen von IMPALA und APPO konvergieren dabei bereits bei einer deutlich geringeren Schrittzahl, sind aber weniger robust und etwas volatiler als A3C und PPO. PPO erreicht mit etwa 40 einen etwas höheren Belohnungswert als A3C mit etwa 35, benötigt für das Training jedoch signifikant mehr Rechenzeit¹.

¹engl. wall-clock time

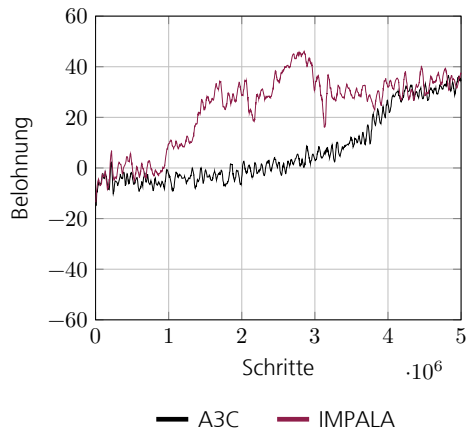


Abbildung A.3: Verlauf der Belohnung für die Algorithmen A3C und IMPALA

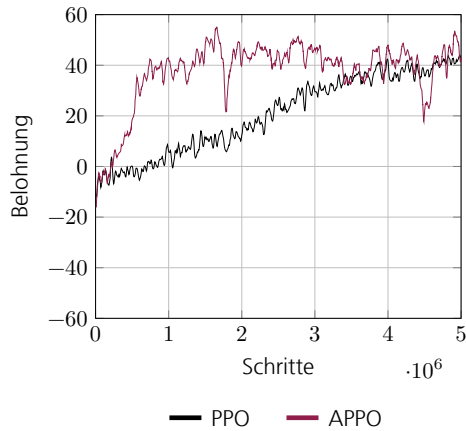


Abbildung A.4: Verlauf der Belohnung für die Algorithmen PPO und APPO

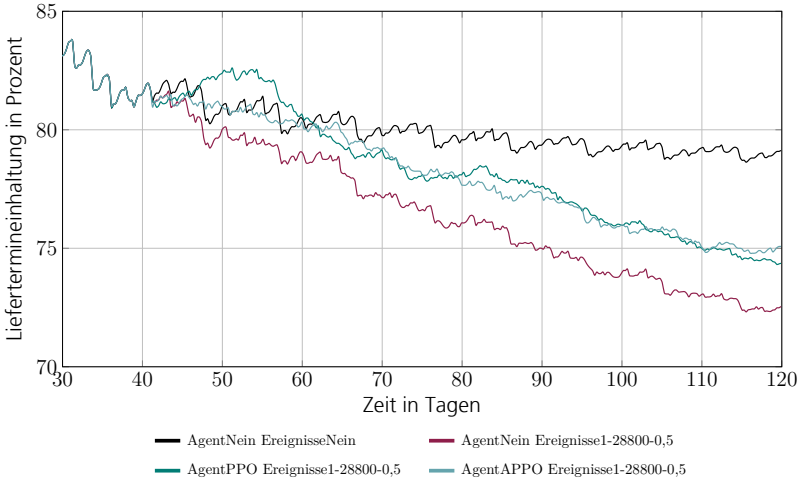
Alle Algorithmen werden anschließend zusätzlich bis $T_{\text{Epoche}} = 1 \cdot 10^7$ trainiert. Mit PPO kann der Belohnungswert hierdurch noch auf etwa 50 gesteigert werden, weitere Änderungen sind nicht aufgetreten. Als Abweichung gegenüber den Standardparametern wurde die Chargengröße² für A3C auf die gleichen Werte wie bei PPO gesetzt. Damit können mit A3C ebenfalls Belohnungswerte von etwa 50 erreicht werden, jedoch bei Konvergenz mit deutlich weniger Schritten und einer signifikant geringeren Rechenzeit, als dies bei PPO der Fall ist.

Nachfolgend werden die wert- und strategiebasierten Algorithmen A3C, IMPALA, PPO und APPO auf ihre Eignung zur Verbesserung der Liefertreue bei Ereignissen aus dem Wertschöpfungsnetzwerk analysiert. Während das Training jeweils beginnend mit einem eingeschwungenen Simulationsmodell mit $t_0 = 30$ Tage und mit Episoden à 12,5 Tage ausgeführt wird, wird für die Validierung der Verbesserung der Liefertermineinhaltung bis $t = 120$ Tage simuliert. Die Performanz der Algorithmen kann dann in Form der Liefertermineinhaltung mit den Szenarien mit und ohne Ereignisse verglichen werden. Im Szenario ohne Ereignisse stabilisiert sich die Liefertermineinhaltung bei 79 Prozent, im Szenario mit Ereignissen bei 72 Prozent. Angelehnt an Pfund et al. (2006, S. 221) und Schritt 5 in Abschnitt 5.3.2.1 werden die Ereignisse anhand einer Poisson-Verteilung mit dem Erwartungswert $\mu_P = 1$ und dem Zeitintervall $t_e = 28800s$ sowie $\Delta F F_{Plan} = 0,5$ generiert. Mit den Algorithmen PPO und APPO kann die Liefertermineinhaltung auf 74 Prozent verbessert werden (vgl. Abbildung A.5a). IMPALA erlaubt eine Verbesserung auf 78 Prozent, A3C sogar auf 80 Prozent (vgl. Abbildung A.5b).

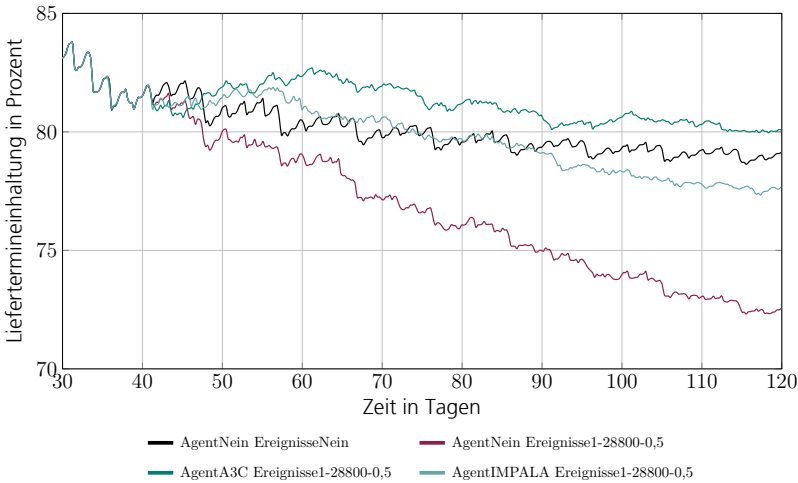
Aus den derzeit verfügbaren Algorithmen wird deshalb A3C für die Implementierung und Validierung ausgewählt. A3C erlaubt die Abbildung von diskreten und kontinuierlichen Größen im Zustandsraum, einen diskreten Aktionsraum, ist in der Literatur sehr gut dokumentiert und in diversen Open-Source-Frameworks verfügbar. Im Vergleich zu allen anderen betrachteten Algorithmen erreicht A3C die höchste Verbesserung der Liefertermineinhaltung, konvergiert robust und mit wenigen Schritten sowie in einer akzeptablen Rechenzeit. Dass die Liefertermineinhaltung sogar über das Szenario ohne Ereignisse hinaus verbessert werden kann, ist mit der vereinfachten Anwendungsumgebung für die Algorithmusaus-

²Die Chargengröße, engl. batch size, gibt die Länge n_τ der Zustands-Aktions-Belohnungs-Trajektorie τ an, welche für das Training gleichzeitig durch den Algorithmus verarbeitet wird.

wahl zu erklären. Da der Agent nicht nur bei Ereignissen, sondern zu jedem Zeitschritt eine Aktion ausführen kann, können auch produktionsinterne Optimierungspotenziale adressiert werden.



(a) PPO und APPO



(b) A3C und IMPALA

Abbildung A.5: Liefertermineinhaltung mit den Algorithmen PPO, APPO, A3C und IMPALA

A.8 Hyperparameter-Optimierung

Die Hyperparameter-Optimierung wird anhand der Variation in Tabelle 6.2 mit der Basislinie $\alpha = 1 \cdot 10^{-4}$, $\beta = 0,01$, $\gamma = 0,99$ und $D_{KNN} = 256$ – 256 durchgeführt.

Lernrate In Bezug auf die Lernrate α werden vier statische Variationen und eine dynamische Variation analysiert. Mit $\alpha = 5 \cdot 10^{-4}$ steigt der Verlauf der Belohnung deutlich steiler und erreicht bereits bei etwa $4 \cdot 10^5$ Schritten nahezu seinen Maximalwert, während dies bei der Basislinie mit $\alpha = 1 \cdot 10^{-4}$ erst bei etwa $1,6 \cdot 10^6$ Schritten der Fall ist. Anschließend konvergiert die Belohnung jedoch zu einem niedrigeren Wert und wird instabil. Noch deutlicher wird der Effekt bei größeren Lernraten. Während mit $\alpha = 1 \cdot 10^{-3}$ noch eine kurze, steile Zunahme des Belohnungsverlaufs zu erkennen ist, ist der Verlauf der Belohnung mit $\alpha = 1 \cdot 10^{-2}$ direkt von Beginn an instabil und sehr volatil. Zunehmende Lernraten erlauben damit zwar in Grenzen eine Beschleunigung des Trainings durch Konvergenz in weniger Schritten, bergen jedoch die Gefahr eines instabilen Agenten (vgl. Abbildung A.6).

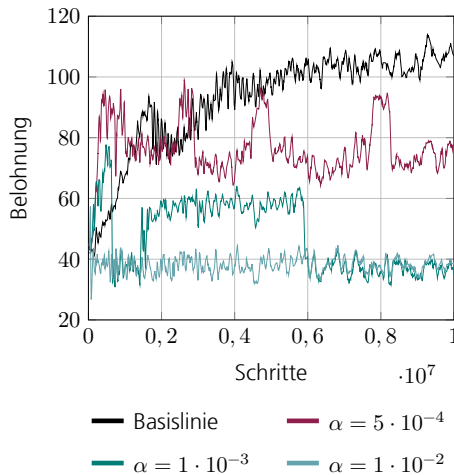


Abbildung A.6: Einfluss der Variation der Lernrate, höhere Werte als in der Basislinie

Mit einer Lernrate $\alpha = 1 \cdot 10^{-5}$ und damit niedriger als bei der Basislinie wird der Belohnungsverlauf robuster und weniger volatil. Allerdings müssen zum Erreichen desselben

Maximalwerts mehr als doppelt so lange Epochen durchlaufen und damit auch eine signifikant höhere Rechenzeit aufgewendet werden. Kombiniert werden können die Vorteile hoher und niedriger Lernraten in einer dynamischen Variation mit $\alpha = [5 \cdot 10^{-4}, 1 \cdot 10^{-5}]$. Mittels Code 2 in Abschnitt 5.3.2.2 wird die Lernrate α dabei über $n_\alpha = 3 \cdot 10^6$ Schritte von $\alpha_0 = 5 \cdot 10^{-4}$ auf $\alpha_{Final} = 1 \cdot 10^{-5}$ abgesenkt. Hierdurch wird derselbe Maximalwert der Belohnung wie in der Basislinie erreicht, jedoch bei schnellerer Konvergenz und einem stabileren Verhalten im konvergierten Zustand (vgl. Abbildung A.7).

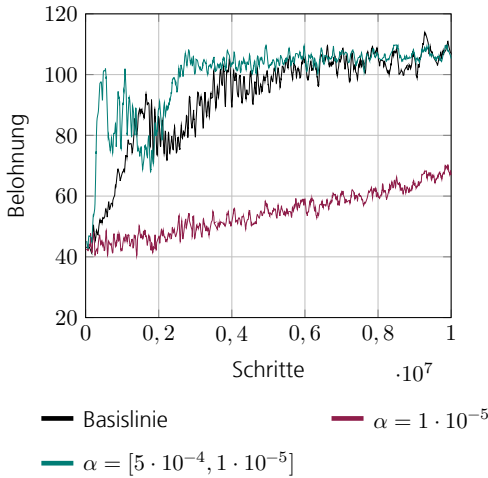


Abbildung A.7: Einfluss der Variation der Lernrate, niedrigere Werte als in der Basislinie und absinkende Werte

Entropiefaktor Für den Entropiefaktor β werden zwei statische Variationen durchgeführt. Mit $\beta = 0$, also ohne Berücksichtigung der Entropie, wird gegenüber der Basislinie ein höherer Maximalwert der Belohnung erreicht. Bei einer stärkeren Berücksichtigung der Entropie mit $\beta = 0,02$ wird ein niedrigerer Maximalwert der Belohnung erreicht. Dies ist anhand des direkten Einflusses der Entropie auf die Strategie des Agenten erklärbar (vgl. Code 3). Die Volatilität der Belohnungskurve ist jedoch bei beiden Variationen vergleichbar zur Basislinie, ebenso der Konvergenzzeitpunkt. Ein Einfluss hinsichtlich beschleunigter

Exploration kann deshalb mittels des Entropiefaktors für die vorliegende Anwendungsumgebung nicht erzielt werden, jedoch eine höhere maximale Belohnung (vgl. Abbildung A.8).

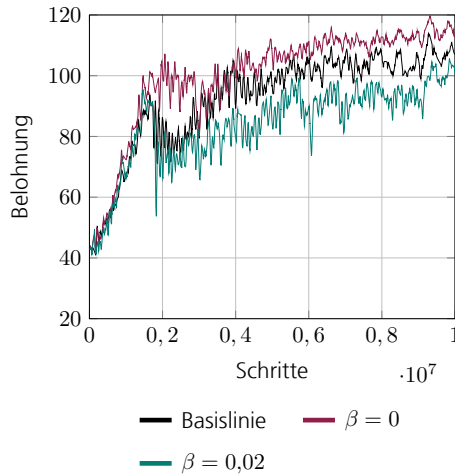


Abbildung A.8: Einfluss der Variation des Entropiefaktors

Diskontierungsfaktor Hinsichtlich des Diskontierungsfaktors γ werden zwei statische Variationen durchgeführt. Mit $\gamma = 0,9999$, also einer stärkeren Berücksichtigung zukünftiger Belohnungen, wird ein zur Basislinie vergleichbarer Maximalwert der Belohnung erreicht, deren Verlauf ist jedoch instabiler. Dieses Verhalten kann damit erklärbar sein, dass sich zu diesem Zeitpunkt bereits überwiegend Lose im Zustandsraum befinden, auf welche die gewählte Aktion nur noch einen marginalen Einfluss hat. Ein deutlich niedrigerer Maximalwert der Belohnung wird mit $\gamma = 0,9$ erreicht. Langfristige Konsequenzen ausgewählter Aktionen werden hierdurch nicht ausreichend berücksichtigt. Der beste Wert für die vorliegende Anwendungsumgebung entspricht folglich dem Standardwert aus der Literatur in Form der Basislinie (vgl. Abbildung A.9).

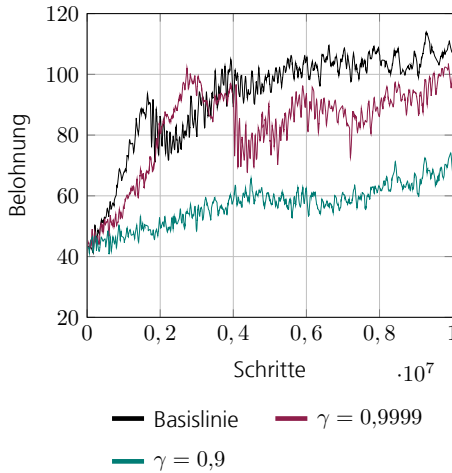
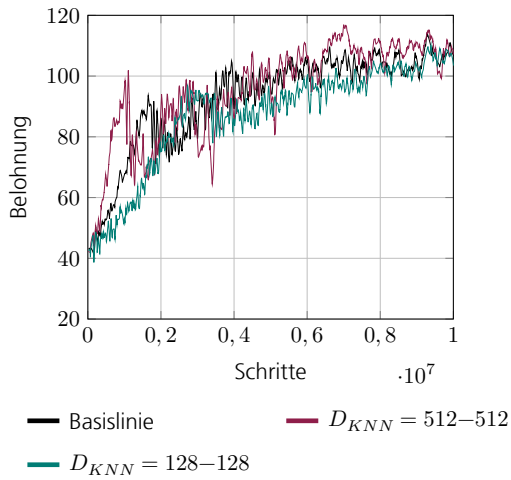


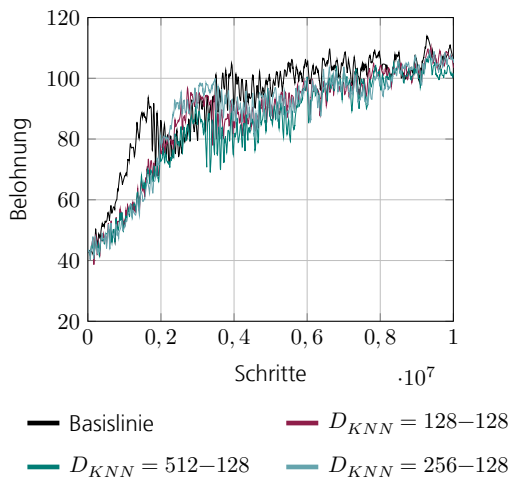
Abbildung A.9: Einfluss der Variation des Diskontierungsfaktors

Struktur des KNN Bezüglich der Struktur des KNN D_{KNN} werden acht Variationen hinsichtlich der Größe des Netzes in Form der Anzahl an Neuronen pro versteckter Schicht, der Anzahl versteckter Schichten und der Ergänzung eines Long Short-Term Memory (LSTM)³ untersucht. Eine höhere Anzahl an Neuronen pro versteckter Schicht mit $D_{KNN} = 512-512$ konvergiert dabei etwas schnell, erreicht jedoch denselben Maximalwert der Belohnung wie die Basislinie und ist gleichzeitig instabiler. Mit $D_{KNN} = 128-128$ wird ebenfalls derselbe Maximalwert der Belohnung erreicht, auch die Stabilität ist vergleichbar mit der Basislinie, die Konvergenz jedoch langsamer (vgl. Abbildung A.10a). Bei der Betrachtung unterschiedlich großer versteckter Schichten zeigt sich, dass die Anzahl der Neuronen in der letzten versteckten Schicht vor der Ausgangsschicht dominant für den Verlauf der Belohnungskurve ist. Unabhängig von der Anzahl der Neuronen in der ersten Schicht entspricht der Verlauf jeweils nahezu identisch dem Verlauf von $D_{KNN} = 128-128$ (vgl. Abbildung A.10b).

³LSTM, ursprünglich publiziert von Hochreiter & Schmidhuber (1997), sind eine Form rekurrenter KNN, die in der Lage sind, neben Eingangsgrößen auch langfristige Abhängigkeiten zwischen Sequenzen abzubilden und zu speichern (Goodfellow et al. 2016, S. 25, 408–411). Hierfür sind LSTM aus Blöcken aufgebaut, welche jeweils aus einer inneren Zelle, einem Eingangs-Gate, einem Ausgangs-Gate und einem Vergiss-Gate bestehen (van Houdt et al. 2020, S. 5931). Für eine ausführliche Beschreibung von LSTM sei auf die Literatur verwiesen (van Houdt et al. 2020; Sherstinsky 2020).



(a) Versteckte Schichten mit gleich großer Neuronenanzahl



(b) Versteckte Schichten mit unterschiedlich großer Neuronenanzahl

Abbildung A.10: Einfluss der Variation der Struktur des KNN, Anzahl der Neuronen

Auch die Tiefe des KNN in Form der Anzahl versteckter Schichten spiegelt sich im Verlauf der Belohnungskurve wider. Mit $D_{KNN} = 256-256-256$, also einem tieferen KNN, wird zwar derselbe Maximalwert der Belohnung erreicht, das Verhalten ist jedoch instabiler. Für ein weniger tiefes KNN mit $D_{KNN} = 256$ entspricht der Verlauf der Belohnungskurve nahezu der Basislinie bei gleichzeitig geringerer Rechenzeit (vgl. Abbildung A.11). Um eine Unteranpassung⁴ für die Validierung zu vermeiden, soll hinsichtlich der Tiefe des KNN deshalb die Basislinie beibehalten werden.

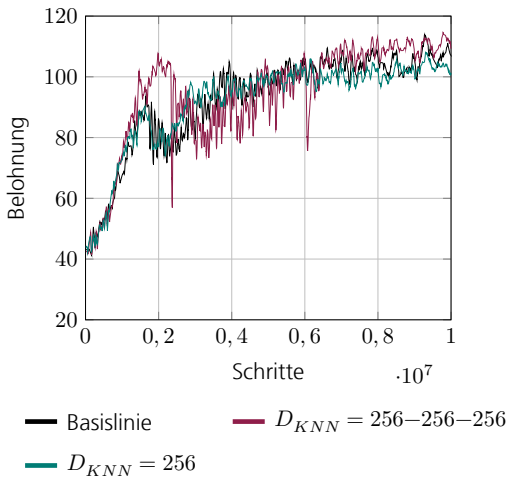


Abbildung A.11: Einfluss der Variation der Struktur des KNN, Anzahl der versteckten Schichten

Im Vergleich zur Basislinie kann durch die Ergänzung eines LSTM keine Verbesserung erzielt werden. Zum Ende der Epoche wird zwar der gleiche Maximalwert der Belohnung erreicht, gleichzeitig ist deren Verlauf jedoch instabiler (vgl. Abbildung A.12a). Bei tieferen KNN kann die Ergänzung eines LSTM eine stabilisierende Wirkung haben, sodass die Zunahme der Belohnungskurve im Verlauf der Epoche gleichmäßiger wird (vgl. Abbildung A.12b). Wenngleich für die vorliegende Anwendungsumgebung durch die Ergänzung eines LSTM

⁴Eine Unteranpassung, engl. *underfitting*, liegt vor, wenn das Modell nicht in der Lage ist, einen ausreichend niedrigen Fehlerwert bzw. resultierend daraus eine ausreichend hohe Belohnung zu erzielen. Einen großen Einfluss darauf hat die Kapazität des Modells, also dessen Größe und Tiefe. Modelle mit geringer Kapazität können sich teilweise nur schwer an einen Datensatz anpassen. Gleichzeitig ist jedoch auch die Gefahr der Überanpassung, engl. *overfitting*, zu berücksichtigen. Modelle mit hoher Kapazität können sich oftmals übermäßig gut anpassen, indem sie sich Eigenschaften der Trainingsmenge merken, und anschließend weniger gut auf unbekannte Daten generalisieren (Goodfellow et al. 2016, S. 111–112).

keine Verbesserung gegenüber der Basislinie erzielt werden kann, so kann ein LSTM für größere Anwendungsumgebungen, welche ein größeres und/oder tieferes KNN erfordern, dennoch zielführend sein.

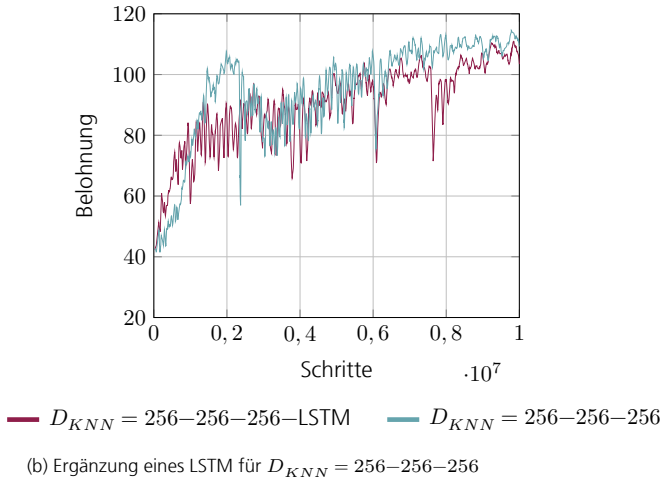
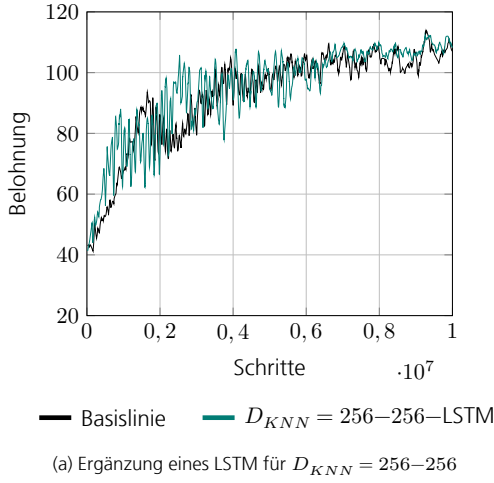


Abbildung A.12: Einfluss der Variation der Struktur des KNN, Ergänzung eines LSTM

A.9 Parametervariation der Agentenparameter

Die Variation der Agentenparameter wird anhand Tabelle 6.3 mit der Basislinie $\Delta FF_{Steuer} = 0,25$ und $\Delta t = 3600s$ durchgeführt.

Flussfaktor-Anpassung Für die Flussfaktor-Anpassung ΔFF_{Steuer} werden vier statische Variationen durchgeführt. Mit $\Delta FF_{Steuer} = 0,125$ und damit kleiner als in der Basislinie ist die Charakteristik der Belohnungskurve ähnlich, es wird jedoch ein geringerer Maximalwert erreicht. Einen höheren Maximalwert bei gleichzeitig stabilerem Verhalten erreicht $\Delta FF_{Steuer} = 0,5$. Mit zunehmendem ΔFF_{Steuer} wird die Belohnungskurve anschließend nicht weiter beeinflusst, gleichzeitig steigt der Anteil der Aktion "keine Handlung" signifikant an (vgl. Abbildung A.13).

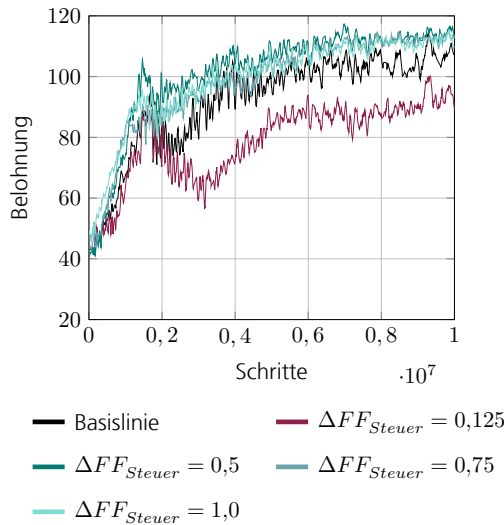


Abbildung A.13: Einfluss der Variation der Flussfaktor-Anpassung

Schrittweite Hinsichtlich der Schrittweite Δt werden zwei statische Variationen untersucht. Um die Episodenlänge dabei konstant zu halten, wird die Anzahl von Schritten bis zum End-Zeitpunkt T umgekehrt proportional zur Schrittweite der Basislinie angepasst. Im

Vergleich zur Basislinie kann bei der Variation der Schrittweite jedoch keine Verbesserung erzielt werden, da das Verhalten bei kleineren und größeren Werten instabiler wird (vgl. Abbildung A.14).

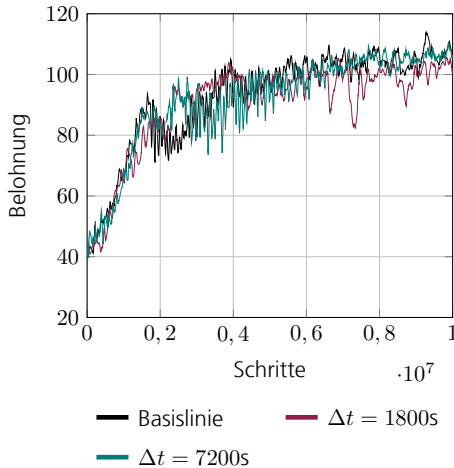


Abbildung A.14: Einfluss der Variation der Schrittweite

A.10 Grenzbetrachtung

Die Variation der Ereignisse für die Grenzbetrachtung wird anhand Tabelle 6.5 mit den Modellparametern der Trainingslinie und Episoden mit $T = 2880$ Schritten durchgeführt.

Zeitintervall Ohne den ARIW-Regler beträgt die Liefertermineinhaltung im Szenario ohne Ereignisse 84 Prozent. Im Szenario mit Ereignissen fällt die Liefertermineinhaltung abhängig vom Zeitintervall t_e der Poisson-Verteilung. Ein kürzeres Zeitintervall bedeutet dabei, dass mehr Ereignisse während der Episode auftreten. Für $1-86400-0,5$ fällt die Liefertermineinhaltung um 2 Prozentpunkte auf 82 Prozent, durch den ARIW-Regler kann sie auf 82,5 Prozent verbessert werden (vgl. Abbildung A.15). Mit den Ereignissen $1-14400-0,5$, welche ein Fallen der Liefertermineinhaltung um 8 Prozentpunkte auf 76 Prozent bewirken,

erreicht der ARIW-Regler eine Verbesserung der Liefertermineinhaltung auf 79,5 Prozent (vgl. Abbildung A.16).

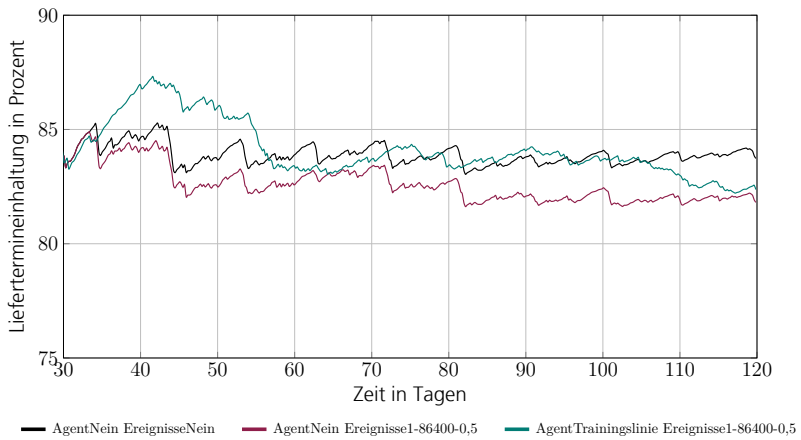


Abbildung A.15: Liefertermineinhaltung mit Ereignissen 1-86400-0,5

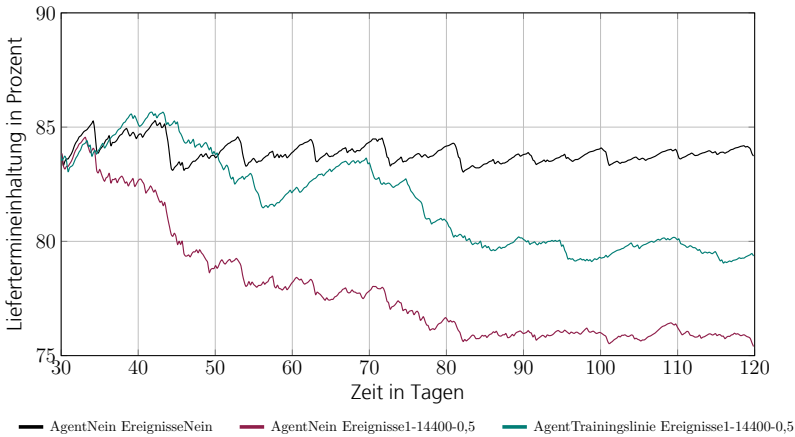


Abbildung A.16: Liefertermineinhaltung mit Ereignissen 1-14400-0,5

Für 1–3600–0,5 fällt die Liefertermineinhaltung um 16 Prozentpunkte auf 68 Prozent, durch den ARIW-Regler kann sie auf 72 Prozent verbessert werden. Es ist jedoch auch zu erkennen, dass sich die Liefertermineinhaltung bei 120 Tagen noch nicht stabilisiert hat und weiter fällt (vgl. Abbildung A.17). Das weitere Abfallen der Liefertermineinhaltung ist auch bei 1–900–0,5 zu sehen. Die Liefertermineinhaltung fällt hier um 21 Prozentpunkte auf 63 Prozent, durch den ARIW-Regler kann sie auf 67 Prozent verbessert werden (vgl. Abbildung A.18).

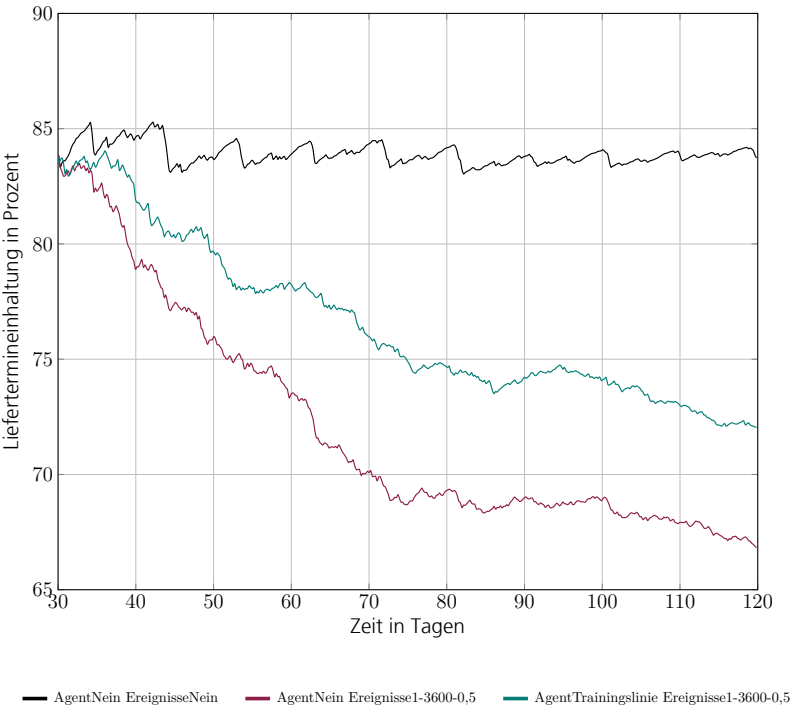


Abbildung A.17: Liefertermineinhaltung mit Ereignissen 1-3600-0,5

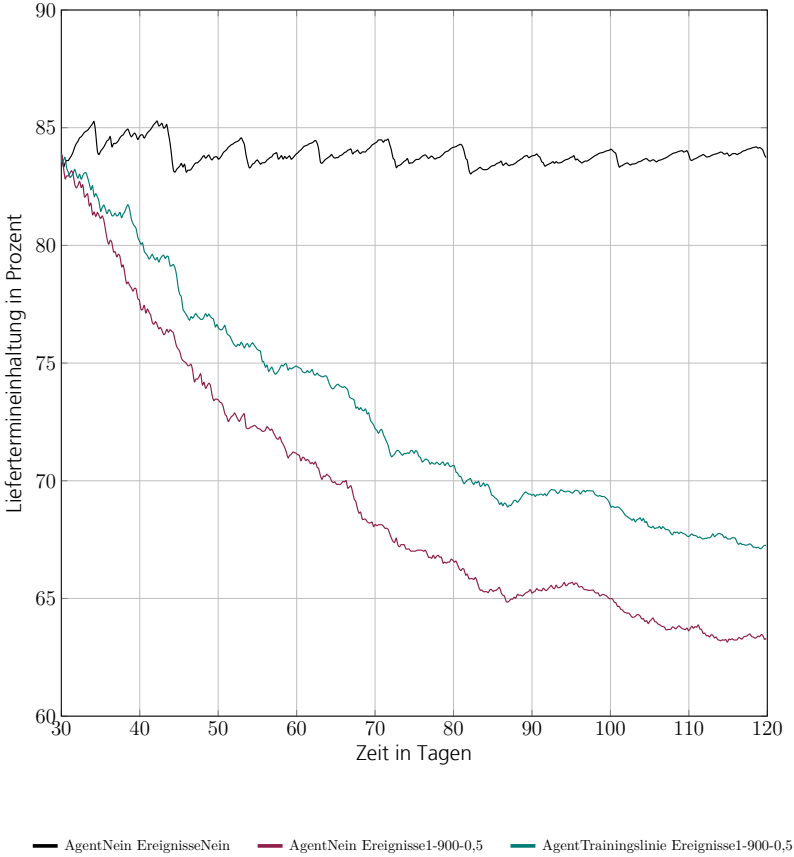


Abbildung A.18: Liefertermineinhaltung mit Ereignissen 1-900-0,5

Lieferterminänderung Ohne den ARIW-Regler beträgt die Liefertermineinhaltung im Szenario ohne Ereignisse 84 Prozent. Im Szenario mit Ereignissen fällt die Liefertermineinhaltung abhängig von der Lieferterminänderung ΔFF_{Plan} der Ereignisse. Mit einem größeren ΔFF_{Plan} werden Lose durch Ereignisse entsprechend stärker beeinflusst. Für $1-28800-0,25$ fällt die Liefertermineinhaltung um 2,5 Prozentpunkte auf 81,5 Prozent, durch den ARIW-Regler kann sie auf 83,5 Prozent verbessert werden (vgl. Abbildung A.19). Bei den Ereignissen $1-28800-0,75$, welche ein Absinken der Liefertermineinhaltung um 5 Prozentpunkte auf 79 Prozent bewirken, kann der ARIW-Regler diese auf 80 Prozent verbessern (vgl. Abbildung A.20). Das Verhalten für $1-28800-1,0$ ist dabei sehr ähnlich, ein weiteres Absinken der Liefertermineinhaltung für zunehmende ΔFF_{Plan} ist nicht mehr zu beobachten (vgl. Abbildung A.21). Die Ereignisse $1-28800-1,25$ lassen die Liefertermineinhaltung deshalb ebenfalls um 5 Prozentpunkte auf 79 Prozent absinken, der Einsatz des ARIW-Reglers sorgt jedoch für ein weiteres Absinken auf 77 Prozent (vgl. Abbildung A.22). Bei einer Lieferterminänderung $\Delta FF_{Plan} > 1,0$ ist der ARIW-Regler also nicht mehr in der Lage, durch die ausgewählten Aktionen eine Verbesserung der Liefertermineinhaltung zu bewirken. Durch die Gleichsetzung des Agentenparameters der Flussfaktor-Anpassung ΔFF_{Steuer} mit der Lieferterminänderung ΔFF_{Plan} kann dabei auch keine Verbesserung erzielt werden.

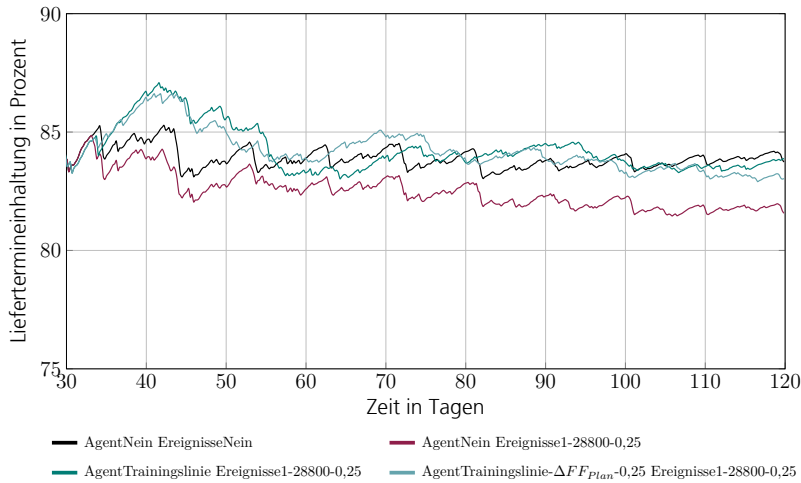


Abbildung A.19: Liefertermineinhaltung mit Ereignissen 1-28800-0,25

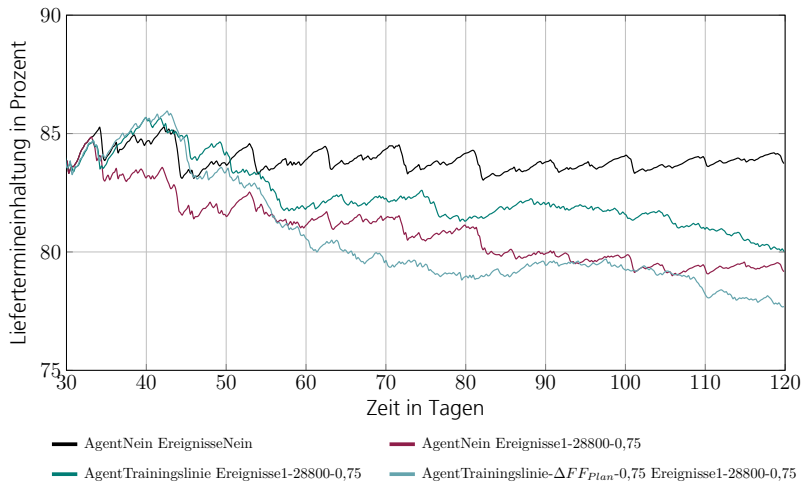


Abbildung A.20: Liefertermineinhaltung mit Ereignissen 1-28800-0,75

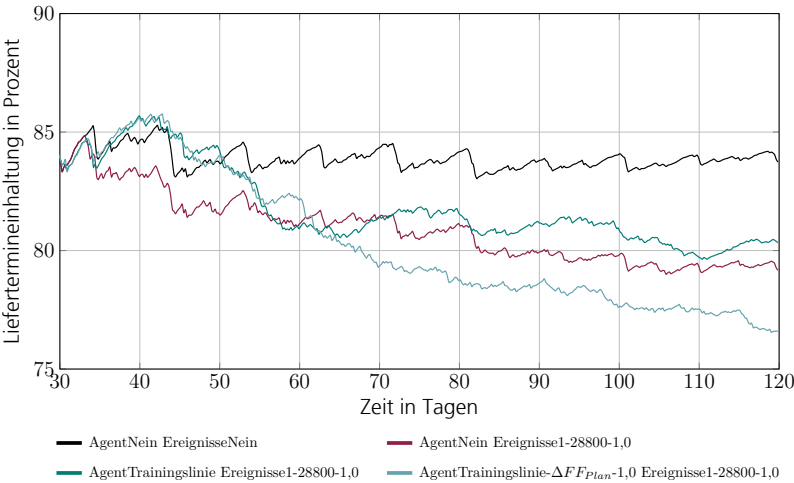


Abbildung A.21: Liefertermineinhaltung mit Ereignissen 1-28800-1,0

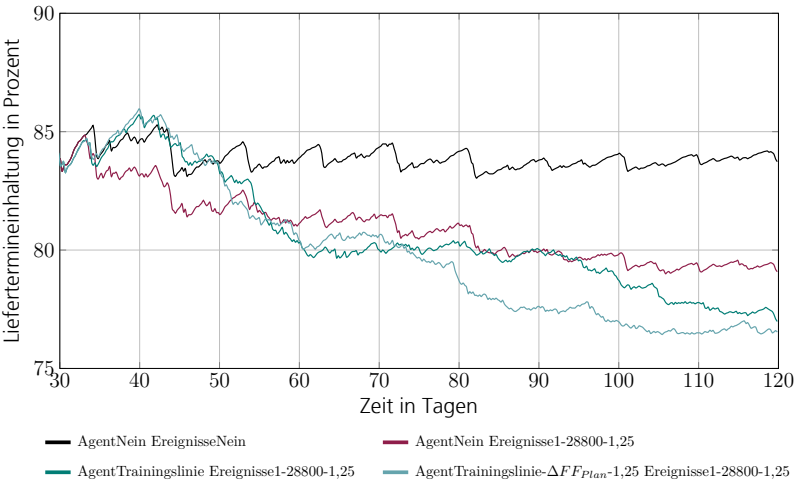
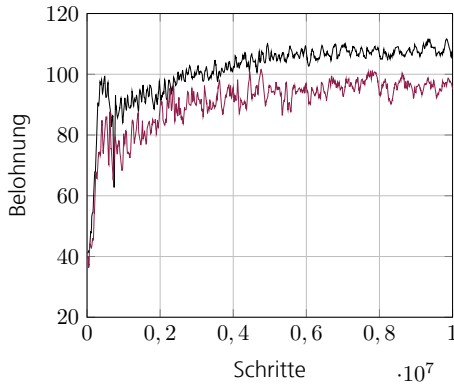


Abbildung A.22: Liefertermineinhaltung mit Ereignissen 1-28800-1,25

Zeitintervall und Lieferterminänderung Die Belohnungskurve des ARIW-Reglers mit dem Modellparametern der Trainingslinie, trainiert auf die Ereignisse 1–14400–0,75, ist Abbildung A.23 zu entnehmen. Die Auswirkung auf die Liefertermineinhaltung ist in Abbildung 6.14 dargestellt.



— Trainingslinie — TrainingslinieEreignisse1-14400-0,75 Ereignisse1-14400-0,75

Abbildung A.23: Trainingslinie trainiert mit Ereignissen 1-14400-0,75

A.11 Studentische Arbeiten

Im Rahmen dieser Dissertation wurden unter wissenschaftlicher, fachlicher und inhaltlicher Anleitung des Autors studentische Arbeiten angefertigt. Im Folgenden sind die betreuten studentischen Arbeiten in alphabetischer Reihenfolge des Nachnamens der Autorinnen und Autoren aufgeführt.

Marinas Tapia, Alicia del Carmen

Development of a new Classification System for Production Organization Types and Development of an optimized Production Principle for Mass Personalization, Universität Stuttgart, 2018

Schlereth, Andreas

Aufbau eines Modells zur ereignisgetriebenen Produktionssteuerung, Karlsruher Institut für Technologie, 2018

Schneider, Mirko

Charakterisierung existierender Ansätze zum Einsatz von Reinforcement Learning in der Produktionsplanung und -steuerung, Universität Stuttgart, 2021

Umgelter, Daniel

Entwicklung und Implementierung einer Produktionssteuerung für ein Simulationsmodell einer komplexen Werkstattfertigung in der Halbleiterindustrie, Universität Stuttgart, 2020

Würth, Patrick

Analyse intelligenter Produktionssteuerungsverfahren am Beispiel der Front-End-Fertigung in der Halbleiterproduktion, Universität Stuttgart, 2017

A.12 Thematisch bezogene Veröffentlichungen des Autors

Bauer, Dennis; Schlereth, Andreas; Bauernhansl, Thomas, 2019.

Systemdynamik-Ansätze zur Optimierung der PPS: Nutzen und Anwendung in der Halbleiter-industrie.

wt Werkstattstechnik **109** (4), S. 225–229.

DOI: 10.5281/zenodo.2653050

Bauer, Dennis; Bauernhansl, Thomas; Sauer, Alexander, 2019.

Enhanced Classification of Events for Manufacturing Companies in Supply Networks.

Procedia CIRP **81**, S. 87–92.

DOI: 10.1016/j.procir.2019.03.016

Bauer, Dennis; Bauernhansl, Thomas; Sauer, Alexander, 2020.

Wertschöpfungsnetzwerk und Produktion integrieren: Ein adaptiver, regelkreisbasierter Ansatz.

wt Werkstattstechnik **110** (3), S. 92–97

Bauer, Dennis; Bauernhansl, Thomas; Sauer, Alexander, 2020.

Approach for an Adaptive Control Loop between Supply Network and Manufacturing.

Procedia CIRP **93**, S. 1552–1557.

DOI: 10.1016/j.procir.2020.04.058

Bauer, Dennis; Bauernhansl, Thomas; Sauer, Alexander, 2021.

Improvement of Delivery Reliability by an Intelligent Control Loop between Supply Network and Manufacturing.

Applied Sciences **11** (5), S. 2205.

DOI: 10.3390/app11052205

Bauer, Dennis; Böhm, Markus; Bauernhansl, Thomas; Sauer, Alexander, 2021.

Increased resilience for manufacturing systems in supply networks through data-based turbulence mitigation.

Production Engineering **15** (3-4), S. 385–395.

DOI: 10.1007/s11740-021-01036-4

