

# Nonlinear Model Predictive Control for Industrial Manufacturing Processes with Reconfigurable Machine Tools

Vom Fachbereich Produktionstechnik  
der  
UNIVERSITÄT BREMEN

zur Erlangung des Grades  
Doktor-Ingenieur  
genehmigte

Dissertation

von

M.Sc. Qiang Zhang

Gutachter: Prof. Dr. Jürgen Pannek

Gutachter: Prof. Dr.-Ing. Sergio Lucia (Technische Universität Berlin)

Tag der mündlichen Prüfung: 18.07.2019



# ACKNOWLEDGEMENT

---

First and foremost I am indebted to my advisor Prof. Dr. Jürgen Pannek for his constant availability, immense knowledge, professional guidance and unfailing patience during my PhD study. A lot of constructive comments and suggestions were received to improve and complete this thesis, which would have been impossible without his continued support, help and encouragement.

Besides my advisor, I am very grateful to Dr. -Ing Ingrid Rügge for her much aid and support. She is the managing director and the “mother” of the International Graduate School (IGS) for Dynamics in Logistics. I am very grateful to IGS for providing me with the structured doctoral training program and the interdisciplinary culture research discussions with other IGS members, which improved me quite a lot and widened my horizon.

My sincere thanks to the funding gLINK (Sustainable Green Economies through Learning, Innovation, Networking and Knowledge Exchange) program under Erasmus Mundus and External Cooperation: Asia Region, the STIBET (Stipendien- und Betreuungsprogramm für ausländische Doktoranden und Postdoktoranden) project of DAAD, and the research cluster LogDynamics of the University of Bremen. Without support of these fundings, it would be hard to focus on my PhD study in one hundred percent.

I am very grateful to Prof. Dr.-Ing. Sergio Lucia for agreeing to review this thesis and giving me a lot of constructive comments and suggestions to improve this thesis. I also thank to Prof. Dr.-Ing. Michael Freitag to be part of my doctoral committee members and thank to his outstanding contributions on production control which give me lots of inspirations. I also thank to Prof. Dr. Rolf Drechsler for his insightful comments in my first-year oral examination.

I thank my friends – Quan Deng, Molin Wang, Yao Lin, Ziwang Wang and my colleagues from the group of Dynamics in Logistics (DiL) – Tobias Sprodowski, Kishwer Abdul Khaliq, Marcella Bernardo Papini, Haniyeh Dastyar and Elham Behmanesh, for accompanying me through so many wonderful unforgettable times. I also thank to Chris Ohlogge to read my thesis.

I would like to thank Bremer Institut für Produktion und Logistik GmbH (BIBA) for providing me with a very nice office environment. Also, I cannot forget the annual BIBA Christmas party where there was a lot of fun with much precious and sweet memories.

Last but not least, I am profoundly grateful to my family: my parents and my sister for their unconditional support and endless love. In particular, I would like to express my deepest gratitude to my wife Ping Liu for being with me in the past years, for her continuous patience and encouragement, incredible kindness and thoughtfulness, and selfless dedication and forever love.

# ABSTRACT

---

Due to globalization and competition in the manufacturing sector, companies are faced with challenges to produce a high variety of products with short life circles while maintaining a cost-effective level of production. Despite suffering from high work in process (WIP), long lead time and low reliability of delivery time especially in the presence of unpredictable events (e.g. demand fluctuations, machine breakdowns), the highly flexible job shop organization still plays an important role in today's manufacturing systems. In order to respond to volatile market demands quickly and flexibly, capacity adjustments represent one of the major proactive approaches to balance capacity and load for improvement of manufacturing control performance. Such adjustments are typically achieved by labor-oriented approaches via, e.g., flexible staffs and working time, which are comparably expensive and not a sustainable solution in a long term consideration.

In this thesis, we consider a machinery-based capacity adjustment via Reconfigurable Machine Tools (RMTs) to compensate for unpredictable events. Since WIP is essential for all key performance indicators in job shop systems, we propose to control the WIP level directly to track a planned or an acceptable level. To this end, RMTs are utilized to eliminate or periodically shift bottlenecks within the process. To include these tools effectively on the operational and tactical layers, we propose a complementing feedback approach using nonlinear model predictive control (NMPC) to identify the potential of RMTs for a better compliance with logistics objectives and a sustainable demand oriented capacity allocation. To this end, we build a discrete state space model of a job shop system with RMTs including nonlinearities, reconfiguration delays as well as disturbances. To effectively utilize RMTs concerning capacity adjustment, we formulate a set of reconfiguration rules to resolve the integer assignment of RMTs via the floor operator as well as the well known genetic and branch and bound algorithm.

Independent from the approach, stability of the closed-loop is of utmost importance, especially in the presence of demand fluctuation along with possibly frequent reconfigurations. To this end, we employ MPC schemes with and without terminal conditions for evaluation and comparison of their closed-loop performance. More specially, we first qualitatively analyze the asymptotic stability property in a continuous optimization case without considering integer constraints and reconfiguration delays with and without endpoint con-

straints, respectively. In the presence of integer constraints and reconfiguration delays, we first consider the terminal cost and terminal region method for practical stability where ultimate boundedness must be guaranteed in the designed terminal region. Yet, such terminal region is difficult to construct in the time-varying case. Also, due to the lack of explicit characterization on the practical region, the results may be over conservative. To this end, we quantitatively analyze practical stability in accordance with trajectory-dependent approach to estimate the degree of suboptimality and adapt the prediction horizon online. Furthermore, we analyze the influence of the reconfiguration delays on key performance indicator such as, e.g., setting time, during the transient phase, and on the steady-state performance via the size of the practical stability region. Due to the existence of reconfiguration delays, the cost value may increase locally even before reaching the practical stability area. To cover this issue, we propose to combine the adaptive NMPC approach with the principle of flexible Lyapunov functions that allow the value of cost functional to increase locally but with an average decrease in future steps. Last, a sensitivity analysis for the closed-loop performance is carried out including initial conditions, prediction horizon, reconfiguration delay and stage cost.

We demonstrate the effectiveness and plug-and-play availability of the proposed method for a four-workstation job-shop system and compare it to a state-of-the-art method. Utilizing numerical simulations, we show that the WIP can be practically asymptotically stabilized by usage of RMTs. We provide theoretical and practical instructions concerning implementation of the proposed method for manufacturers to help them improve operational reliability and shop floor control performance, and achieve a sustainable, effective and efficient capacity allocation policy.

# ZUSAMMENFASSUNG

---

Aufgrund der Globalisierung und des Wettbewerbs im verarbeitenden Gewerbe stehen Unternehmen vor der Herausforderung, eine große Vielfalt an Produkten mit kurzen Lebenszyklen unter Beibehaltung eines wirtschaftlichen Produktionsniveaus herzustellen. Trotz des Risikos hoher Auslastungen (WIP), langer Vorlaufzeiten und geringer Zuverlässigkeit in der Lieferzeit, insbesondere bei unvorhersehbaren Ereignissen (z.B. Bedarfsschwankungen, Maschinenausfälle), spielt die hochflexible Auftragsorganisation in den heutigen Fertigungssystemen nach wie vor eine wichtige Rolle. Um schnell und flexibel auf unbeständige Marktanforderungen reagieren zu können, stellen Kapazitätsanpassungen einen der proaktiven Ansätze dar, um Kapazität und Last auszugleichen und die Leistung der Fertigungssteuerung zu verbessern. Solche Anpassungen werden typischerweise durch arbeitsorientierte Ansätze erreicht, z.B. durch flexible Mitarbeiter und Arbeitszeiten, die vergleichsweise teuer und langfristig keine nachhaltige Lösung sind.

In dieser Arbeit betrachten wir eine maschinenbasierte Kapazitätsanpassung über Reconfigurable Machine Tools (RMTs), um diese unvorhersehbaren Ereignisse auszugleichen. Da WIP für alle Key Performance Indicators in Job-Shop-Systemen unerlässlich ist, schlagen wir vor, das WIP-Level direkt zu steuern, um ein geplantes oder akzeptables Level zu erreichen. Zu diesem Zweck werden RMTs eingesetzt, um Engpässe innerhalb des Prozesses zu beseitigen oder periodisch zu verschieben. Um diese Instrumente effektiv auf der operativen und taktischen Ebene einzubinden, schlagen wir einen ergänzenden Feedback-Ansatz mit Hilfe von Nonlinear Model Predictive Control (NMPC) vor, um das Potenzial von RMTs für eine bessere Einhaltung der logistischen Ziele und eine nachhaltige bedarfsorientierte Kapazitätzuweisung zu identifizieren. Zu diesem Zweck erstellen wir ein diskretes Zustandsraummodell eines Job-Shop-Systems mit RMTs einschließlich Nichtlinearitäten, Rekonfigurationsverzögerungen sowie Störungen. Um RMTs hinsichtlich der Kapazitätsanpassung effektiv zu nutzen, formulieren wir einen Satz von Rekonfigurationsregeln, um die ganzzahlige Zuordnung von RMTs über den Etagenbetreiber sowie den bekannten genetischen, verzweigten und gebundenen Algorithmus zu lösen.

Unabhängig vom Ansatz ist die Stabilität des geschlossenen Regelkreises von größter Bedeutung, insbesondere bei Nachfrageschwankungen und möglicherweise häufigen Rekonfigurationen. Zu diesem Zweck verwenden wir MPC-Schemata mit und ohne Endbedin-

gungen zur Bewertung und zum Vergleich der Closed-Loop-Performance. Insbesondere analysieren wir zunächst qualitativ die asymptotische Stabilitätseigenschaft in einem kontinuierlichen Optimierungsfall, ohne Ganzzahlbeschränkungen und Rekonfigurationsverzögerungen mit bzw. ohne Endpunktbeschränkungen zu berücksichtigen. Bei Vorliegen von ganzzahligen Beschränkungen und Rekonfigurationsverzögerungen betrachten wir zunächst die Endkosten- und Zielbereichsmethode für praktische Stabilität, bei der die Beschränkung in der entworfenen Terminalregion gewährleistet sein muss. Allerdings ist ein solcher Zielbereich im zeitvariablen Fall schwer zu konstruieren. Auch aufgrund des Fehlens einer expliziten Charakterisierung der praktischen Region können die Ergebnisse zu konservativ sein. Zu diesem Zweck analysieren wir quantitativ die praktische Stabilität nach dem trajektorieabhängigen Ansatz, um den Grad der Suboptimalität zu schätzen und den Vorhersagehorizont online anzupassen. Darüber hinaus analysieren wir den Einfluss der Rekonfigurationsverzögerung auf wichtige Leistungsindikatoren wie z.B. die Einstellzeit während der Übergangsphase und auf die stationäre Leistung über die Größe des praktischen Stabilitätsbereichs. Aufgrund von Rekonfigurationsverzögerungen kann der Kostenwert noch vor Erreichen des praktischen Stabilitätsbereichs lokal steigen. Um dieses Problem zu lösen, schlagen wir vor, den adaptiven NMPC-Ansatz mit dem Prinzip der flexiblen Lyapunov-Funktionen zu kombinieren, die es ermöglichen, bei einer durchschnittlichen Abnahme der Kosten in den zukünftigen Schritten den Wert der funktionalen Kosten lokal zu erhöhen. Schließlich wird eine Sensitivitätsanalyse für die Closed-Loop-Performance durchgeführt, die Ausgangsbedingungen, Vorhersagehorizont, Rekonfigurationsverzögerungen und aktuelle Kosten beinhaltet.

Wir demonstrieren die Effektivität und Plug-and-Play-Verfügbarkeit der vorgeschlagenen Methode für ein Job-Shop-System mit vier Arbeitsplätzen und vergleichen sie mit dem aktuellen Forschungsstand. Anhand numerischer Simulationen zeigen wir, dass das WIP durch den Einsatz von RMTs praktisch asymptotisch stabilisiert werden kann. Wir geben theoretische und praktische Anweisungen zur Umsetzung der vorgeschlagenen Methode für Hersteller, um ihnen zu helfen, die Betriebssicherheit und die Leistung der Fertigungssteuerung zu verbessern und eine nachhaltige, effektive und effiziente Strategie der Kapazitätszuweisung zu erreichen.



# TABLE OF CONTENTS

---

<b>ACKNOWLEDGEMENT</b>	<b>III</b>
<b>ABSTRACT</b>	<b>V</b>
<b>ZUSAMMENFASSUNG</b>	<b>VII</b>
<b>LIST OF TABLES</b>	<b>XII</b>
<b>LIST OF FIGURES</b>	<b>XIII</b>
<b>LIST OF ABBREVIATIONS AND SYMBOLS</b>	<b>XV</b>
<b>1. INTRODUCTION</b>	<b>1</b>
1.1. Motivation . . . . .	1
1.2. Research question . . . . .	2
1.3. Research objectives . . . . .	4
1.4. Outline of thesis and research contribution . . . . .	4
<b>2. THE STATE OF THE ART</b>	<b>7</b>
2.1. Types of manufacturing system . . . . .	7
2.2. Modeling and control of manufacturing process . . . . .	8
2.2.1. Non-control theoretic approaches . . . . .	9
2.2.2. Control theoretic approaches . . . . .	9
2.3. Logistics objectives in a manufacturing control model . . . . .	10
2.3.1. Work in process . . . . .	11
2.4. Approaches of capacity control . . . . .	12
2.4.1. Capacity control via control-theoretical approaches . . . . .	12
2.4.2. Reconfigurable manufacturing systems . . . . .	14
2.5. Reconfigurable machine tools . . . . .	16
2.5.1. Architecture design . . . . .	17
2.5.2. Configuration design with optimisation . . . . .	18
2.5.3. System integration of control . . . . .	18
2.6. Capacity control with RMTs in job shop systems . . . . .	19
2.7. Summary . . . . .	20

<b>3. DEFINITIONS AND PRELIMINARIES</b>	<b>21</b>
3.1. Basic concepts of discrete time control systems . . . . .	22
3.2. Nonlinear programming . . . . .	22
3.2.1. Mixed-integer nonlinear programming . . . . .	25
3.3. Stability of discrete time control systems . . . . .	26
3.3.1. Asymptotic stability . . . . .	27
3.3.2. Practical asymptotic stability . . . . .	28
3.4. Types of controller . . . . .	29
3.4.1. Proportional integral derivative . . . . .	30
3.4.2. Linear quadratic regulator . . . . .	30
3.4.3. Nonlinear model predictive control . . . . .	31
3.5. Stability and feasibility of nonlinear model predictive control . . . . .	34
3.5.1. Terminal endpoint constraints . . . . .	34
3.5.2. Terminal cost and terminal region . . . . .	35
3.5.3. Without terminal conditions . . . . .	36
3.6. Summary . . . . .	39
<b>4. MODELING OF A JOB SHOP SYSTEM WITH RMTs FOR WIP CONTROL</b>	<b>40</b>
4.1. Assumptions for establishing model . . . . .	41
4.2. Discrete state space model for WIP control . . . . .	41
4.3. A general reconfiguration rule . . . . .	44
4.4. Summary . . . . .	46
<b>5. OPTIMIZATION AND PLANNING TO MAINTAIN WIP VIA RMTs</b>	<b>48</b>
5.1. Integer programming within model predictive control . . . . .	48
5.1.1. Branch and bound . . . . .	50
5.1.2. Genetic algorithm . . . . .	50
5.1.3. Strategies for reducing computational cost . . . . .	51
5.2. Reconfiguration rule within integer predictive control setting . . . . .	53
5.2.1. Reconfiguration rule based on priority of WIP tracking . . . . .	53
5.2.2. Counting times of reconfiguration for production cost . . . . .	55
5.2.3. Possible unnecessary triggered reconfiguration . . . . .	57
5.2.4. Prediction horizon and step of reconfiguration delay . . . . .	58
5.3. Summary . . . . .	58
<b>6. STABILITY OF A JOB SHOP SYSTEM WITH RMTs FOR WIP CONTROL</b>	<b>60</b>
6.1. Asymptotic stability without integer constraints and delays . . . . .	61
6.1.1. Terminal endpoint constraints . . . . .	62
6.1.2. Unconstrained MPC scheme . . . . .	66
6.2. Practical stability with integer constraints and reconfiguration delays . . . . .	69
6.2.1. Terminal cost and terminal region . . . . .	70
6.2.2. Unconstrained MPC scheme . . . . .	74
6.2.3. Estimate of $\varepsilon$ . . . . .	78
6.2.4. Incomplete optimization with NP-hard for stability . . . . .	78

6.2.5.	Analytical approximate estimate of practical region . . . . .	80
6.2.6.	Local instability with reconfiguration delays . . . . .	82
6.2.6.1.	Flexible Lyapunov function . . . . .	82
6.2.6.2.	Relaxed and flexible Lyapunov inequality . . . . .	84
6.2.6.3.	Adaptive NMPC with flexible Lyapunov function . . . . .	87
6.3.	Summary . . . . .	88
<b>7.</b>	<b>NUMERICAL RESULTS</b>	<b>90</b>
7.1.	Asymptotic stability without integer constraints and delays . . . . .	91
7.1.1.	Comparison between standard PID and MPC . . . . .	92
7.1.2.	Sensitivity analysis . . . . .	94
7.1.2.1.	Initial conditions . . . . .	94
7.1.2.2.	Dynamic flow control . . . . .	95
7.1.2.3.	Prediction horizon . . . . .	96
7.2.	Practical stability with integer constraints and reconfiguration delays . . . . .	99
7.2.1.	Comparison of different integer control strategies . . . . .	100
7.2.2.	Practical stability via an a posteriori suboptimality estimate . . . . .	104
7.2.3.	Analysis of reconfiguration delays on closed-loop performance . . . . .	105
7.2.4.	Sensitivity analysis . . . . .	108
7.2.4.1.	Initial conditions . . . . .	108
7.2.4.2.	Dynamic flow control . . . . .	109
7.2.4.3.	Prediction horizon . . . . .	110
7.3.	Summary . . . . .	114
<b>8.</b>	<b>CONCLUSION AND OUTLOOK</b>	<b>117</b>
8.1.	Conclusion . . . . .	117
8.2.	Outlook . . . . .	119
	<b>REFERENCES</b>	<b>121</b>
<b>A.</b>	<b>APPENDIX</b>	<b>141</b>
A.1.	Terminal endpoint constraints . . . . .	142
A.2.	Proof for Proposition 3.32 . . . . .	143
A.3.	Linear quadratic regulator . . . . .	144
A.4.	Induction process for (6.25)-(6.27) . . . . .	145
<b>B.</b>	<b>LIST OF PUBLICATIONS</b>	<b>146</b>

## LIST OF TABLES

---

2.1.	RMT combines features of dedicated and CNC machines [1] . . . . .	16
7.1.	The variables definition in the job shop system . . . . .	91
7.2.	Comparison between standard PID and MPC with different performance indicators . . . . .	93
7.3.	Process flow of each product [2] . . . . .	96
7.4.	Comparison between standard PID and different integer control strategies based on MPC . . . . .	100
7.5.	A posterior estimate of degree of suboptimality without reconfiguration delay in case of practical asymptotic stability . . . . .	105
7.6.	A posterior estimate of degree of suboptimality with reconfiguration delay in case of practical asymptotic stability . . . . .	106

# LIST OF FIGURES

---

1.1.	Types of demand fluctuation [3] . . . . .	2
1.2.	Production planning, scheduling, and control hierarchy . . . . .	3
1.3.	Main research structure of thesis . . . . .	5
2.1.	Interpretation of control theory applied in manufacturing process . . . . .	10
2.2.	A manufacturing control model based on [4] . . . . .	11
2.3.	Structure of a reconfigurable machine system . . . . .	14
2.4.	Workflow of a manufacturing process with RMTs [5] . . . . .	17
3.1.	An illustrated example of convex optimization . . . . .	26
3.2.	An illustrated example of non-convex optimization . . . . .	26
3.3.	$\mathcal{K}$ function . . . . .	27
3.4.	$\mathcal{L}$ function . . . . .	27
3.5.	Asymptotic stability (3.17) . . . . .	29
3.6.	Practical asymptotic stability (3.20) . . . . .	29
3.7.	Principle of Model predictive control . . . . .	32
3.8.	Difference between open-loop and closed-loop trajectory . . . . .	33
4.1.	$j$ th workstation in multi-workstation production system . . . . .	42
4.2.	Illustrate example of reconfigurable rule [6] . . . . .	45
5.1.	Illustration of chromosome coding of GA within MPC setting . . . . .	51
5.2.	Flow chart of MPC-GA . . . . .	52
5.3.	Comparison of adjacent states for the triggered reconfiguration . . . . .	55
5.4.	Triggered reconfiguration based on WIP priority with possibly increased cost	57
6.1.	Minimal and maximum invariant sets within terminal region . . . . .	73
6.2.	Relationship among $\mathcal{B}_{\delta_s}$ , $\mathcal{B}_{\delta_m}$ and $\mathbb{X}_F$ . . . . .	73
6.3.	Relationship between terminal region and practical stability region . . . . .	74
6.4.	Illustration for Proposition 6.8 . . . . .	75
6.5.	Local increase before reaching $\varepsilon$ . . . . .	75
6.6.	One step increase but with an average decrease in future two steps . . . . .	83

6.7. One step increase but with a small penalty on the decreased costs . . . . .	83
6.8. Prolongation of $k$ satisfies flexible Lyapunov equation . . . . .	84
6.9. Prolongation of $k$ may render flexible Lyapunov function fail . . . . .	84
6.10. General case for determination of flexible Lyapunov function within $k$ step	85
7.1. Job shop manufacturing systems with RMTs . . . . .	91
7.2. Variation of $WIP$ . . . . .	93
7.3. Variation of usage of RMTs . . . . .	93
7.4. $WIP$ levels for different initial conditions . . . . .	95
7.5. Variation of $WIP_1$ under different order release rate . . . . .	96
7.6. Variations of $WIP$ in the dynamic flow . . . . .	96
7.7. Comparison between with and without terminal endpoint constraints . .	97
7.8. Different control strategies for controlling $WIP$ by integer assignment of RMTs . . . . .	99
7.9. Comparison of $WIP$ level in each workstation by different methods with $N = 2$ . . . . .	101
7.10. Difference between current $WIP$ value with planned value for each work- station . . . . .	102
7.11. Optimal value function for $N = 2$ . . . . .	102
7.12. Number of RMTs by B&B associated with MPC for each workstation . . .	103
7.13. Variation of $\alpha$ without delay . . . . .	104
7.14. Variation of $\alpha$ with delay . . . . .	104
7.15. Comparison of usage of RMTs with and without reconfiguration delay . .	107
7.16. Comparison of $WIP$ tracking with and without delay . . . . .	107
7.17. Estimate under reconfiguration delays but with $\varepsilon = 2.8$ . . . . .	107
7.18. $WIP$ levels for different initial conditions in practical case . . . . .	108
7.19. Variation of $\alpha$ for different situations . . . . .	109
7.20. Number of produced products and $WIP$ in the dynamic flow . . . . .	110
7.21. Number of used RMTs in the dynamic flow . . . . .	110
7.22. Comparison of optimal value function with and without delay . . . . .	111
7.23. Computing time with $N = 2$ with flexible Lyapunov function . . . . .	111
7.24. Comparison of $WIP$ tracking With different $N$ . . . . .	112
7.25. Variation of optimal value function $V_N$ with $N = 3$ and $N = 4$ . . . . .	112
7.26. Computing time with $N = 3$ . . . . .	113
7.27. Computing time with $N = 4$ . . . . .	113
7.28. Comparison of $V_2(x(n))$ under different stage costs . . . . .	114
7.29. Comparison of $\ell(\cdot, \kappa_2(x(n)))$ under different stage costs . . . . .	114
7.30. Variation of $WIP$ under different stage costs . . . . .	114
7.31. Usage of RMTs under different stage costs . . . . .	114

# LIST OF ABBREVIATIONS AND SYMBOLS

---

## Abbreviations

$MPC_1$	Model predictive control with floor operator
$MPC_2$	Model predictive control with brach and bound
$MPC_3$	Model predictive control with genetic algorithm
$\mu(WIP)$	Mean value of WIP
$\sigma(WIP)$	Standard Deviation of WIP
$\mu(RMT)$	Mean value of usage of RMTs
B&B	Branch and Bound
BIBO	Bounded Input Bounded Output
DES	Discrete Event Simulation
DMT	Dedicated Machine Tool
DP	Dynamic Programming
FIFO	First in First Out
FMT	Flexible Machine Tool
GA	Genetic Algorithm
IAE	Integral Absolute Error
IAU	Integral Absolute Control
IoT	Internet of Things
IP	Interior-Point
ISE	Integral Square Error

ISS	Input to State Stability
ISU	Integral Square Control
KKT	Karush-Kuhn-Tucker
LQR	Linear Quadratic Regulator
MIMO	Multi Input Multi output
MPC	Model Predictive Control
MPC-B&B	Model predictive control with branch and bound
MPC-floor	Model predictive control with floor operator
MPC-GA	Model predictive control with genetic algorithm
NMPC	Nonlinear Model Predictive Control
NP	Non-Polynomial
PID	Proportional Integral Derivative
PID-floor	PID with floor operator
PMP	Pontryagins Maximum Principle
PSO	Particle Swarm Optimization
RFID	Radio Frequency Identification
RMS	Reconfigurable Manufacturing System
RMT	Reconfigurable Machine Tool
SD	System Dynamics
SISO	Single Input Single output
SOSC	Second Order Sufficient Conditions
SQP	Sequential Quadratic Programming
WIP	Work in Process

**Symbols**

$x$	State-work in process
-----	-----------------------



$x^+$	Successor state
$u$	Control input–number of RMTs
$x_0$	Initial work in process
$x^*$	Equilibrium point
$u^*$	Equilibrium control
$m$	Total number of RMTs in job shop system
$r^{RMT}$	Production rate of RMT
$r^{DMT}$	Production rate of DMT
$P$	Probability matrix of product flow in job shop system
$p$	Number of workstation
$j$	Index of workstation
$I_j$	Input rate to $j$ workstation
$O_j$	Output rate from $j$ workstation
$p_{ij}$	Probability from workstation $i$ to workstation $j$
$d$	External input rate to system
$i_{0j}$	External input rate to $j$ workstation
$r$	Reference
$y$	Output
$e$	Difference between reference with output
$r_d$	Reconfiguration delay
$\xi_{i+r}$	Disturbance caused by discretization of control input
$\mathbb{X}$	State constraints
$\mathbb{U}$	Control constraints
$T$	Control interval
$N$	Prediction horizon

$N_0$	Additional parameter used together with an a prior estimate method
$N_{min}$	“Minimal” prediction horizon considering reconfiguration delay
$\ell$	Stage cost
$\lambda$	Weighting coefficient on control input
$\Delta u(k)$	Penalty for the reconfiguration of RMTs
$k$	Index of prediction horizon
$J_N$	Cost functional
$u^*(\cdot)$	An optimal control sequence
$x^*(\cdot)$	An optimal open-loop control sequence
$\kappa_N$	Feedback control law
$V_N$	Optimal value function
$\nu$	Parameters in flexible Lyapunov function
$V_\infty$	Infinite original optimal function value
$V_\infty^{\kappa_N}$	Infinite feedback optimal function value
$\sigma$	Decay rate
$C$	Overshoot
$\gamma$	Auxilliary characteristic for degree of suboptimality
$\varepsilon$	Truncated constant for practical stability
$\gamma_N$	Auxilliary characteristic for degree of suboptimality depending on $N$
$\alpha$	Local estimate of degree of suboptimality
$\alpha_N$	Estimate of degree of suboptimality depending on $N$
$\alpha_{min}$	Closed-loop estimate of degree of suboptimality
$\bar{\alpha}$	A given stability suboptimality estimate
$\delta$	Practical stability region
$F$	Terminal cost

$u_F$	Terminal control
$\mathbb{X}_F$	Terminal region
$x(N)$	Terminal state
$K_p, K_i, K_d$	Parameters of PID
$K$	Feedback gain of LQR

# 1

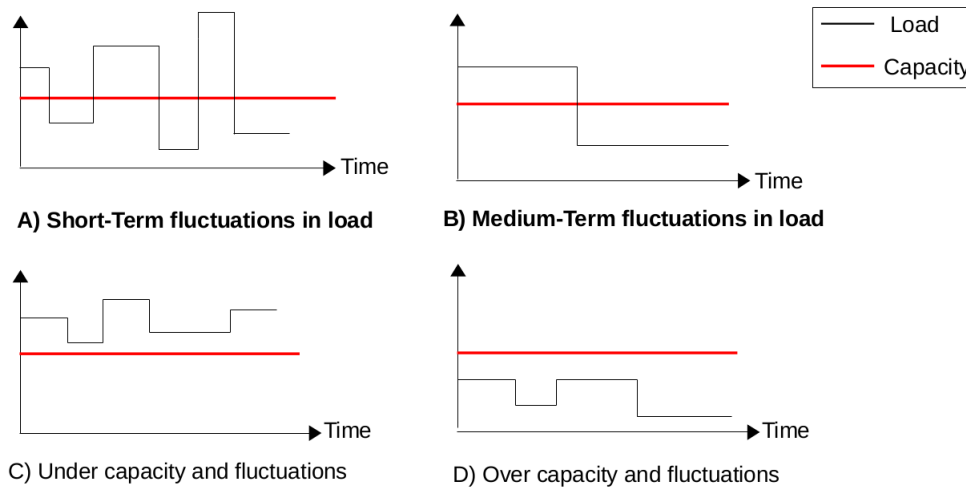
## INTRODUCTION

### 1.1. MOTIVATION

Nowadays, manufacturing companies are confronted with the challenges to rapidly react to dynamical, unpredictable market changes driven by global competition while maintaining cost-efficiency, high quality as well as a diversity level of production. That leads to a significant increase in dynamics and complexity [7]. Manufacturers as well as researchers agree that the ability of a company to cope with market demand fluctuation is primarily determined by capacity. Generally, the capacity in the production can be divided into three aspects: the capacity of time, the capacity of space, and the capacity of manufacturing [8]. In particular, the last one is defined as the maximum rate of production and the ability to yield production. Due to introduction of new kinds of products, machine breakdowns, unreliable deliveries or rush orders, the capacity is affected and changed over time [9, 10], which renders manufacturing processes to be more complex and dynamic. Typically, the required capacity is estimated during the planning and structuring phase of the design of a manufacturing site and statically fixed on the strategic long term level. Unfortunately, only rough data regarding future demands can be used, which typically leads to functional shortages or unused capacities.

To deal with the resulting performance degradation and to achieve a good shop floor performance, capacity adjustment is one of the major effective tools in industrial manufacturing and logistic processes to react on fluctuations in demand [11]. Even small modifications during a high load period may improve performance significantly [12]. Generally, an adjustment of capacities is done by purchasing new equipment or reallocating work force or utilities to temporarily increase/decrease the capacity, e.g. via flexible operators or extra hours, which are not long-term sustainable and are expensive especially in the western countries where the labor cost is high [13, 14]. In this thesis, we propose a machinery-based approach to adjust capacity via reconfigurable machine tools (RMTs), i.e. we adapt the number of RMTs assigned to specific tasks on the shop floor. As an additional degree

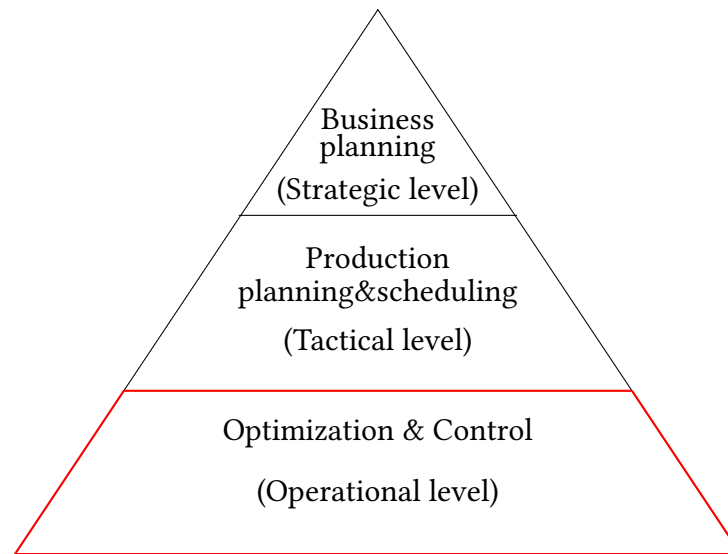
of freedom with respect to capacity adjustment, RMTs exhibit quite competitive features in the industrial manufacturing system relying on its strong ability to flexibly change capacity and functionality. This reconfigurability and flexibility can be exploited best within manufacturing systems with high product diversity at small lot sizes, which is the case, e.g., for job shop systems [11, 15].



**Figure 1.1.: Types of demand fluctuation [3]**

## 1.2. RESEARCH QUESTION

Due to the frequent changes in demand, cf., Figure 1.1 and possible machine failures, the job shop production processes exhibit highly dynamic and turbulent behavior, typically suffer from high work in process (WIP) and occurrence of bottlenecks which seriously disturb the normal operation. As WIP is essential for all key performance indicators, we propose to control the WIP level directly to the planned or an acceptable level through utilizing capacity adjustment via RMTs to eliminate or periodically shift bottlenecks within the process [4]. As an enabler for capacity adjustment through changing modulars to perform different operational requirements, RMTs can be allocated to workstations to balance capacity and load in case of bottleneck. However, due to the inherent reconfiguration delay including the installation time, reconfiguration time and ramp-up time, *reconfigurable control with respect to RMTs is not just a simple machine allocation but with non-symmetric*. In the reconfiguration phase, the number of RMTs with an increase requirement does not change, in the contrast, the capacity of already reconfigured machines and the whole system's capacity immediately decrease which impacts on the system performance, extends the throughput time or even may destabilize the system. Controlling such reconfiguration becomes a building block on dependability of feedback control [16].



**Figure 1.2.:** Production planning, scheduling, and control hierarchy

As job shop systems typically are coupled constrained multi-input multi-output (MIMO) processes with disturbance, they are intractable for traditional proportional integral derivative (PID) controllers, which have been successfully applied to single-input single-output (SISO) systems [13, 17]. Also, the capacities of all shop floors shall be considered at once to avoid shortages and unused capacities of different shop floors. Therefore, a central control instance is appropriate, especially with prediction functionality.

Faced with complex and dynamic manufacturing environment, keeping the manufacturing process as stable as possible is of utmost importance for manufacturers to facilitate increased production, lower energy costs and achieve a high reliability of delivery time [18, 19, 20]. Typically, the basic stability refers to general predictability and consistent availability in terms of manpower, machines, materials and methods, which is also called 4Ms. In this thesis, we focus on the latter three elements on the operational layer where the manufacturing process control is integrated in an enterprise planning and control hierarchy [21, 22], cf., Figure. 1.2. As a result, we cannot determine the order release rates to any of the workstations. That is different from workload control in which the WIP can be controlled by means of changing order release rate (materials) to the shop floor [23, 24]. Instead, we employ control (methods) to internally dynamically adjust capacity via RMTs (machines) to maintain WIP in the manufacturing process for a given order release rate, which is a prerequisites for achieving a good shop floor performance. Through the possible minimum reconfiguration effort under constraints, our aim is to *guarantee the stability of the closed-loop system with possible frequent reconfiguration, and steer the system state (WIP) to the desired levels after disturbance* (e.g. rush order, demand fluctuation in quantity or product variety). That is, we not only guarantee the steady-state performance but also pursuit the transient performance.

Considering all the above factors, the main research question along with several sub-questions are given:

*Given the operational layer, how can the WIP be asymptotically stabilized by usage of RMTs?*

- Q 1.** How can RMTs be integrated in a WIP based model of a job shop system?
- Q 2.** How can the model be complemented by operational key performance indicators (KPIs) to allow for predictive planning?
- Q 3.** How to guarantee the asymptotic stability?

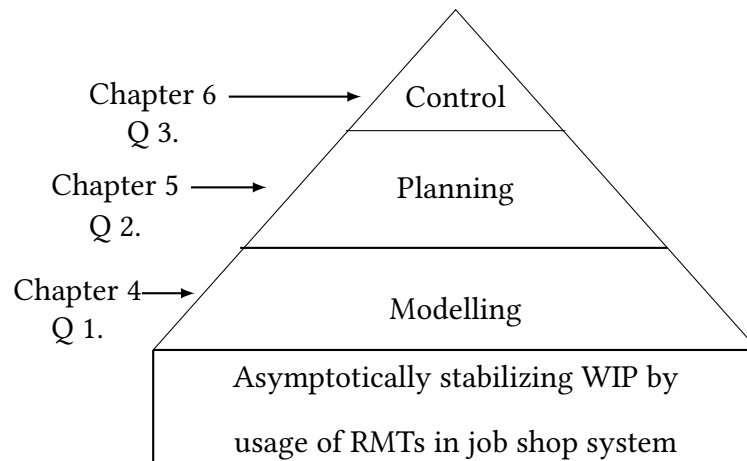
### **1.3. RESEARCH OBJECTIVES**

The research objectives of this project are summarized as follows:

- Improve the potential of current processes to flexibly react on these fluctuations in the medium and short term via RMTs on the operational layer to eliminate or shift bottleneck via capacity adjustment.
- Develop and formulate a set of reconfiguration rules for the determination of the triggered RMTs in accordance with different KPIs.
- Develop a closed-loop system with RMTs for capacity adjustment to achieve the damping of negative effects of external demand fluctuations for a smooth operation on the shop floor.
- Improve the operational reliability of the logistic efficiency through guaranteeing stability of dynamic manufacturing process with the possible but necessary frequent reconfiguration.

### **1.4. OUTLINE OF THESIS AND RESEARCH CONTRIBUTION**

The research aims at a systematic analysis of the potential of RMTs and the development of control strategies utilizing the dynamics of the manufacturing and logistic process for a sustainable demand oriented capacity allocation. The development of control strategies will be parameterized and analyzed in a simulation model. The results will allow us to refine the individual selection and parameterization of control strategies for the specific influence on logistics key performance indicators. The main research structure of thesis is sketched with Figure 1.3, and the detailed structure of this thesis is given as follows:



**Figure 1.3.:** Main research structure of thesis

- Chapter 2 is driven by system theory with literature review which includes the modelling methods of manufacturing process, approaches of capacity control as well as development of reconfigurable manufacturing system and RMTs.
- Chapter 3 is the brief introduction about mathematical theory including stability, control, and optimization. As a theoretic foundation, it is indispensable for the later implementation of the proposed method.
- In Chapter 4, we integrate the new class of RMTs into a shop floor for capacity adjustment and built up a discrete state space model under reasonable assumptions. In addition, we introduce a general reconfiguration rule for the determination of the triggered RMTs, which is applicable to any types of controller, especially for those without prediction functionality.
- In Chapter 5, we propose a real time applicable closed loop control law utilizing model predictive control (MPC) with RMTs to maintain WIP in the presence of demand fluctuations. To resolve the integer assignment of RMTs, we propose three different approaches - the straightforward way via floor operator, deterministic branch and bound (B&B), and stochastic genetic algorithm (GA). Relying on the capability of dealing with the relationship between prediction horizon with reconfiguration delay via MPC, we formulate a tailored reconfiguration rule concerning WIP tracking based on a priority strategy for the determination of the triggered RMTs in the framework of MPC setting. Also, we provide an algorithm with respect to counting times of reconfiguration whiling taking reconfiguration cost into account. Last, some ideas are given to reduce the computational burden.
- In Chapter 6, we systematically analyze the stability in both of qualitative and quantitative ways via two categories of stabilizing MPC controllers. We first qualitatively



analyze the asymptotic stability property in case of continuous variables with and without terminal endpoint constraints, respectively. Further, in the presence of integer constraints and reconfiguration delays, we consider the terminal cost and terminal region method for practical stability where ultimate boundedness must be guaranteed in the designed terminal region which is difficult to construct in the time-varying case. Then we quantitatively analyze practical stability in accordance with the trajectory-dependent approach to online estimate the degree of suboptimality in both a posterior and a priori ways. Since reconfiguration delays may lead to local instability, we first approximately analyze the closed-loop trajectory and show that the closed-loop trajectory would converge to the practical stability region. Then we propose to combine the adaptive NMPC approach with the principle of flexible Lyapunov functions that allow the value of cost functional to increase locally but with an average decrease in future steps.

- Chapter 7 provides numerical results including validation of the proposed method in Chapter 6, interpretation of simulation results and sensitivity analysis. In case of continuous usage of RMTs, we compare the closed-loop performance between MPC with terminal endpoint constraints with PID. Later, considering integer constraints and reconfiguration delays, we compare PID-floor (i.e., PID with floor operator), MPC-floor (i.e., MPC with floor operator), MPC-B&B (i.e., MPC with branch and bound) and MPC-GA (i.e., MPC with genetic algorithm) concerning tracking performance, computational time, etc. Furthermore, a posterior suboptimality estimate is computed to show practical stability. The influence derived by the reconfiguration delay is quantitatively analyzed in terms of the transient performance (e.g. settling time) and the size of practical region in the steady-state phase. The sensitivity analysis is carried out including initial conditions, prediction horizon, dynamic flow control. Last, through a simple case, we show how the flexible Lyapunov function works in case of local instability.
- In Chapter 8, we conclude our work and provide instructions concerning implementation of the proposed method for manufacturers to help them improve operational reliability and shop floor control performance, and achieve a sustainable, effective and efficient capacity allocation policy. After our conclusion, an outlook is presented for further possible research directions.

# 2

## THE STATE OF THE ART

This chapter presents the review of related references. Section 2.1 is a brief introduction on types of manufacturing system. In Section 2.2, several modelling approaches of manufacturing processes are reviewed. Then, a manufacturing control model incorporating logistic objectives (e.g., WIP) is presented in Section 2.3. To effectively regulate the WIP, capacity control from the view of output control is introduced, which includes control theoretic approaches in Section 2.4.1 as well as the development of reconfigurable manufacturing systems (Section 2.4.2) along with RMTs for implementing flexible capacity adjustment (Section 2.5).

### 2.1. TYPES OF MANUFACTURING SYSTEM

Manufacturing systems are considered as the industrial backbone of countries, especially in the context of Industry 4.0. The evolution of manufacturing systems along with future enabling directions was comprehensively reviewed in [25]. Typically, manufacturing systems include five general categories, i.e., project shop, flow line, continuous manufacturing system, batch process and job shop. The position of products manufactured in the project shops are fixed due to their size or/weight during the manufacturing process, e.g., aircraft and ship building industries. Its production scale is the lowest one among all types of manufacturing systems. A flow line is used to produce high volumes of products at high production rates and low costs. Its WIP is low and manufacturing lead time is short. Since the machines are designed for specific operations, only the specified products can be produced. In a batch process, the products are manufactured sequentially over a series of workstations and different batches can be produced. Continuous manufacturing systems are analogous to flow lines which produce a low variety but large volume of products, yet the raw material are liquids, powders, gases, etc.

To meet individual requirements of products, a job shop production system is one effective organization to manufacture a variety of products of small lot sizes at a reasonable

cost by utilizing the best suitable technology for every job [26]. It is organized via multiple workstations, which typically consist of machines offering identical functionality. Most researches concentrated on studying job shop scheduling problems by using optimization methods including mathematical theory or meta-heuristic methods to improve on certain performance criteria, e.g., minimizing the total completion time for all jobs or minimizing the makespan [27, 28, 29, 30, 31]. Yet, the performance during the optimization period cannot be guaranteed if the system is subject to unpredictable events during operations (e.g., demand fluctuation, machine breakdown), and the exact schedule may often be infeasible [24]. Additionally, these unexpected disturbances may render job shop systems to be more complex and dynamic as they may lead to bottlenecks of one machine or workstation or results in high WIP, long lead time, low capacity utilization and low reliability of due date [12]. In turn, these issues may results in instability of the system and impose high pressure on the shop floor control.

In order to shorten lead time and improve the reliability of delivery time, one could release orders earlier, which is intended to increase output rates. Doing so, however, may destabilize the system, cause unbounded growth of WIP, additional inventory cost, requirement of large storage space and even loss of consumers [32]. Alternatively, manufacturers can offer discounts for consumers as compensation for possible delivery delay. This reactive solution is ineffective and may be infeasible if the due date is a hard constraint. Thus, a real time control to compensate the above issues is indispensable. In order to alleviate the resulting performance degradation and to achieve a good performance of the shop floor control, the critical task is to match time-phased capacity requirements with the available capacities which can be achieved via e.g., capacity adjustment [11].

## **2.2. MODELING AND CONTROL OF MANUFACTURING PROCESS**

Typically, job shop systems are utilized within dynamic environments and are subject to internal and external disturbances [19]. External ones, such as rush orders may cause the product variety and quantity to be changed. Internal ones, on the other hand, such as machine breakdowns, production rates, etc., may influence downstream production stages. Additionally, dynamic job shops are complex in design as future conditions cannot be anticipated given historical data. Hence, it is important to predict and control the dynamic behavior to maintain the performance criterion at each decision point and keep the process alive. In operational planning and control for manufacturing systems, several different approaches are known to predict and improve the dynamic behavior of such systems, which includes queuing network theory, discrete event simulation (DES), system dynamics (SD) and control theoretic approaches [17].

### 2.2.1. NON-CONTROL THEORETIC APPROACHES

Queuing network theory is a common approach to analyze and evaluate the performance of production systems and represents a mathematical abstraction from real production systems. However, this approach generally is typically loaded at the scheduling layer for the analysis of the steady-state behavior of a production system, and is not applicable to study the detailed behavior of the dynamics [33]. In general, most real production systems show a multivariable property and negative feedback loops under transient conditions, which are too complex to analytically derive a mathematical formulation using queuing network analysis.

Based on discrete inherent properties, DES is one of the most commonly applied techniques for analyzing and understanding dynamic production systems, and can be seen as a representation of real system [34]. The process flow and the on-screen movement in DES can be valuable tools in providing increased understanding of a process [35]. It has the capability to perform a detailed analysis of a specific system or linear process and change the state variables instantaneously at separate points in time [36]. Yet, DES entails rather high modeling efforts and is limited to time frames. It is not typically capable of predicting the dynamic behavior under transient conditions and cannot be utilized to design strategies to modify or mitigate the negative influence of perturbations without a closed loop.

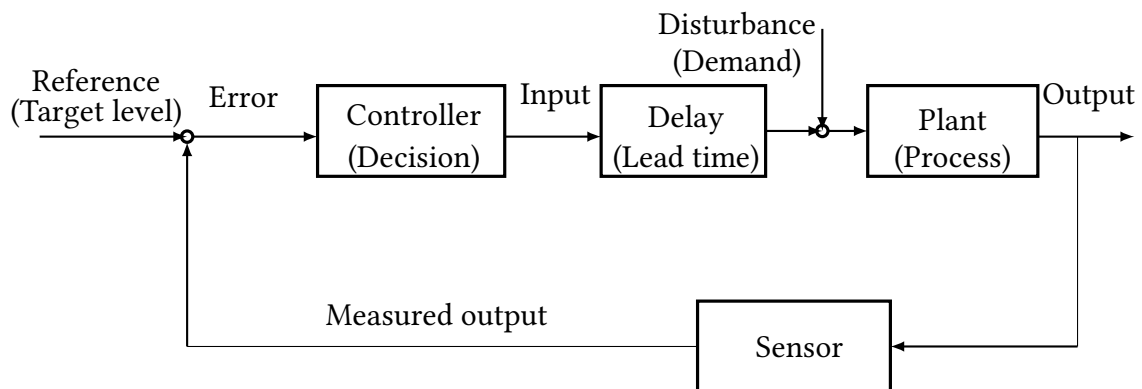
SD is an alternative methodology, which allows decision-makers to model a variety of scenarios and observe how the system could perform under different conditions. Particular, capturing the information flow and offers of feedback for dynamic processes can be analyzed by SD [37]. A major point of system dynamics is that the structure of a system can be modified to improve performance. In [38], the authors developed a comprehensive dynamic model for real-time production and planning control system in an arbitrary job-shop via SD. Yet, this technique typically does not provide analytical solutions and then cannot identify the transient dynamic behavior. A good transient performance is very important in any dynamic turbulent environment and a sensitivity analysis is always carried out in the transient phase (e.g., setting time) besides the steady-state [39]. Additionally, using DES and SD approaches, a heuristic or a trial and error method has to be applied repeatedly to derive a good solution, which represents an open loop scheme only. Unfortunately, these approaches fail to address the permanent and stochastic nature of the problem at hand.

### 2.2.2. CONTROL THEORETIC APPROACHES

Control theory provides various tools to analyze control process dynamics and has been widely applied in mechanical systems or chemical processes for many years. Yet, only a couple of researchers studied on the usage of feedback control in manufacturing systems. One major reason may be that the lack of sufficient data for feedback information involved in modeling production systems – hence the outcomes had limited applicability

in production control. However, the development of Radio Frequency Identification (RFID) technology and other sensor systems in the framework of Internet of things (IoT) render closed-loop control to be practically applicable [40, 41, 42].

Control theory methods in production were recognized in [43], where the authors reviewed respective applications for with the production-inventory problem. They indicated that there is a big potential in manufacturing for predicting demand and providing the required feedback information, especially for deterministic models at an aggregated level. A comprehensive review of control theory on production lines was presented in [44]. It included that classical control theory such as block diagram algebra, bode plots, transfer function,  $z$  transform and optimal control that use Dynamic Programming and Pontryagin Maximum Principle, etc. [45, 46]. In [47], the authors summarized the application of control theory in the dynamic modeling and control of supply chain systems. The control theoretical modeling of the transient behavior in the context of production planning and control was surveyed in [17]. The recent study of control on the field of inventory, manufacturing and queuing systems has been comprehensively reviewed in [48]. A quiet recent survey of control theory application on modern production and logistics systems, supply chains and Industry 4.0 was conducted in [49]. With the advanced physical equipment and maturely developed control theory methods, there are many possibilities and potentials to apply appropriate control techniques in modern manufacturing, cf., Figure 2.1.



**Figure 2.1.:** Interpretation of control theory applied in manufacturing process

## 2.3. LOGISTICS OBJECTIVES IN A MANUFACTURING CONTROL MODEL

In order to help practitioners to better understand the relationship between manufacturing control with logistics objectives, a manufacturing control model was built incorporating four performance indicators – low WIP inventories, low throughput time, high machine utilisation and high delivery reliability, cf., Figure 2.2. *The aim of control theory method on*

manufacturing control is to regulate the controlled variables (e.g., WIP) to the desired levels by the defined actuating variables even if disturbances occur [4]. As WIP is a key and essential variable in the manufacturing processes control, which indirectly influences other indicators, we focus on controlling WIP directly.

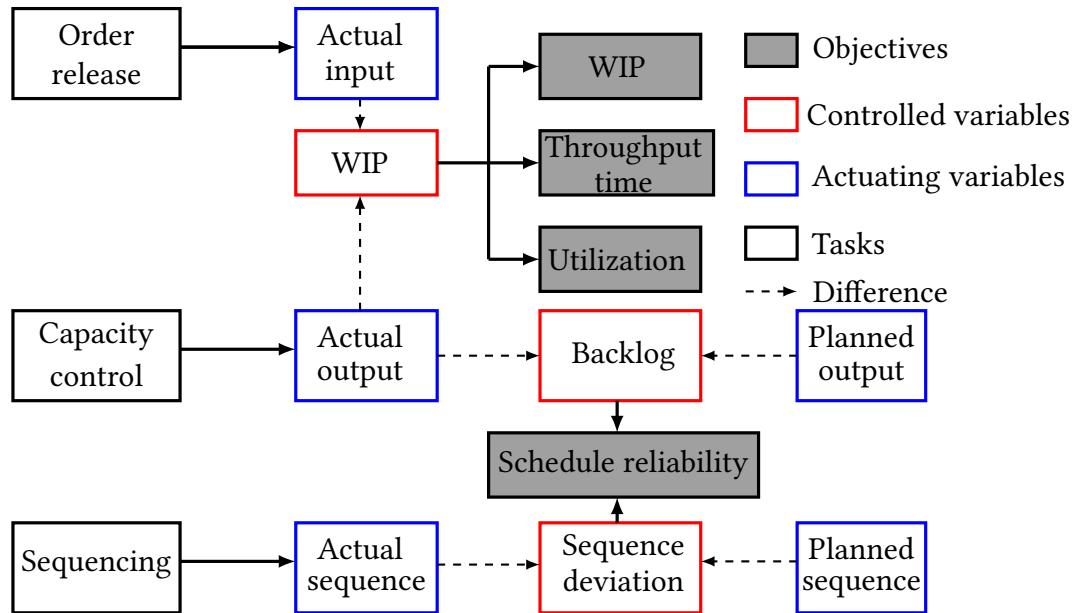


Figure 2.2.: A manufacturing control model based on [4]

### 2.3.1. WORK IN PROCESS

Work in process (WIP) in manufacturing refers to the orders already being processed at workstations and those which are still waiting to be processed. Since WIP ties up capital and belongs to non-reinvesting cost and the capacities of buffer for storing WIP is typically small, it needs to be optimized and controlled [40]. Generally, WIP is regulated through adjusting the order release rate to achieve a stable production process and improve the shop floor control performance. Many methods have been developed, e.g., constant WIP control [50], decentralized WIP-oriented manufacturing control [51] and workload control [23, 24, 52, 53]. Yet, the probability of high load periods was not decreased by using order release control in the form of input control [53], cf., Figure 2.2. On the other hand, capacity control from the view of output control has not received much attention. As a crucial actuating variable, the actual output rate (cf., Figure 2.2) determined by any capacity control influences both WIP and backlog control variables and therefore significantly impacts all of the logistic objectives [4]. Typically, capacity planning is concerned with resources and factors which affect the decision of manufacturer to produce products including equipments, manpower, time, etc. The goal of capacity control is to reach the best optimal policy on performance indicators and resources utilization to meet customer demand while considering

the constraints in kinds of operational objectives. In [12], the authors demonstrated that during a high load stage, even a slightly decrease in WIP via capacity adjustment would alleviate the strain and improve the production and scheduling effectiveness.

Apart from WIP, there exists other performance indicators. Throughput time refers to the time period from order released to the shop floor until its completion. In made to order production, throughput time is the lower limit for the delivery time to customers. Improving the schedule reliability and achieving a short delivery time is important for both of manufacturers and consumers. Utilization represents that usage of workforce (e.g., manpower, machines). Typically, if the WIP is higher, then the utilization is comparatively better due to sufficiently many arrival orders to be processed without waiting time. Yet, if customer demand is greater than the available capacity of a bottleneck, then this significantly impacts on production efficiency and stability [54]. Moreover, the machine utilization before a bottleneck workstation decreases, which leads to lost sales and profits. At the same time, it leads to a long lead time and low reliability of due dates for the overall system. Regarding the area of optimal WIP, it has been studied with the help of logistics characteristics curves for one work system [55] and further extended to production networks [56]. Note that the optimal area of WIP is typically achieved by adjustment of the order release rate to obtain a comparatively short throughput time along with a comparatively high machine utilization [56]. In this thesis, we only focus on the operational layer. As a consequence, we cannot influence the order release rates to any of the workstations. Instead, given the expected WIP level, our task is to utilize capacity adjustments via RMTs to balance capacity and load in case of bottlenecks.

## 2.4. APPROACHES OF CAPACITY CONTROL

In general, approaches of capacity control that focus on working times are divided into backlog control, plan oriented capacity control, due date oriented capacity control, output rate maximizing capacity control and inventory based capacity control [4, 11]. Yet, the cost of labor oriented approaches is relatively high and it is not a sustainable solution in a long-term consideration. Alternatively, capacity control via machinery can be accomplished by control-theoretical approaches and reconfigurable manufacturing systems [11].

### 2.4.1. CAPACITY CONTROL VIA CONTROL-THEORETICAL APPROACHES

Control-theoretical approach as a very strong powerful tool for analysis and control of manufacturing process was introduced in Section 2.2.2. In this thesis, our aim is not to go through all performance indicators available for job shop control. Instead, we only consider how to effectively regulate WIP. More specifically, we restrict ourselves to controlling WIP of each workstation to the desired values via capacity control. For this arena, we conduct



a review of the related literature and present the representative works.

Utilizing a funnel model and logistics operating curve, [57] built up a continuous flow model for the description of dynamic job shop systems. Moreover, they designed a WIP controller in conjunction with a backlog controller to control WIP and eliminate the backlog through adjusting order release rate and capacity. The results indicated that the convergence of backlog and mean WIP level with control was faster than in the uncontrolled case in the presence of an unplanned urgent order. In [58], using transfer function and proportional capacity control, the authors developed a dynamic closed-loop model of a multi-workstation system without information sharing and further extended it to a coupled closed-loop capacity adjustment with information sharing to control WIP [59]. In [10], continuous time modeling and simulation methods concerning local capacity control were evaluated via a discrete state space model setting and further compared with DES. The results indicated that there was a subtle difference in terms of mean and variation of WIP and lead time, which showed that the control theoretic approach provides an additional research possibility on manufacturing process control [57, 58, 60]. However, due to lack of representation of individual orders and machines, such fidelity of regulating WIP by using a control-theoretic model would decrease at extreme operating conditions, e.g., at low WIP level, or require large capacity adjustments [61]. Nevertheless, using a trial-and-error method and a theoretical analysis can improve the potential fidelity of such dynamic models and therefore can be applied in the design of large production control network. [62] introduced a class of production network in the form of a discrete-time singular. They employed  $H_\infty$  control for capacity changes with time-delay. The WIP level in each workstation was maintained and the system's asymptotic stability was guaranteed by an appropriate Lyapunov function. In [63], an optimal adaptive WIP regulation scheme was proposed in accordance with the coordination of multiple capacity adjustments modes under different adjustment periods and delays. [64] applied the concept of bond graphs to the manufacturing system and introduced a model of multi-product job shop system. The system was designed to maintain a desired WIP level by means of a proportional controller. [65] further utilized bond graph and presented a dynamic multi-product model, which was driven by MPC for the frequency analysis. In [66], the authors systematically summarized the traditional methods in production line control, e.g. base stock, Kanban, constant WIP, dynamic WIP and learning agents. A comparative analysis of the such methods was carried out. Thereafter, they also presented a collaborative control theory approach to maintain throughput while reducing the required level of WIP. The study pointed out that the latter was superior to the previous relying on optimization of the system performance with the updated system states.

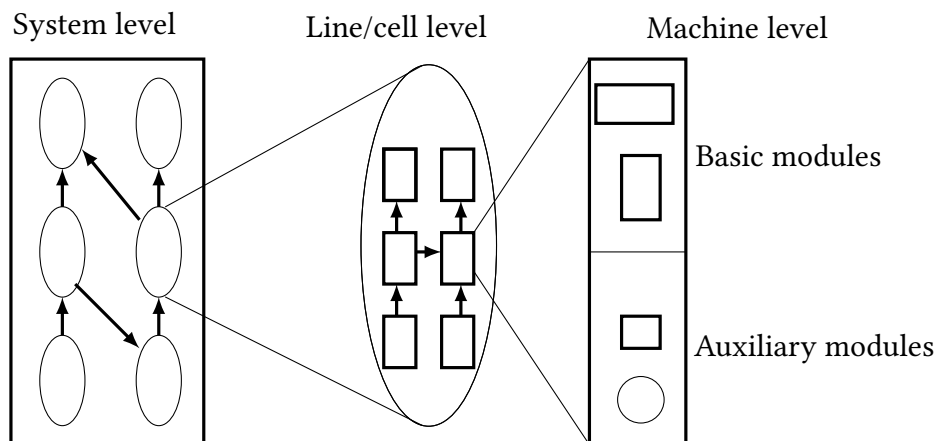
The above reported works gained fruitful results and significantly improved shop floor control performance. However, the potential of reconfigurable manufacturing systems along with reconfigurable machine tools for capacity adjustment was ignored.



### 2.4.2. RECONFIGURABLE MANUFACTURING SYSTEMS

Reconfigurable machine systems (RMSs) was first proposed by Yoram Koren at University of Michigan. These are flexible by construction and designed to rapidly change production capacity and functionality with a part family in response to market requirements [67, 68]. Such systems show significant impact on sustainable manufacturing to improve the responsiveness to market changes while remaining cost-effective [69]. The characteristic of RMS include modularity, scalability, integrability, convertibility, diagnosability and customization [1]. Scalability and convertibility refer to the ability of flexibly changing the capacity and functionality, which are achieved through modularity, integrability, and diagnosability. Typically, the scalability of RMS can be achieved more easily to achieve than convertibility as the latter will result in production interruption [70]. Last characteristics, customization means that the system and machine flexibility are not broadly like FMS but limited and customized to a specific product family. In [71], a fluid dynamic model analogy for RMS was developed by a control theoretic approach. The stability was ensured by the bounded reconfiguration rate and the response delay to demand variations. Considering risk and uncertainty in the RMS before investment, the feasibility of the construction of an RMS on a tactical level was evaluated by the fuzzy analytical hierarchical process in [72]. Inspired by the delayed product differentiation, a concept of delayed RMS was proposed in [70]. The authors constructed a new basic RMS structure to maintain part of production activities via semi-finished products to reduce the production loss.

In order to enhance the responsiveness in the dynamic production environments and to react to external demand, companies push forward from flexibility to reconfigurability [73], e.g., reconfiguring their products and processes in the RMS. Reconfigurability is a very important property, which is defined as the ability to permanently change its components in a flexible and cost-effective way to accomplish predefined objectives [7, 74, 75]. The achievement of reconfigurability in RMS includes three hierarchical levels shown in Figure 2.3. On



**Figure 2.3.:** Structure of a reconfigurable machine system

the system level, reconfigurability is achieved by adjusting the number of manufacturing lines. Through adding and removing or changing the positions of these machines, the reconfigurability at line level is achieved. Regarding the machine level, the core component and key enabler of RMS is the so-called RMT which is customized to perform a variety of operations by flexibly adding or removing its auxiliary modules associated with the usage of basic modules when and where needed, owning a plug and play characteristics. The RMS may consist of dedicated machines, CNC machines, and RMTs. Investment and configuration of these machines are determined by managers on a strategic and planning level. For instance, the optimal portfolio of reconfigurable and dedicated capacity under uncertainty was investigated on a planning level in [76]. It indicated that the percentage of Dedicated Machine Tools (DMTs) and RMTs should be driven by the relative cost to maximize the portfolio as well as by the prediction on how often new products would be introduced and by the expected level of demand.

The development of RMS along with RMTs plays an important role in the context of Industry 4.0 and increasingly attracts a lot of researchers to contribute on it. Faced with changeable customer demand, the core is the configuration design of manufacturing system. In [77], a configuration management and a sequential approach for production planning in RMS was proposed to optimize allocation of resources. In [78], an optimal configuration planning method was introduced to enhance flexibility in a dynamic manufacturing system. The capacity scalability and functionality of RMS changes could be realized by using mixed integer linear programming. In [79], the authors proposed a reconfiguration point decision method for the best reconfiguration time in RMS. In [80], the authors proposed an IoT-based RMS to solve the reconfiguration planning problem. The four main components – people, process, data and things in the framework of IoT corresponds to operator, reconfiguration, RMT and difficulty of reconfiguration, respectively. The authors built a mathematical model by Plant Simulation to determine the most effective configurations while minimizing the reconfiguration cost and time. Recent developments and future directions of RMS were comprehensively introduced in [81], which is strongly linked to Industry 4.0.

The study of RMS gained a lot of fruitful results but mainly focused on the system level. The approaches of reconfigurable manufacturing systems regarding capacity adjustment still remains on a rather abstract modelling level [11, 82]. Considering the limited space, investment and high reliability of delivery time, reconfiguring an existing RMT is of preference compared to replacing the entire RMT [5]. Therefore, in this thesis, we focus on the machine level and exploit the potential of RMT concerning capacity adjustment and combine it with a control theoretic model of job shop systems to achieve logistics objectives.

## 2.5. RECONFIGURABLE MACHINE TOOLS

In the view of machine equipments, traditional machine tools such as DMTs and Flexible Machine Tools (FMTs) cannot meet the requirements to compensate for frequent demand fluctuations. DMTs are designed to produce a single product at a high rate of production, which is achieved by utilizing all tools simultaneously. The only way to enhance the capacity is to purchase additional equipment. While this covers for peaks in demand, these tools may stay idle or remain underutilized at normal demand levels. FMTs can produce a variety of products according to respective demands, but are very expensive and typically their full functionality is not exploited which results in unnecessary costs due to software development and installation. As they are not designed for structural changes, they are not suitable to react to persistent market fluctuations. Hence, considering a manufacturer's point of view, a sustainable machine tool is required which allows for increasing flexibility, improving the productivity, reducing energy consumption, shortening delivery times and enhancing responsiveness in the presence of demand fluctuations [83].

Aiming to fulfill these wishes, RMTs were developed as a new class of machine tools, which combine the high throughput of DMTs and flexibility of FMTs [84], cf. Table 2.1. These machines are designed modularly for customized operation requirements and may be cost-effectively, quickly and permanently reconfigured in an open-architecture, which renders them ideally suited for a machinery based capacity adjustment in the presence of short or mid-medium fluctuation in demand [56, 84], cf., Figure. 1.1. In contrast to rebuilding, reconfiguration as the key factor to handle the exceptions and performance deteriorations in manufacturing systems [85] does not focus on the preparation of the machine to complete an order, but to adapt the structure, technology and functionality including the functional capacity of the manufacturing site.

**Table 2.1.:** RMT combines features of dedicated and CNC machines [1]

	Dedicated	RMT/RMS	CNC/FMS
System structure	Fixed	Adjustable	Adjustable
Machine structure	Fixed	Adjustable	Fixed
System focus	Part	Part family	Machine
Productivity	High	Medium	Low
Scalability	No	Yes	Yes
Flexibility	No	Customized	General
Cost	Low	Medium	High
Simultaneous operating tools	Yes	Yes	No

The illustration of a workflow of a manufacturing process with RMTs is given in Figure 2.4. In the first place, the customers launch order requests to industry and place the orders. Second, technicians convert these orders into design requirements and send them to designers who are responsible for establishing the manufacturing system to satisfy the requirement. Third, operators adjust the manufacturing system and handle those orders on the shop floor. Last, when the parts are completed, they will be delivered to the customers. In this thesis, we consider the operational layer which is the case of the third step. Recent development of RMTs was investigated in [86], indicating that the contributions mainly include the three fields: architecture design, configuration design with optimization as well as system integration and control. Within this thesis, we cannot consider all references available concerning study of RMTs. Instead, we choose some key contributions which were published in the high level journals. Also, we pay attention to the development of RMTs, especially for the recent years.

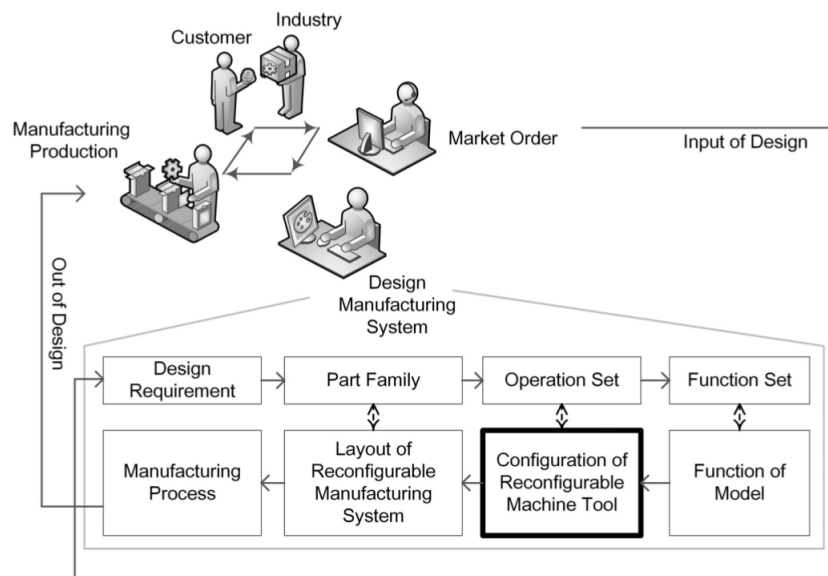


Figure 2.4.: Workflow of a manufacturing process with RMTs [5]

### 2.5.1. ARCHITECTURE DESIGN

In order to fulfill a variety of operations to meet productivity and functionality, the designed RMTs have to take the mechanical requirements including kinematic viability and structure stiffness into account. The modular design of RMTs on hardware level was conducted in [87, 88, 89, 90, 91, 92]. The control and mechanical requirements of RMT was presented in [84, 93, 94, 95]. Several types of RMTs have been developed for the design of manufacturing processes, such as module 3-axis RMT, arch type RMT, which allow the implementation of multiple operations [96, 97].

### 2.5.2. CONFIGURATION DESIGN WITH OPTIMISATION

In [98], the authors optimized the production process plan associated with the configurations of RMTs via a genetic algorithm to improve production capacity and reduce production cost. In [99], the authors investigated the optimal design of dedicated and reconfigurable manufacturing systems for adapting to a variable market demand with the minimization of investment costs. Such kind of combinatorial optimization problem is NP-hard, which can be solved by meta-heuristics strategies to achieve suboptimal solutions, e.g., genetic algorithm, particle swarm optimization, ant colony optimization, etc. In [97], the authors developed a hybrid generic design AND-OR tree and AND-OR graph for the optimal design of a reconfigurable process and of configurations of RMTs. The solutions were later identified by a multi-objective hybrid optimization. To balance the trade-off between economy and responsiveness, [100] applied the non-dominated sorting genetic algorithm to solve a multi-objective optimization problem for the optimal selection of machine configuration. The performance index indicated cost, operational capability and machine reconfigurability. Then these obtained solutions were further analyzed and ranked by the “technique for order preference by similarity to ideal simulation” approach.

In [69], relying on production capability of RMTs, the authors studied a single product line to satisfy demand changes. The production capacity was increased or decreased through adding or removing auxiliary modules for performing different operations, and also purchasing new RMTs to increase capacity while minimizing reconfiguration cost and capital investment cost. To exploit the best configuration, a mixed integer linear programming problem was formulated. Two cases concerning cost management were presented to demonstrate the efficiency of the proposed method. In [5], based on graph theory method, a tree-based decision approach for the determination of a configuration design of RMTs was proposed. The satisfactory configuration was conducted to meet operational requirements while minimizing cost and maximizing reconfigurability. The most satisfactory configuration from the feasible design space could be derived based on the performance indicators. In [101], the authors built a multi-objective optimization model incorporating configurability, cost and process accuracy for RMTs design. A modified fuzzy-Chebyshev programming approach was proposed which allows to dynamically adjust objective weights in the search space. The derived solution was obtained by particle swarm optimization.

### 2.5.3. SYSTEM INTEGRATION OF CONTROL

In [102], the authors formulated reconfigurability of RMTs in a discrete event system regarding its corresponding configurations. A polynomial-time algorithm was proposed for the construction of a reconfiguration supervisor. The configuration states were monitored by means of supervisor control and allowed controller modifications during time run. In [103], an intelligent Fuzzy PID controller was designed to provide a viable solution in an

open-architecture control system for RMTs. The controller was able to guarantee stability, to achieve set point tracking and also to provide a certain robustness over a range of operating conditions and parameters changes. In [104], using quantitative feedback theory with artificial neural networks, a robust controller was designed for running reconfigurable micro-machine tools under different operation points with parametric uncertainties. In [15], a hybrid system which consists of DMTs and RMTs was analyzed in terms of productivity and make span by means of discrete event simulation. Taking into account frequent reconfiguration cost, the responsiveness of RMT was measured and evaluated by operational capability and machine reconfigurability metrics [105]. The reconfiguration time was much less than the time to repair a broken machine, usually less than two hours [11], which showed that reconfiguration is an effective strategy to handle exceptions or short-term fluctuations in terms of production system performance [85]. This reconfigurability and flexibility can be exploited best within manufacturing systems with high product diversity at small lot sizes, which is the case, e.g., for job shop systems [11].

## **2.6. CAPACITY CONTROL WITH RMTs IN JOB SHOP SYSTEMS**

The potential of RMTs for capacity adjustment was first exploited in job shop systems in conjunction with workforce flexibility in [11], which is referred as “throughput time harmonising capacity control (THCC)” to react to demand fluctuations in the presence of bottlenecks. Further, the proposed method was evaluated in real time and compared to the real-world data from a manufacturing company and the due-date oriented capacity control method in terms of mean throughput time, mean inventory range, mean utilization, adherence to due date and makespan. The results indicated that using THCC method outperformed the other two for all performance indicators. As discussed before, capacity control on machinery can be achieved by either using control theoretic approaches or reconfigurable machine tools. Taking the respective advantages of both approaches, i.e., capacity control via RMTs conducted in a control theoretic model, on the one hand, production flexibility and responsiveness can be accomplished by RMTs under unpredictable events. On the other hand, the inherent characteristics of reconfigurability with changeable capacity in RMSs/RMTs allows to apply control theory method to identify a good configuration. Based on the captured production data in a closed-loop feedback, RMTs can reconfigure to fluctuations in demand [86]. In [106], capacity adjustment with RMTs was first implemented by a PID control method, which aimed at maintaining WIP on a planned level in a four-workstation job shop system. The achievement of a desirable dynamic transition behavior was attained. Yet, this method cannot handle constraints effectively and additional measures to decouple the MIMO process are required. Further, considering the uncertainties



and disturbances in the dynamic manufacturing process, the authors modeled a job shop system with RMTs and decomposed it into two operators for the design of robust stabilizing controllers [60]. Afterwards, they designed a tracking controller with respect to WIP. The tracking performance could be ensured in the presence of bounded uncertainty by robust right coprime factorization. The advantage of this method is to naturally handle coupling effect between the workstation. Once the robust controllers was analytically derived, it was ready to be used in the nonlinear dynamic process. Yet, such operator controllers are very difficult to be derived and also cannot effectively deal with constraints.

In order to reduce the negative effects and achieve a high quality of shop floor control performance, in this thesis, we will use a predictive technique with a time planning horizon (i.e., MPC) to control the manufacturing process with RMTs for capacity adjustments. Since MPC is capable of incorporating the system dynamics including reconfiguration delays, the key performance indicators and the constraints of the process into a feedback law, it is ready to be used and expected to improve the operational reliability.

## **2.7. SUMMARY**

In order to cope with fluctuations in case of bottlenecks, capacity adjustment is an effective proactive measure and generally achieved by flexible working hours. Yet, this is not a sustainable solution especially in the western countries with a requirement of high-paid wages. As a new type of promising machine tools, RMTs can change their capacity and functionality when and where needed to meet production requirements in a cost-effective way, which is especially beneficial for job shop systems [11]. In a turbulent dynamic manufacturing environment, the potential of RMTs as well as new of possibilities regarding planning and control of the capacity shall be urgently exploited and employed. To this end, the development of RMTs, of modelling strategies of manufacturing process and of capacity control approaches were introduced. The control-theoretical approach can be used to understand and analyze the fundamental behaviors in both steady-state and transient phases. Combining capacity flexibility of RMTs with control theoretic approaches, both capacity and load can be effectively balanced with a quick response to demand fluctuations.

# 3

## DEFINITIONS AND PRELIMINARIES

In Chapter 2, we showed that RMTs may be applied as an additional degree of freedom for capacity adjustment in job shop systems to maintain WIP in the presence of demand fluctuations. Yet, RMTs are only an enabling technology on the operational level. To render capacity adjustment effective and ensure sustainable quality and preferable performance in the future industrial manufacturing, it is necessary to complement these tools with respective modeling and control strategies to improve efficiency and reduce risks with respect to changing configurations. Thus, the goal of this chapter is to present the basic concepts with respect to optimization (Section 3.2), control and stability (Section 3.3), and in particular the stability of MPC (Section 3.5), which is basically cosponsoring to the main research route, cf., Figure 1.3. As a mathematical foundation and tool, this will be used to address the integer assignment of RMTs in Chapter 5 based on the introduced optimization techniques from Section 3.2 and analyze the stability of closed-loop system controlled by MPC with RMTs in Chapter 6 based on the knowledge introduced in Section 3.5.

**Notation:** Through this thesis, we denote the natural numbers including zero by  $\mathbb{N}_0$  and the nonnegative reals by  $\mathbb{R}_{\geq 0}$ . For a vector  $x \in \mathbb{R}^n$ ,  $n \in \mathbb{N}$ ,  $\|x\| = \sqrt{\sum_{j=1}^n x_j^2}$  is represented as the 2-norm. For sets  $\Omega_A$  and  $\Omega_B$ , we denote the set subtraction  $\Omega_A \setminus \Omega_B := \{x \in \mathbb{R}^n \mid x \in \Omega_A, x \notin \Omega_B\}$ . For a square matrix  $P \in \mathbb{R}^{n \times n}$ , we denote  $\lambda_{max}(P)$  and  $\lambda_{min}(P)$  as the maximum and minimum eigenvalue of the matrix, respectively. We denote  $P^\top$  as the transpose matrix of  $P$  and denote its derivative as  $\frac{\partial x^\top P x}{\partial x} = (P + P^\top)x$ . For a variable  $x \in \mathbb{R}$ , the floor operator  $\lfloor \cdot \rfloor$  is defined as  $\lfloor x \rfloor := \max\{a \in \mathbb{Z} : a \leq x\}$ . For vectors  $x \in \mathbb{R}^n$ ,  $y \in \mathbb{R}^n$ ,  $b \in \mathbb{R}^n$ , we have  $\frac{\partial \|x-b\|_2}{\partial x} = \frac{x-b}{\|x-b\|_2}$ ,  $\frac{\partial \|x\|_2}{\partial x} = 2x$ ,  $\frac{\partial y^\top x}{\partial x} = y$ . For  $A \in \mathbb{R}^{m \times n}$ ,  $B \in \mathbb{R}^{m \times n}$ ,  $a \in \mathbb{R}$ ,  $x \in \mathbb{R}^n$ , we have positivity of matrix norm  $\|A\| \geq 0$ , and  $\|A\| = 0$  iff  $A = 0$ , homogeneity:  $\|aA\| = |a|\|A\|$ , triangle inequality:  $\|A + B\| \leq \|A\| + \|B\|$  and  $\|A - B\| \geq \|A\| - \|B\|$ , subordinance:  $\|Ax\| \leq \|A\|\|x\|$ . If  $m = n$ , we also have submultiplicativity:  $\|AB\| \leq \|A\|\|B\|$ .



### 3.1. BASIC CONCEPTS OF DISCRETE TIME CONTROL SYSTEMS

In this thesis, the system considered is of discrete time form and will be modelled in detail in Chapter 4.

**Definition 3.1** (Discrete time control system)

Consider a general discrete time control system

$$x(n+1) = f(x(n), u(n)) \quad (3.1)$$

with  $n \in \mathbb{N}$  and  $f : \mathbb{R}^p \times \mathbb{R}^m \rightarrow \mathbb{R}^p$ .  $x \in \mathbb{X} \subset X \in \mathbb{R}^p$  and  $u \in \mathbb{U} \subset U \in \mathbb{R}^m$  are the system state and the control action within corresponding constraints, respectively. Here,  $\mathbb{X}$  is a closed and convex set and  $\mathbb{U}$  is a compact set, not necessarily convex.

**Definition 3.2** (Equilibrium)

A pair  $(x^*, u^*)$  that satisfies  $f(x^*, u^*) = x^*$  with  $x^* \in \mathbb{X}$  and  $u^* \in \mathbb{U}$  is called equilibrium point for the system (3.1).

**Definition 3.3** (Discrete time dynamical system)

Consider a function  $f : \mathbb{X} \rightarrow \mathbb{X}$ ,  $n \in \mathbb{N}$ , the system

$$x(n+1) = f(x(n)) \quad (3.2)$$

is called discrete dynamical system.

If the control input  $u$  in (3.1) is a function of state  $x$ , then (3.1) can be transformed into the form of (3.2) which is independent of  $u$ . Since controllers like model predictive control are state feedback control laws, we will analyze the stability of such resulting dynamical systems in Section 3.3.

**Definition 3.4** (Positive control invariant sets)

A set  $\Omega \in \mathbb{R}^p$  is called positive control invariant set for the described system (3.1) if there exists a control input  $u \in \mathbb{U}$  such that  $f(x_0, u) \in \Omega$  holds for any initial current value  $x_0 \in \Omega$ . A set  $\Omega \in \mathbb{R}^p$  is called positive invariant set for dynamical system (3.2), if  $f(x_0) \in \Omega$  holds for any  $x_0 \in \Omega$ .

### 3.2. NONLINEAR PROGRAMMING

Discrete time optimal control problems can be transformed to standard nonlinear programming by discretization [107]. Thus, before reaching the control layer, we first go for the optimization and planning layer and introduce some basic concepts concerning nonlinear

programming and further mixed integer nonlinear programming which refers to the integer assignment of RMTs.

**Definition 3.5** (Nonlinear programming)

The problem

$$\min f(x) \quad (3.3)$$

$$\text{s.t. } c_j(x) \leq 0, j = 1, 2, \dots, m, \quad (3.4)$$

$$h_k(x) = 0, k = 1, 2, \dots, l. \quad (3.5)$$

is called nonlinear programming, if at least one of the functions  $f : \mathbb{R}^p \rightarrow \mathbb{R}$ ,  $c_j : \mathbb{R}^p \rightarrow \mathbb{R}$ ,  $h_k : \mathbb{R}^p \rightarrow \mathbb{R}$  is nonlinear. The goal is to determine variable  $x$  to minimize the given objection function  $f(x)$  in the presence of inequality constraints (3.4) and equality constraints (3.5).

**Definition 3.6** (Feasible points)

The points  $x \in \mathbb{R}^p$  are called feasible if it satisfies the constraints (3.4) and (3.5).

**Definition 3.7** (Global optimal solution)

Suppose  $\mathbb{X}$  is the set of all feasible points. If there exists  $x^* \in \mathbb{X}$  such that  $f(x) \geq f(x^*)$  for all  $x \in \mathbb{X}$ , then  $x^*$  is called global optimal solution.

**Definition 3.8** (Local optimal solution)

Suppose  $\mathbb{X}$  is the set of all feasible points, if for a set  $S \in \mathbb{X}$ , there exists  $x^* \in S$  such that  $f(x) \geq f(x^*)$  for all  $x \in S$ , then  $x^*$  is called local optimal solution.

**Definition 3.9** (Positive-(semi) definite matrix)

A matrix  $M \in \mathbb{R}^{p \times p}$  is called positive-definite if  $x^\top M x > 0$  holds for all  $x \in \mathbb{R}^p \setminus \{0\}$ . It is positive semi-definite if  $x^\top M x \geq 0$  holds for all  $x \in \mathbb{R}^p$ .

**Definition 3.10** (Lagrangian function and Lagrangian multipliers)

A Lagrangian function of the nonlinear programming problem (3.3), (3.4), (3.5) is given by  $L(x^*, \mu, \lambda) := f(x^*) + \mu^\top c(x^*) + \lambda^\top h(x^*)$  are called Lagrangian multipliers, where  $\mu \in \mathbb{R}^m$ ,  $\lambda \in \mathbb{R}^l$ .

Now, we briefly introduce the Karush-Kuhn-Tucker (KKT) conditions, which play a significant role in the nonlinear programming.

**Definition 3.11** (KKT conditions—first order optimality conditions)

Given a nonlinear programming problem [Definition 3.5], assume  $f(x)$ ,  $c(x)$  and  $h(x)$  are all twice continuously differentiable functions. If  $f(x)$  attains a local minimum at  $x^*$  subject to the constraints (3.4) and (3.5), then there exist Lagrange multipliers  $\mu, \lambda$  such that the

following conditions hold:

$$\frac{\partial f(x^*)}{\partial x_i} + \sum_{j=1}^m \mu_j \frac{\partial c_j(x^*)}{\partial x_i} + \sum_{k=1}^l \lambda_k \frac{\partial h_k(x^*)}{\partial x_i} = 0, \quad \text{Stationarity} \quad (3.6)$$

$$\mu_j c_j(x^*) = 0, \quad \text{Complementary slackness} \quad (3.7)$$

$$\mu_j \geq 0, \quad \text{Dual feasibility} \quad (3.8)$$

$$c_j(x^*) \leq 0, \quad \text{Primal feasibility} \quad (3.9)$$

$$h_k(x^*) = 0. \quad \text{Primal feasibility} \quad (3.10)$$

for  $i = 1, 2, \dots, p$ ,  $j = 1, 2, \dots, m$  and  $k = 1, 2, \dots, l$ . Note that the KKT conditions are only necessary but not sufficient. That is, if a local minimum exists, it must satisfy the KKT conditions. Yet, those  $x$  that satisfy the conditions are not necessarily a local minimum. To be sufficient for optimality, we need to additionally check the Second Order Sufficient Conditions (SOSC). Yet, if the solved optimization problem is convex, then the KKT conditions are necessary and also sufficient. Note that the aim of this thesis is not to deeply go into optimization algorithms but use these techniques to solve the allocation of RMTs on the shop floor, which typically refers to solving a convex integer optimization problem. Regarding the details concerning SOSC and also the constraints qualifications that find a local minimum to satisfy KKT conditions, we refer to [108, 109].

**Definition 3.12** (Quadratic programming)

Quadratic programming is a special case of nonlinear programming, in which the objective function  $f(x)$  is quadratic and the constraints are linear, i.e.,

$$\begin{aligned} \min \quad & f(x) = \frac{1}{2}x^\top Qx + q^\top x + d \\ \text{s.t.} \quad & Ax \leq b. \end{aligned} \quad (3.11)$$

where  $x \in \mathbb{R}^p$ ,  $q \in \mathbb{R}^p$ ,  $Q \in \mathbb{R}^{p \times p}$ ,  $A \in \mathbb{R}^{m \times p}$  and  $b \in \mathbb{R}^m$ ,  $d \in \mathbb{R}$ .

**Definition 3.13** (Least square problem)

Given a matrix  $A \in \mathbb{R}^{m \times p}$ ,  $m \geq p$ ,  $x \in \mathbb{R}^p$ ,  $b \in \mathbb{R}^m$ ,  $a_j^\top$  are the rows of  $A$ , a least square problem is an optimization problem without constraints and the associated objection function is a sum of squares in the form of

$$\begin{aligned} \min \quad & f(x) = \sum_{j=1}^m (a_j^\top x - b_j)^2 = \|Ax - b\|_2^2 \\ & = x^\top A^\top Ax - 2b^\top Ax + b^\top b. \end{aligned} \quad (3.12)$$

For the linear least squares problem (3.12), the analytic solution can be derived with  $x = (A^\top A)^{-1} A^\top b$  [108]. Regarding the nonlinear standard least squares, the numerical solution

can be obtained with Levenberg-Marquardt or Gauss-Newton algorithm [109]. Note that the constrained least squares problem is a special case of quadratic programming problem and can be used for tracking purposes in metric spaces for dynamical system (3.3)

$$f(x) = \|x - x^*\|_2^2 = \sum_{j=1}^p (x_j - x_j^*)^2, \quad (3.13)$$

where  $x - x^*$  represents the distance between current state to the equilibrium.

**Definition 3.14** (Convex optimization problem)

Given the objective function (3.3) and the associated constraints (3.4), for  $x, y \in \mathbb{R}^p$  and  $\alpha, \beta \in \mathbb{R}$  with  $\alpha + \beta = 1$ ,  $\alpha \geq 0$ ,  $\beta \geq 0$ , if the following inequalities

$$f(\alpha x + \beta y) \leq \alpha f(x) + \beta f(y) \quad (3.14)$$

$$c_j(\alpha x + \beta y) \leq \alpha c_j(x) + \beta c_j(y) \quad (3.15)$$

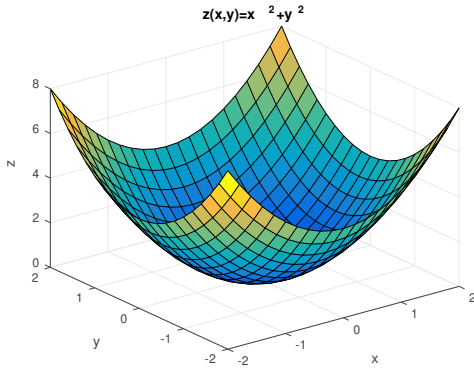
hold and (3.5) are affine, then it is called a convex optimization problem [108].

### 3.2.1. MIXED-INTEGER NONLINEAR PROGRAMMING

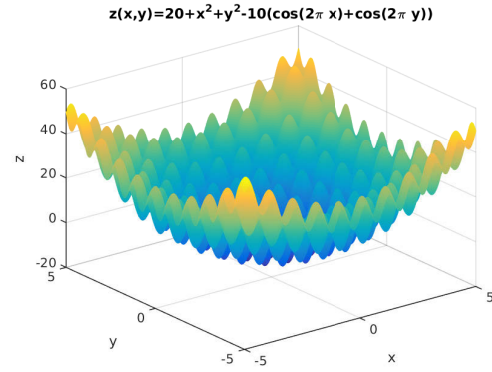
For solving the general nonlinear convex optimization problems, a lot of algorithms have been developed, e.g., Sequential Quadratic Programming (SQP), or Interior-Point (IP) Method [109]. Yet, in our case, the assignment of RMTs in job shop refers to an integer programming problem. Thus, based on the introduced nonlinear programming, we further briefly introduce mixed-integer nonlinear programming, which requires both of continuous and discrete variables to solve the optimization problem with the nonlinear object function and/or constraints [110]. In reality, a variety of practical applications can be transformed to mixed-integer (non) linear programming problem, such as job planning and scheduling problems [111, 112], control of hybrid system [113], unmanned aerial vehicles [114], or piecewise affine systems [115, 116]. Since this kind of problem typically cannot be solved in polynomial time, it represents an NP-hard combinatorial problem. Its mathematical expression is given by:

$$\begin{aligned} \min \quad & f(x) & (3.16) \\ \text{s.t.} \quad & c_j(x) \leq 0, \quad j = 1, 2, \dots, m \\ & h_k(x) = 0, \quad k = 1, 2, \dots, l \\ & x \in X, x_i \in \mathbb{Z} \quad \forall i \in I. \end{aligned}$$

where  $x \in \mathbb{R}^p$ ,  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $c : \mathbb{R}^p \rightarrow \mathbb{R}^m$  are twice continuously differentiable functions and  $I \subseteq \{1, \dots, p\}$  is a set of all indexes. If  $f(x)$ ,  $c(x)$ ,  $h(x)$  are convex, cf., Figure 3.1, then the optimization problem (3.16) is called convex mixed-integer nonlinear program-



**Figure 3.1.:** An illustrated example of convex optimization



**Figure 3.2.:** An illustrated example of non-convex optimization

ming [117, 118]. Methods to solve this type of optimization problem include branch and bound [119], generalized Benders Decomposition [120] and Outer Approximation [121]. Concerning the non-convex mixed-integer nonlinear programming problems where either the objective function and/or the constraints are non-convex, cf., Figure 3.2, the feasible region is in general non-convex which results in many possible minima [108, 122]. This reveals a global optimization problem and is very hard to be solved and verified. Due to less complexity and computational cost to derive near-optimal solutions, meta-heuristic methods [123] have been widely applied in practice e.g., genetic algorithm, particle swarm optimization, ant colony algorithm, simulated annealing, tabu search, etc.

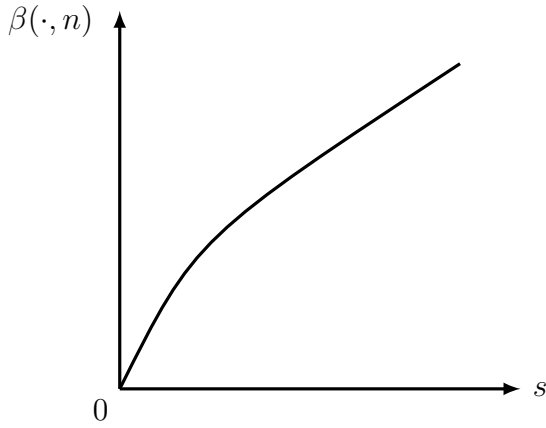
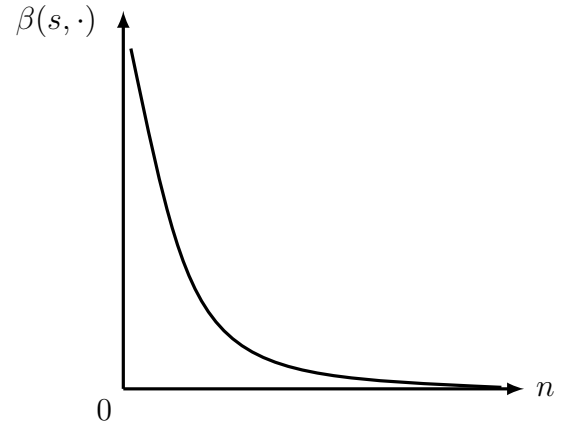
### 3.3. STABILITY OF DISCRETE TIME CONTROL SYSTEMS

As we discussed before, due to the capability of flexibly changing capacity and functionality, RMTs can improve the shop floor performance with possibly frequent reconfigurations. Here, we present some basic concepts concerning stability, particular for the stability of MPC which is the type of controller that will be used throughout this thesis.

Lyapunov stability plays an important role in the stability analysis of control systems, especially for nonlinear systems. It allows us to analyze the stability property by using the concept of energy dissipation without solving ordinary differential equations. Therefore, it is widely applied and accounted to be predominant in the development of control theory [124]. Here, the stability term we considered is related to initial conditions. Concerning the bounded input bounded output (BIBO) stability, we refer to [124]. To analyze stability of (non)linear systems, comparison functions are an effective and strong tool concerning the convergence property of a closed loop of a nonlinear dynamical systems.

**Definition 3.15** (Comparison functions)

A function  $\alpha : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$  is of class  $\mathcal{K}$  if  $\alpha$  is continuous, strictly increasing and  $\alpha(0) = 0$ . class  $\mathcal{K}_{\infty}$  is unbounded.

Figure 3.3.:  $\mathcal{K}$  functionFigure 3.4.:  $\mathcal{L}$  function

A function  $\varphi : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$  is of class  $\mathcal{L}$ , if  $\varphi$  is continuous, non-increasing and its limit is zero.

A function  $\beta : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$  is of class  $\mathcal{KL}$ , if  $\beta(\cdot, n) \in \mathcal{K}$  for all  $n \in \mathbb{R}_{\geq 0}$ , cf., Figure 3.3, and  $\beta(s, \cdot)$  is strictly decreasing in its second argument with  $\lim_{n \rightarrow \infty} \beta(s, n) = 0$  for all  $s \in \mathbb{R}_{\geq 0}$ , cf., Figure 3.4.

### 3.3.1. ASYMPTOTIC STABILITY

**Definition 3.16** (Asymptotic stability)

Suppose a system (3.1) and a control  $u(\cdot)$  to be given such there exists a forward invariant set  $Y \subset \mathbb{X}$ . If there exists a function  $\beta \in \mathcal{KL}$  such that

$$\|x(n) - x^*\| \leq \beta(\|x_0 - x^*\|, n) \quad (3.17)$$

holds for all  $x_0 \in Y$  and all  $n \in \mathbb{N}_0$ , then the control  $u(\cdot)$  asymptotically stabilizes  $x^*$ , cf. Figure 3.5.

**Definition 3.17** (Control Lyapunov Function)

Given the system (3.1) with  $x^* \in Y \subseteq \mathbb{X}$ , a function  $V : \mathbb{R}^n \rightarrow \mathbb{R}_{\geq 0}$  is called a control Lyapunov function if there exist functions  $\alpha_1, \alpha_2, \alpha_3 \in \mathcal{K}_\infty$  and a control input  $u \in \mathbb{U}$  such that

$$\alpha_1(\|x - x^*\|) \leq V(x) \leq \alpha_2(\|x - x^*\|), \forall x \in \mathbb{X} \quad (3.18)$$

$$V(f(x, u)) - V(x) \leq -\alpha_3(\|x - x^*\|), \forall x \in Y \quad (3.19)$$

hold for all  $x \in Y \setminus x^*$ .

Based on converse theorem [125], the stability defined in Definition 3.16 can be induced by a existing control Lyapunov function defined in (3.18) and (3.19). From (3.19), we can

obtain that  $V$  is strictly decreasing along with solutions away from  $x^*$ , then we can construct a  $\bar{\beta} \in \mathcal{KL}$ , such that  $V(x(n)) \leq \bar{\beta}(V(x(0)), n)$ , then we can get  $\alpha_1(\|x(n) - x^*\|) \leq V(x(n)) \leq \bar{\beta}(V(x(0)), n) \leq \bar{\beta}(\alpha_2(\|x(0) - x^*\|), n)$ . Therefore,  $\|x(n) - x^*\| \leq \alpha_1^{-1}(\bar{\beta}(\alpha_2(\|x(0) - x^*\|), n))$ , where,  $\beta(r, n) := \alpha_1^{-1}(\bar{\beta}(\alpha_2(r), n)) \in \mathcal{KL}$ .

**Remark 3.18**

*The Lyapunov function condition (3.19) is not necessary condition for (3.17) to hold [126, 127]. That is, if  $\|x(n) - x^*\| \rightarrow 0$  can be induced as  $n \rightarrow \infty$ , the Lyapunov function does not necessarily strictly decrease but may show with possible local increases [128], which is termed as the so-called flexible control Lyapunov functions [129]. In our case, this phenomenon may occur when we take reconfiguration delays into account.*

In most cases, asymptotic stability may not be achieved, such as in the application of sampling system, quantization system and also in the presence of uncertainty [125, 130, 131, 132]. In this case, what we aim to show is that the state is confined to a bounded region around the equilibrium, which is termed as practically asymptotically stable. For instance, in our case, the assignment of RMTs as controlled variables are restricted to be integer. Due to imposing integer constraints, WIP trajectories are expected to be practically asymptotically stable.

### 3.3.2. PRACTICAL ASYMPTOTIC STABILITY

**Definition 3.19** (Practical asymptotic stability)

Suppose a system (3.1) and a control  $u(\cdot)$  to be given such there exists a positive control invariant set  $Y \subseteq \mathbb{X}$ . If there exists function  $\beta \in \mathcal{KL}$  and positive constant  $\delta$  such that

$$\|x(n) - x^*\| \leq \max\{\beta(\|x_0 - x^*\|, n), \delta\} \quad (3.20)$$

holds for all  $x_0 \in Y$  and all  $n \in \mathbb{N}_0$  and some  $\delta \geq 0$ , then the system (3.1) is said to be practically asymptotically stable. If  $\delta = 0$ , then the system (3.1) is asymptotically stable. If  $Y = \mathbb{X} = \mathbb{R}^n$ , then the system (3.1) is said to be globally practically asymptotically stable, cf. Figure 3.6.

**Definition 3.20** (Practical Control Lyapunov Function)

Given the system (3.1) with  $x^* \in Y \subseteq \mathbb{X}$ , a function  $V : \mathbb{R}^n \rightarrow \mathbb{R}_{\geq 0}$  is called a practical control Lyapunov function if there exists functions  $\alpha_1, \alpha_2, \alpha_3 \in \mathcal{K}_\infty$  and a control input  $u \in \mathbb{U}$  and some constant  $\vartheta \geq 0$ , such that the following inequalities

$$\begin{aligned} \alpha_1(\|x - x^*\|) &\leq V(x) \leq \alpha_2(\|x - x^*\|), \forall x \in \mathbb{X} \\ V(f(x, u)) - V(x) &\leq -\alpha_3(\|x - x^*\|) + \vartheta, \forall x \in Y \end{aligned} \quad (3.21)$$

hold for all  $x \in Y \setminus x^*$ .

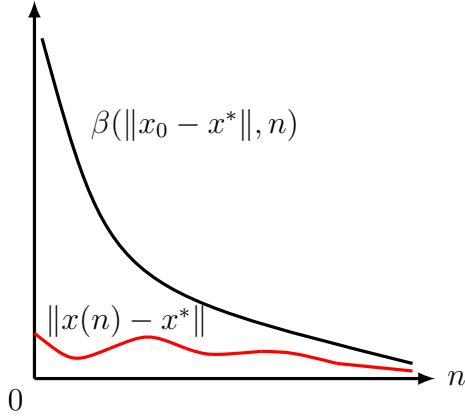


Figure 3.5.: Asymptotic stability (3.17)

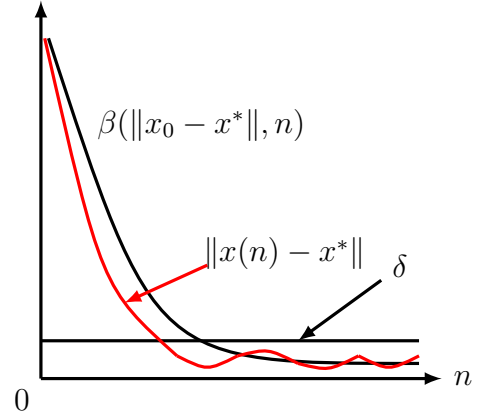


Figure 3.6.: Practical asymptotic stability (3.20)

**Theorem 3.21** (Estimated practical stability region)

Let  $V$  is a practical control Lyapunov function and there exists some positive constant  $\vartheta > 0$ . Assume that  $Y = V^{-1}[0, \varrho] := \{x \in \mathbb{X} | V(x) \leq \varrho\} \subseteq \mathbb{X}$  is a positive invariant set with  $\varrho > \tau := \alpha_2 \circ (\alpha_3^{-1}(\vartheta) + \vartheta)$ , and  $D := V^{-1}[0, \tau]$ . Then the system (3.1) is practically asymptotically stable on  $Y$ , and  $V(x)$  is a truncated control Lyapunov function on  $Y \setminus D$  with radius of ultimately boundedness  $\delta = \alpha_1^{-1} \circ \alpha_2 \circ (\alpha_3^{-1}(\vartheta) + \vartheta)$ .

The proof can be found in [133].

**Remark 3.22** (Finite time to the practical stability region)

From the practical control Lyapunov function (3.21), we have  $\alpha_3(\|x - x^*\|) \geq \vartheta$ . Based on the results presented in Theorem 3.21, since  $D \subseteq Y \subseteq \mathbb{X}$ , there exists a constant  $\epsilon > 0$ , such that  $\forall x_1 \in \mathbb{X} \setminus Y$  and  $x_2 \in Y \setminus D$  make  $\alpha_3(\|x_2 - x^*\|) \leq \alpha_3(\|x_1 - x^*\|) - \epsilon$  holds. Then we can get  $-\alpha_3(\|x_1 - x^*\|) + \epsilon \leq -\alpha_3(\|x_2 - x^*\|) \leq -\vartheta$ . Last, we can conclude  $V(x_1^+) - V(x_1) \leq -\alpha_3(\|x_1 - x^*\|) + \vartheta \leq -\epsilon, \forall x_1 \in \mathbb{X} \setminus Y$ , i.e. for the initial current value  $x_0 \in \mathbb{X} \setminus Y$ , it will be able to reach the region  $Y$  in a finite time. We call  $T_Y := \{n \in \mathbb{N} | x(n) \in Y\}$  is the minimum time to reach the region  $Y$  [134]. Similarly, we call  $T_D := \{n \in \mathbb{N} | \|x(n) - x^*\| \leq \delta\}$  is the minimum time to reach the  $\mathcal{B}_\delta$ . From (3.21), we know the system trajectories are steered to  $\mathcal{B}_\delta$  as behavior like strictly asymptotic stability on  $Y \setminus D$ , i.e.  $V(x(n+1)) \leq V(x(n))$  holds for  $n \in [0, T_D - 1]$  until reach the ball  $\mathcal{B}_\delta$  and then will stay inside the ball, i.e.  $V(x) \leq \tau$ . Thus,  $V(x(n+1)) \leq \max\{V(x(n)), \tau\}, \forall n \in \mathbb{N}_0$ .

### 3.4. TYPES OF CONTROLLER

After introducing the concepts of stability, in this section we review the development of controllers which will be applied to the manufacturing process with RMTs to accomplish capacity adjustment. A number of different controllers have been developed, such as PID, linear quadratic regulator (LQR), MPC, sliding model control, Lyapunov-based control, operator-based control, optimal control, adaptive control, etc. [48, 60, 135, 136, 137]. In



this thesis, we mainly concentrate on the former three and particularly emphasize the importance of MPC which will be applied throughout this thesis, cf., Section 2.6.

### 3.4.1. PROPORTIONAL INTEGRAL DERIVATIVE

PID controller has been widely applied in industrial processes [135, 138]. This controller is usually designed in the frequency domain via Laplace transform instead of time domain. The implementation of PID feedback controller depends on the error  $e(\cdot)$  derived from the difference between reference  $r(\cdot)$  with output  $y(\cdot)$ , which in continuous time reveals

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau + k_d \frac{d}{dt} e(t) \quad (3.22)$$

or in discrete time:

$$u(k) = k_p e(k) + k_i \sum_{i=1}^k e(i)T + k_d \left( \frac{e(k) - e(k-1)}{T} \right) \quad (3.23)$$

where  $T$  is sampling time. The P (Proportional) part of the controller, which is proportional to the current error, has the ability of amending the error. Typically, the higher the proportional constant is, the faster the response is. Yet, only using a P controller may result in steady-state error. To this end, the I part is required for eliminating the latter. However, this will limit the response speed and affect the stability of the system. Generally, the speed of the response will be increased along with the decrease of integral gain. The D part has the capability to incorporate the change of the error and improve the stability of system by compensating the phase lag caused by I part. The combination of the respective parameters setting of this controller significantly influences the system performance, and may even render the system unstable if inappropriate parameters are selected. Thus, before a PID controller is implemented, the optimal combination of such parameters needs to be tuned repeatedly to achieve a desirable performance. The existing methods for parameter tuning include trial and error, process reaction curve technique, Ziegler-Nichols method, etc., [135]. This type of controller is well understood. Yet, it is typically applied in SISO systems and is not designed from the view of minimization of energy consumption. Currently, a research focus on PID is the transition from the classical traditional integer order to fractional order, i.e.  $PI^\lambda D^\mu$  [139] to improve performance.

### 3.4.2. LINEAR QUADRATIC REGULATOR

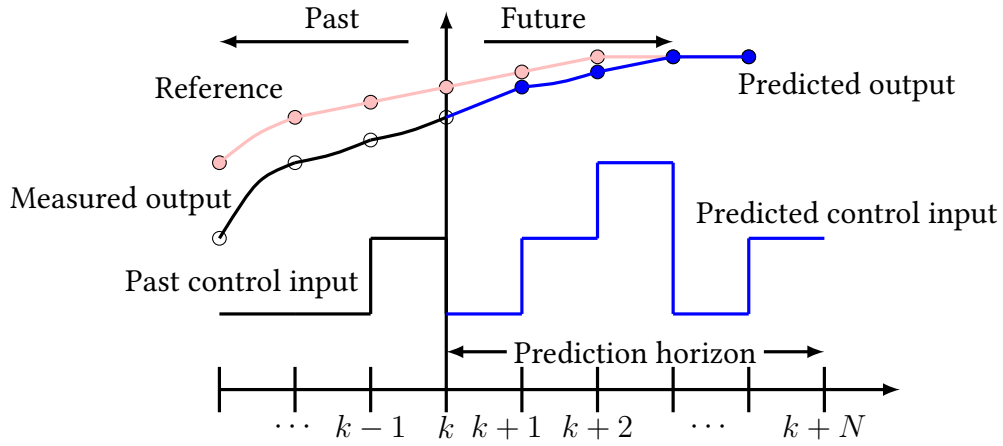
In the field of optimal control theory, Pontryagin's Maximum Principle (PMP) and Dynamic Programming (DP) are the two famous methods [46, 140]. The former is an extension of the well-known calculus of variations. A boundary value optimization problem with constraints is numerically solved by an indirect method, which is termed as "first optimize,

then discretize". The main drawback is that differential equations with strong nonlinearity are difficult to solve and are difficult to handle active inequality constraints [140]. The latter attempts to derive an explicit feedback law off-line. Based on Bellmans principle of optimality, a multi-step optimization problem over a long horizon is split up into many sub-problems. This method provides a comprehensive insight into the mathematical problem and considers all information concerning the optimal control problems. Optimality can be guaranteed by an exhaustive search of all the control and state grids. Yet, it provides an exact solution at the expense of computational complexity and may be difficult to be applied in practice due to the curse of dimensionality. Thus, both approaches may be fine for planning, yet they are computationally intractable for real time control. However, if the cost function is quadratic and system is linear, then the infinite optimization problem can be simplified into solving an Riccati equation, which is known as linear quadratic regulator (LQR). A basic derivation for LQR feedback control law is given in Appendix A.3. LQR is widely applied in the field of optimal control and also provides a way for the design of terminal cost for stability purpose in the framework of MPC [141]. Unfortunately, this method cannot handle constraints.

### 3.4.3. NONLINEAR MODEL PREDICTIVE CONTROL

Model predictive control (MPC) is an control method based on optimization to generate a static state feedback for a possibly nonlinear dynamical system, which is approximately optimal with respect to a given performance index, subject to constraints and used as a closed loop implementation of optimal control [107]. Although the method is demanding from both a computational and modeling point of view, MPC has been widely applied for mechanical or chemical systems [136, 142], inventory management in supply chains [143, 144] and has grown mature over the last decades [107].

Linear MPC has been maturely developed and widely used in the process control for many years. The reasons can be summarized as follows. Firstly, the system identification is relatively easy to be derived based on the measurable process data. Secondly, the linear models exhibit desirable behavior in the neighborhood of the operating point. Due to that, some nonlinear models with weak nonlinearities can be transformed into linear models by linearization for analysis and control. Yet, this method is not always feasible. For instance, batch processes are not staying at the steady-state but in a transient mode for a long time, often frequently influenced by the market demand. That leads to frequent changes in operation points. In this case, a gain scheduling technique can be adopted and the process operation will be divided into a set of operating regions. In each region, the model is identified and the controller is automatically updated [142]. This seamless switching between linear controllers needs to be cautiously implemented. Note that although the system is linear, the closed-loop dynamic by MPC is actually nonlinear due to the constraints. The



**Figure 3.7.:** Principle of Model predictive control

stability of closed-loop system cannot be simply analyzed by poles due to nonlinearity.

Unlike linear model predictive control, nonlinear model predictive control (NMPC) is applicable for both linear and nonlinear systems, linear and nonlinear constraints, and typically requires to solve a nonconvex optimization problem. Since almost all systems in nature are essentially nonlinear, we employ NMPC as a general method. The work principle of this algorithm is intuitively described with Figure 3.7.

For the given performance index  $\ell : X \times U \rightarrow \mathbb{R}_{\geq 0}$ , the idea of MPC is to approximate the solution of the infinite horizon optimal control problem

$$J_{\infty}(x_0, u) = \sum_{k=0}^{\infty} \ell(x(k), u(k)) \quad (3.24)$$

subject to the dynamics (3.1) and constraints  $x \in \mathbb{X}, u \in \mathbb{U}$ . The optimal value function corresponding to (3.24) is given by  $V_{\infty}(x_0) = \inf_{u \in \mathbb{U}} J_{\infty}(x_0, u)$ . Based on the dynamic programming principle we obtain

$$V_{\infty}(x_0) = \inf_{u \in \mathbb{U}} \{ \ell(x_0, u) + V_{\infty}(f(x_0, u)) \}$$

and can derive an optimal feedback control law

$$\kappa_{\infty}(x(n)) = \operatorname{argmin}_{u \in \mathbb{U}} \{ \ell(x(n), u) + V_{\infty}(f(x(n), u)) \}$$

by using Bellman's principle of optimality. Since this optimal control problem is typically computationally intractable and may not be analytically solved for nonlinear systems and / or in the presence of inequality constraints [145], MPC circumvents solving Bellman's optimality equation by truncating the infinite optimization problem with a fixed finite receding time window and iteratively solving an open-loop optimization problem with a given initial conditions, a three step procedure arises: In the first step, the current state of the system is

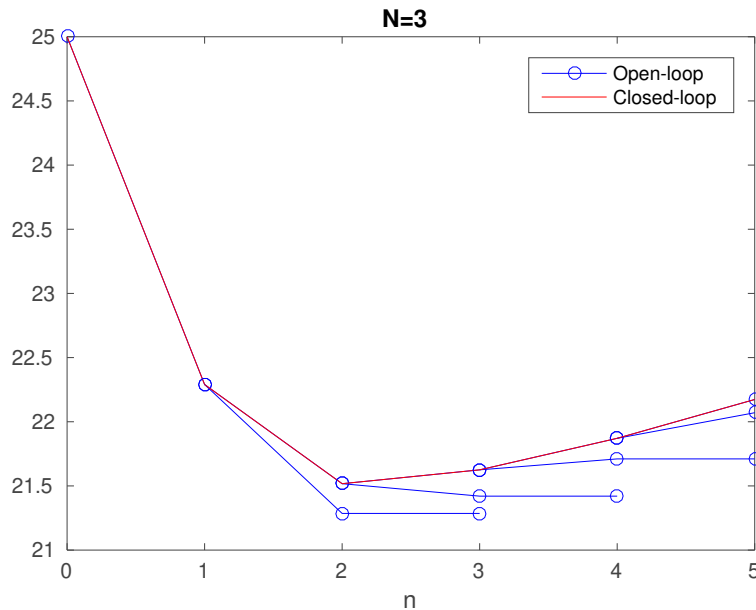
derived. Thereafter, a computationally complex truncated optimal problem

$$\begin{aligned} \min J_N(x_0, u) &= \sum_{k=0}^{N-1} \ell(x(k), u(k)) & (3.25) \\ \text{subject to } & x(k+1) = f(x(k), u(k)), \\ & x(0) = x_0 \\ & x(k) \in \mathbb{X} \forall k \in \{0, \dots, N\} \\ & u(k) \in \mathbb{U} \forall k \in \{0, \dots, N-1\} \end{aligned}$$

with finite prediction horizon  $N$  is solved to obtain a corresponding optimal control sequence. To solve such problems, direct approaches like SQP, IP Methods or heuristics are well established. Moreover, in contrast to the PMP and DP, the NMPC shows reduced computation burden regarding warm start. After solving this optimal control problem (3.25), an optimal open-loop control sequence  $u^*(0), u^*(1), \dots, u^*(N-1)$  and the associated trajectory  $x^*(1), \dots, x^*(N)$  are obtained. For simplicity of exposition, we assume that the minimizer  $u^*(\cdot) := \operatorname{argmin}_{u \in \mathbb{U}} J_N(x_0, u)$  of (3.25) is unique. In the third and last step, only the first element of this control sequence is applied as a feedback control law (i.e.,  $\kappa_N(x) = u^*(0)$ ), rendering the procedure to be iteratively applicable. Then, for a given initial value  $x_0 = x(0)$ , we obtain the closed-loop solution

$$x(n+1) = f(x(n), \kappa_N(x(n))), \quad x(0) = x_0, n \in \mathbb{N}_0. \quad (3.26)$$

An illustrated example concerning the open-loop and closed-loop trajectory is given in



**Figure 3.8.:** Difference between open-loop and closed-loop trajectory

Figure 3.8 with  $N = 3$ . For each time instant, the open-loop trajectory represented by the

blue line has three steps but the closed-loop trajectory highlighted by the red dashed line only has one step <sup>1</sup>, iteratively running the same procedure of the next time instants.

Note that optimality in each iteration is not sufficient to guarantee stability in the sense of Definition 3.16 (e.g., [147]). Therefore, it is crucial to verify the stability of the resulting closed-loop control system. Generally, the classical Lyapunov function is designed off-line to ensure that condition (3.19) holds. In the MPC setting, we can utilize the optimal value function  $V_N(x_0) = J_N(x_0, u^*(\cdot))$  as Lyapunov function. For the optimal control problem (3.25), based on dynamic programming, we obtain that:

$$V_N(x_0) = \inf_{u \in \mathbb{U}} \{\ell(x_0, u) + V_{N-1}(f(x_0, u))\}, \forall x_0 \in \mathbb{X} \quad (3.27)$$

Furthermore, assume  $u^*$  is the attained optimal control sequence, we have

$$J_N(x_0, u^*) = V_N(x_0) = \ell(x_0, u^*(0)) + V_{N-1}(f(x_0, u^*(0))) \quad (3.28)$$

and the MPC closed-loop feedback law is expressed by

$$\kappa_N(x) = \operatorname{argmin}_{u \in \mathbb{U}} \{\ell(x, \kappa_N(x)) + V_{N-1}(f(x, \kappa_N(x)))\}. \quad (3.29)$$

## 3.5. STABILITY AND FEASIBILITY OF NONLINEAR MODEL PREDICTIVE CONTROL

To guarantee the stability of system controlled by MPC, in this section, we introduce two classes of stability methods: terminal endpoint constraints, terminal cost and terminal region [141, 148, 149, 150], as well as without terminal conditions [107, 126, 130, 133, 151, 152].

### 3.5.1. TERMINAL ENDPOINT CONSTRAINTS

The simplest way is to modify the original optimization problem (3.25) is to impose the stabilizing terminal endpoint constraints  $x(N) = x^*$  [148]. To show stability we require the following assumptions hold.

#### Assumption 3.23

For  $x^* \in \mathbb{X}$  there exists  $u^* \in \mathbb{U}$  such that  $f(x^*, u^*) = x^*$ .

#### Assumption 3.24

The stage cost  $\ell : X \times U \rightarrow \mathbb{R}_{\geq 0}$  satisfies  $\ell(x^*, u^*) = 0$  and  $\ell(x, u) > 0$  for all  $u \in \mathbb{U}$  if  $x \neq x^*$ .

---

<sup>1</sup>The one step feedback law is used in this thesis. Regarding the multistep feedback law, we refer to [146].

**Theorem 3.25**

If the viability Assumption 3.23 and the Assumption 3.24 for the reversed  $V_{N-1}(x) \geq V_N(x)$  hold, and if there exist functions  $\alpha_1, \alpha_2$  and  $\alpha_3 \in \mathcal{K}_\infty$  such that

$$\alpha_1(\|x - x^*\|) \leq V_N(x) \leq \alpha_2(\|x - x^*\|) \quad (3.30)$$

$$\alpha_3(\|x - x^*\|) \leq \inf_{u \in \mathbb{U}} \ell(x, u) \quad (3.31)$$

hold, then  $\kappa_N$  is asymptotically stabilizing. In addition, the performance estimate

$$J_\infty(x, \kappa_N) \leq V_N(x)$$

holds for  $x \in \mathbb{X}_N$ , where  $\mathbb{X}_N := \{x \in \mathbb{X} \mid \mathbb{U}^N(x) \neq \emptyset\}$ .

The proof can be found in [107].

**3.5.2. TERMINAL COST AND TERMINAL REGION**

In order to alleviate the requirement of the strictly terminal equality constraints and reduce computational burden, a dual mode control was designed. That is, the MPC controller is only implemented outside of the neighborhood of the equilibrium point. A local feedback controller is used for the system once the state reached the neighborhood of the equilibrium point. Using this kind of method requires switching the controllers [153]. Further, a terminal cost and a terminal region were designed to guarantee the stability, which was termed as quasi-infinite optimization method [141, 154]. The idea is, that the designed local controller is never applied but only used to construct a local Lyapunov function  $F(x(N))$  such that the optimal value function is decreasing. The modified optimization problem is given as:

$$\begin{aligned} \min J_N(x_0, u) &= \sum_{k=0}^{N-1} \ell(x(k), u(k)) + F(x(N)) & (3.32) \\ \text{subject to} \quad &x(k+1) = f(x(k), u(k)), \\ &x(0) = x_0 \\ &x(k) \in \mathbb{X} \quad \forall k \in \{0, \dots, N\}; x(N) \in \mathbb{X}_F \subset \mathbb{X} \\ &u(k) \in \mathbb{U} \quad \forall k \in \{0, \dots, N-1\} \end{aligned}$$

**Assumption 3.26**

For each  $x \in \mathbb{X}_F$ , there exists a terminal control sequence  $u_F$ , such that  $f(x, u_F) \in \mathbb{X}_F$ ,  $F : \mathbb{X}_F \rightarrow \mathbb{R}_{\geq 0}$  such that the following inequality holds

$$F(f(x, u_F)) + \ell(x, u_F) - F(x) \leq 0 \quad (3.33)$$

**Theorem 3.27**

Consider the optimization problem (3.32), if the Assumption 3.26 holds with resulting  $V_{N-1}(x) \geq V_N(x)$ , and if there exist functions  $\alpha_1, \alpha_2$ , and  $\alpha_3 \in \mathcal{K}_\infty$  such that (3.30), (3.31) hold, then the nominal system is asymptotically stable with the associated closed-loop feedback law  $\kappa_N$ . Additionally, for each  $x \in \mathbb{X}_N$ , the inequality  $J_\infty(x, \kappa_N) \leq V_N(x)$  holds.

Compare to using terminal endpoints constraints only, this method may reduce the computational burden with possible shorter prediction horizon. Yet, the additional analytical effort for offline computing the terminal region is challenging and still needs a large prediction horizon to guarantee large feasible sets. Due to the modification of the original control problem for stability, using terminal conditions are not often used in industry due to long horizons and excessively increasing computational requirements [155]. Regarding the stability constraint methods, we refer to the excellent surveys [149, 150].

**Remark 3.28** (Feasibility)

Similar to the zero terminal constraints, feasibility is automatically ensured when the initial states and control sets are feasible [149]. Yet, the required prediction horizon for feasibility is typically unknown and needs to be checked iteratively.

**3.5.3. WITHOUT TERMINAL CONDITIONS**

Considering problem (3.25) without additional terminal cost and/or terminal constraints, we introduce the so-called unconstrained MPC scheme which may exhibit a larger operation region, alleviate the computational burden and improve the system performance [107, 130, 151]. Yet, this method needs the assumption of asymptotically controllability with respect to the cost functional or running cost. The stability can be guaranteed with a sufficient large prediction horizon [151, 156].

**Theorem 3.29**

Consider  $f : X \times U \rightarrow X$ , a running cost  $\ell : X \times U \rightarrow \mathbb{R}_{\geq 0}$  and an optimal value function  $V_N : X \rightarrow \mathbb{R}_{\geq 0}$ . If there exists an admissible feedback control law  $\kappa_N : X \rightarrow U$  and  $\alpha \in (0, 1]$ ,  $N \in \mathbb{N}$  such that the relaxed Lyapunov inequality [157]

$$V_N(x) \geq \alpha \ell(x, \kappa_N(x)) + V_N(f(x, \kappa_N(x))) \quad (3.34)$$

holds and there exists  $\alpha_1, \alpha_2, \alpha_3 \in \mathcal{K}_\infty$  such that (3.30), (3.31) hold, then the MPC closed-loop control system is asymptotically stable. In addition, we obtain:

$$\alpha V_\infty(x) \leq \alpha V_\infty^{\kappa_N}(x) \leq V_N(x) \leq V_\infty(x) \quad (3.35)$$

**Lemma 3.30**

Consider function  $V_N : X \rightarrow \mathbb{R}_{\geq 0}$ ,  $f : X \times U \rightarrow X$ , a feedback law  $\kappa_N : X \rightarrow U$ ,  $N \in \mathbb{N}$ ,

$\alpha \in (0, 1]$ , and  $\ell : X \times U \rightarrow \mathbb{R}_{\geq 0}$ . If

$$V_N(f(x, \kappa_N(x))) - V_{N-1}(f(x, \kappa_N(x))) \leq (1 - \alpha)\ell(x, \kappa_N(x)) \quad (3.36)$$

holds for all  $x \in \mathbb{X}$ , then the relaxed Lyapunov inequality (3.34) holds.

**Assumption 3.31** (Asymptotic controllability)

Assume there exist  $\gamma > 0$  such that

$$V_2(x) \leq (\gamma + 1)V_1(x) \quad \text{and} \quad (3.37)$$

$$V_k(x) \leq (\gamma + 1)\ell(x, \kappa_k(x)), k = 3, \dots, N \quad (3.38)$$

hold for all  $x \in \mathbb{X}$ , then (3.36) holds.

**Proposition 3.32**

Consider  $N \geq 2$  and suppose Assumption (3.31) holds. Then the following inequality holds:

$$\frac{(\gamma + 1)^{N-2}}{(\gamma + 1)^{N-2} + \gamma^{N-1}} V_N(x) \leq V_{N-1}(x) \quad (3.39)$$

The proof of Theorem 3.29 and Lemma 3.30 and Proposition 3.32 were given in [156].

**Proposition 3.33**

Consider  $\gamma > 0$  with  $N \in \mathbb{N}$  such that  $(\gamma + 1)^{N-2} - \gamma^N > 0$ . If Assumption 3.31 holds, then the inequality

$$V_{\infty}^{\kappa_N}(x) \leq \overbrace{\frac{(\gamma + 1)^{N-2}}{(\gamma + 1)^{N-2} - \gamma^N}}^{\frac{1}{\alpha}} V_{\infty}(x) \quad (3.40)$$

holds with the derived  $\alpha$  and the relate accuracy estimate

$$\frac{V_{\infty}^{\kappa_N}(x) - V_{\infty}(x)}{V_{\infty}(x)} \leq \frac{\gamma^N}{(\gamma + 1)^{N-2} - \gamma^N} \quad (3.41)$$

The proof is briefly given in Appendix A.2. In order to ensure the derived  $\alpha \in (0, 1]$ , the prediction horizon  $N$  is chosen sufficiently large such that  $(\gamma + 1)^{N-2} - \gamma^N > 0$ , i.e.,

$$N > 2 \frac{\log(\gamma + 1)}{\log(\gamma + 1) - \log \gamma}. \quad (3.42)$$

Through the above analysis for the stability of unconstrained MPC, the key point is to check the asymptotic controllability Assumption 3.31 with respect to the optimal value function. Yet, Assumption 3.31 is hard to check. There are two alternatives to adapt it, one is using exponential controllability condition (3.43). Another is to only consider the points that



visited by the closed-loop trajectory  $x(n)$  to estimate the degree of suboptimality  $\alpha$  online [130]. We would like to point out that the latter provides a way for the design of an adaptive horizon MPC based on the derived online estimate of suboptimality degree. We will discuss this method in conjunction with a flexible Lyapunov function in Section 6.2.6 which allows for a local increases but with an ensured decrease in future steps.

**Proposition 3.34** (Exponential controllability)

Assume there exists a function  $W : X \rightarrow \mathbb{R}_{\geq 0}$ , a constant  $h > 0$ . For all  $x \in X$  and all  $u \in U$ ,  $\ell(x, u) \geq hW(x)$  holds. In addition, assume there exists a control sequence  $u^* \in \mathbb{U}$ , a constant  $C > 0$ ,  $\sigma \in [0, 1)$  such that

$$\ell(x(n), u^*(n)) \leq C\sigma^n W(x) \quad (3.43)$$

holds along with corresponding solution  $x(n)$  and  $x(0) = x$ . Then  $\gamma = \frac{C}{h(1-\sigma)} - 1$ .

Regarding the proof, we refer to [156].

**Remark 3.35**

Note that actually the  $u$  in (3.43) does not need to be optimal.

Alternatively, we can use linear program to compute the degree of suboptimality [107].

**Assumption 3.36**

For each initial value  $x \in \mathbb{X}$ ,  $\ell^*(x) := \inf_{u \in \mathbb{U}} \ell(x, u)$ , assume there exist suitable functions  $\mathcal{J}_1, \mathcal{J}_2 \in \mathcal{K}_\infty$ , such that

$$\mathcal{J}_1(\|x - x^*\|) \leq \ell^*(x) \leq \mathcal{J}_2(\|x - x^*\|) \quad (3.44)$$

holds. In addition, if there exists an admissible control  $u \in \mathbb{U}$ , such that

$$\ell(x(k), u(k)) \leq C\sigma^k \ell^*(x), k = 0, 1, \dots \quad (3.45)$$

holds with given real overshoot constants  $C \geq 1$  and decay rate  $\sigma \in (0, 1)$ , then

$$V_N(x) \leq \gamma_N \ell^*(x) \quad \text{with} \quad \gamma_N = \sum_{k=0}^{N-1} C\sigma^k = C \frac{1 - \sigma^N}{1 - \sigma}. \quad (3.46)$$

**Lemma 3.37**

If Assumption 3.36 holds, then the performance index  $\alpha_N$  depending on horizon  $N$  can be computed via

$$\alpha_N = 1 - \frac{(\gamma_N - 1) \prod_{k=2}^N (\gamma_k - 1)}{\prod_{k=2}^N \gamma_k - \prod_{k=2}^N (\gamma_k - 1)}. \quad (3.47)$$

The technical proof regarding the compute of  $\alpha_N$  can be found in [Chapter 6 [107]] or in [158].

**Theorem 3.38**

*If Lemma 3.37 holds, then there exists a  $\alpha \geq \alpha_N \in (0, 1]$  satisfying (3.34). In addition, if there exists  $\alpha_1, \alpha_2, \alpha_3 \in \mathcal{K}_\infty$  such that (3.30), (3.31) hold, then the nominal closed-loop system controlled by NMPC feedback law  $\kappa_N$  is asymptotically stable with the optimization  $N \in \mathbb{N}$ . Also, the performance estimate holds for each  $x \in \mathbb{X}$ ,  $V_\infty^{\kappa_N}(x) \leq V_N(x)/\alpha \leq V_\infty(x)/\alpha$ .*

The proof can be found in [107].

**Remark 3.39** (Feasibility)

*Recursive feasibility for unconstrained MPC can be expected on the viability kernel for a sufficient large  $N$ , and under additional assumptions if the state constraint set  $\mathbb{X}$  is not viable, see [Theorem 7.20 [107]] for details. If  $\mathbb{X}$  is viable, then recursive feasibility may be inherited from optimality and stability of closed-loop system.*

## 3.6. SUMMARY

*In this chapter, we first introduced some basic concepts concerning optimization, control and stability. Further, we introduced three common applied controllers, and emphasized on the importance of MPC and presented with and without terminal conditions for stability guarantee. Regarding to the detailed comparison of these two categories of stabilizing methods, we refer to [Chapter 7 [107]].*

# 4

## MODELING OF A JOB SHOP SYSTEM WITH RMTs FOR WIP CONTROL

Fluctuations in demand along with requirements of low cost, high quality, individual customization and short lead time render manufacturing systems more complex and dynamical [159]. Manufacturers expect the production process to be as stable as possible and robust against disturbances to avoid significant performance deterioration [160]. RMTs can provide flexible and robust capacity adjustment to compensate for unpredictable events and adapt to different operating conditions in a cost-effective way. In order to render the production system to be more productive and predictive, the critical task is to analyze the dynamic behavior of manufacturing processes with RMTs. Thus, an appropriate mathematical model and corresponding control strategies are prerequisites for the comprehension of the complexity of a manufacturing system under either static or dynamic circumstances [57].

Generally, there are two categories of approaches to build a mathematical model. One is the experimental approach that deliberately applies known collected input data (e.g., manipulated variables, disturbance) and the observed state or output data (e.g., WIP) to generate a model by simulation. The other uses an analytical approach, in which the model is built based on the adequately understanding of the dynamic behavior of the physical process, which is basically described by difference (differential) equations. Such modelling of production-inventory systems was effectively demonstrated for both discrete time and continuous time domains [161]. In this thesis, we adopt the former and focus on state space models, which are discrete in time but continuous in states, possibly nonlinear and subject to control inputs and disturbances (Section 4.2). Additionally, we proceed with a deep look into the property of RMTs in terms of their inherent uncertain reconfiguration delays which influence the dynamics and performance of the system (Section 4.3).

Parts of this chapter have been published in [6, 162].

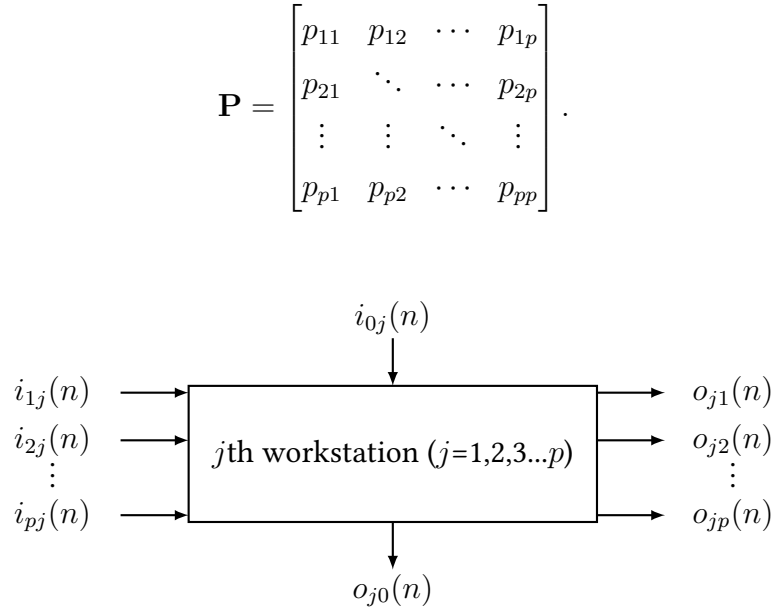
## 4.1. ASSUMPTIONS FOR ESTABLISHING MODEL

In this thesis, manufacturing systems with a high percentage of parallel machines are of particular interest, which is one characteristics of a job shop system. Due to small lot sizes and a large variety within series, respective companies will profit from the usage of reconfigurable machine tools and corresponding control strategies. Since high volume products can be manufactured cost efficiently with a high productivity by means of dedicated machines, and diversity of customized product at a low quantity can be effectively guaranteed via reconfigurable machines, we consider both of RMTs and DMTs for each workstation. This kind of combination and co-existence in industrial practice is natural and frequent [163]. Given the planned investment of RMTs and DMTs, we focus on the operational layer and aim to improve shop floor control performance through internal intended capacity adjustment via assignment of RMTs. Note that in the reconfiguration stage, the system capacity decreases with loss of productivity. If the reconfiguration time is prolonged too long, the system may be rendered unstable inducing a performance degradation. The reconfiguration time is therefore expected to be as short as possible to recover the operating functionality to complete the specific tasks. For such a setting, the following assumptions are reasonable and used within the thesis:

- A 1. The percentage of reconfigurable machines is fixed and limited.
- A 2. The capacity of RMTs is in excess or redundant to achieve robustness against disturbance via flexible capacity adjustment [164].
- A 3. All RMTs can perform the tasks for all workstations, but is only allowed to operate one specific task at a time.
- A 4. Any reconfiguration is completed in less then two hours [11].
- A 5. All capacities of buffer are limited.
- A 6. The transportation time between workstations is negligible.
- A 6. The sequencing policy is First In First Out (FIFO).

## 4.2. DISCRETE STATE SPACE MODEL FOR WIP CONTROL

We consider a simple flow model of a job shop system with  $p$  workstations. The job shop is given by a fully connected graph  $\mathcal{G} = (V, P)$ , where the set of vertexes  $V = \{1, \dots, p\}$  represents the workstations and  $P$  the  $p \times p$  flow probability matrix between the workstations.



**Figure 4.1:** *j*th workstation in multi-workstation production system

Figure 4.1 shows the relationship between inputs and outputs for one workstation in a manufacturing system. The  $j$ th workstation can receive tasks from the initial stage or any of the  $p$  workstations, and then send its products to any workstation or the final stage.  $i_{1j} \cdots i_{pj}$  represent the input rates of each workstation to workstation  $j$ ,  $i_{0j}$  represents the rate of order release to workstation  $j$ .  $o_{j0}, o_{j1} \cdots o_{jp}$  represent the respective output rates. Each of the workstations features  $n^{DMT}$  identical DMTs, which may operate with production rate  $[0, r^{DMT}]$ . The number of RMTs is controlled by the input variable  $u$  and each RMT may operate with production rate  $[0, r^{RMT}]$ . The work in process level  $WIP_j$  is defined as the number of orders waiting to be processed at workstation  $j$ , hence its rate of change is given by the difference between rates of input and output orders  $I$  and  $O$ . Based on Figure. 4.1, we obtain

$$I_j := \sum_{l=0}^p i_{lj} \quad \text{and} \quad O_j := \sum_{l=0}^p o_{jl}.$$

To link outputs to the inputs, we impose the following:

**Assumption 4.1** (Flow conservation)

The job shop system is mass conservative, that is for given flow probability matrix  $P$  we have

$$I_j(n) = \sum_{l=1}^p p_{lj} O_l(n)$$

for each workstation  $j$  with  $p_{lj} \geq 0$  for all  $l = 1, \dots, p$ .

We like to note that Assumption 4.1 is appropriate here for two reasons: First, loss of products within the job shop system can be included by modifying the flow probabilities between the workstations. And secondly, the assumption rules out dissipation, which similar to friction in a mechanical system eases the task of stabilizing a system. Hence, Assumption 4.1 represents the more difficult case and all following results also apply to the case with dissipation.

**Assumption 4.2** (Operational layer)

The order release rates  $i_{0j}(n)$  to each workstation  $j$  are determined externally and must therefore be considered as disturbances  $d_j(n) := i_{0j}(n)$ .

Utilizing Assumptions 4.1 and 4.2, the time dependent dynamics of  $WIP_j$  can be evaluated via its previous value, the inputs from other workstations, the difference between self-loop input with output of itself, and the disturbance, i.e.

$$\begin{aligned} WIP_j(n+1) &= WIP_j(n) + I_j(n) - O_j(n) \\ &= WIP_j(n) + i_{0j}(n) + \sum_{l=1}^p p_{lj} O_l(n) - O_j(n) \\ &= WIP_j(n) + \sum_{\substack{l=1 \\ l \neq j}}^p p_{lj} O_l(n) + (p_{jj} - 1) \cdot O_j(n) + d_j(n). \end{aligned} \quad (4.1)$$

Here, the orders are used as the dependent variable rather than hours of work content, and the production rate is given in orders per hour [59].

Relying on the capability of flexible changing capacity via RMTs, we like to include it into the workstations. Thus, we link system (4.1) to the number of machine tools operating within the workstations. Note that from an economic point of view it only makes sense to buy new machinery if the current capacity is insufficient to deal with all orders. This typically leads to high  $WIP$  levels, which allow us to rewrite the output as

$$O_j(n) = n^{DMT} r^{DMT} + u(n) r^{RMT} \quad (4.2)$$

Such fidelity of regulating  $WIP$  by control-theoretic approach would decrease at the extreme operating conditions, e.g., at low  $WIP$  level, or requirement of large capacity adjustment [61]. Our idea is to guarantee such fidelity of a feedback, i.e. to ensure that all workstations operate close to predefined  $WIP$  levels. If this property can be shown for a feedback at hand, then the assumption of a high  $WIP$  level can be shown to hold. More formally, the latter assumption reads:

**Assumption 4.3** (High  $WIP$  level)

At any time instant  $n$ , the  $WIP$  levels in all workstations are at least as high as the total

machine capacity, i.e. (4.2) holds.

To prove that Assumption 4.3 holds, one way is to show that there exists a feedback such that the closed loop renders the set  $y = \{(WIP_1, \dots, WIP_p) | WIP_j \geq O_j, j = 1, \dots, p\}$  is a control forward invariant of (4.1). To this end, let  $x = (WIP_1, \dots, WIP_p) \in \mathbb{X} \subset X$  represent the WIP level of all workstations and  $u = (u_1, \dots, u_p) \in \mathbb{U} \subset U$  denote the vector of RMTs assigned to all workstations. Here, the sets  $\mathbb{X}$  and  $\mathbb{U}$  allow us to incorporate possibly wanted constraints on the WIP level and the total number of RMTs  $\sum_{j=1}^p u_j(n) \leq m \in \mathbb{N}_0$ . Utilizing Assumption 4.3, then from (4.1) we obtain

$$\begin{aligned} x(n+1) &= x(n) + P \cdot \left( n^{DMT} r^{DMT} + u(n) r^{RMT} \right) + d(n) \\ &=: f(x(n), u(n), d(n)). \end{aligned} \quad (4.3)$$

The aim is to control the number of RMTs to balance the capacity in the job shop system to meet the required demand and maintain WIP level to be a desired level.

#### Corollary 4.4

Consider system (4.1) and a predefined reference value  $x_j^*$ . If for the control  $u(\cdot)$  a set  $Y \subset \mathbb{X}$  is forward invariant such that

$$y \geq n^{DMT} r^{DMT} + u(n) r^{RMT} \quad \forall y \in Y, n \in \mathbb{N}_0 \quad (4.4)$$

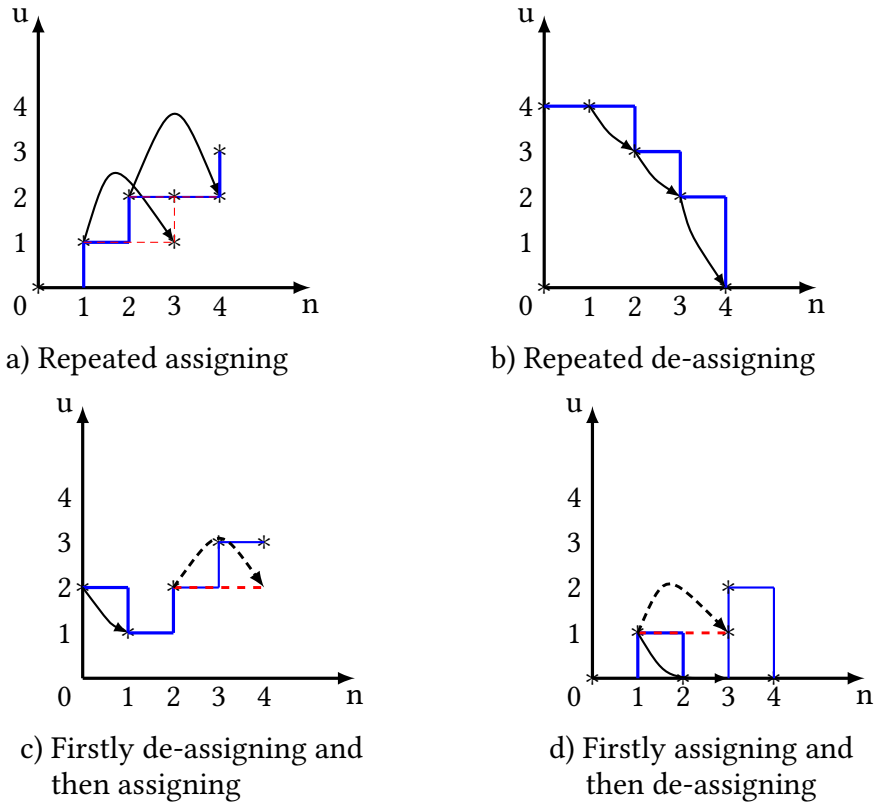
holds, then Assumption 4.3 holds.

In practical terms, inequality (4.4) ensures that for any chosen time instant  $n$  the WIP level is high enough such that all machine tools within a workstation work at full capacity. Forward invariance in turn means the control  $u$  ensures, that the buffers of all workstations will be refilled such that full capacity utilization in the next time step is guaranteed. Actually, these high load situations are very common even under the assumptions of stationarity and moderate utilization, especially in job shop systems [12].

### 4.3. A GENERAL RECONFIGURATION RULE

Note that in (4.3), the input variable  $u$  does not contain the reaction time referring to the reconfiguration delay. Typically, the reconfiguration effort consists of reconfiguration time, ramp-up time and reconfiguration cost [74]. Here, the combination of the former two is considered as reconfiguration delay, which is taking place by system reconfiguration to re-arrange an equipment to compensate the capacity. That is, the capacity may not be compensated instantaneously due to such delay until the reconfiguration completed, the reconfiguration is regarded as an interruption to the original production activities with the resulting production loss at some extent in the reconfiguration phase.

We like to note that the trigger of reconfigured RMTs for each workstation is subject to the respective adjacent control states. That is, the reconfiguration is only triggered if the required number of RMTs is higher than the previous value, i.e.  $u_j(n) > u_j(n - 1)$  in, e.g., case a) of Figure 4.2. Consequently, the completion of such a reconfiguration is delayed



**Figure 4.2.:** Illustrate example of reconfigurable rule [6]

by  $r_d$ . If the required number of RMTs is lower than or equal to the previous value (i.e.  $u_j(n) \leq u_j(n - 1)$ ), then there is no reconfiguration delay. Algorithm 1 summarizes this behavior, which is illustrated in Figure 4.2. We like to point out that the cases c) and d) are interesting and complex due to their uncertainties within the reconfiguration. In case c), the solid line represents the utilized RMTs is decreasing, the dashed line shows that the triggered reconfiguration starts at time instant 2 and ends at time instant 4. In this case, the question is whether the reduced RMTs may be already reconfigured to other workstations. If more RMTs are needed for their own workstations, then this workstation has to wait for available RMTs due to the reconfiguration delay. In case d), the dashed line represents a request for reconfiguration, which will be completed at time instant 3. However, the solid line shows that no reconfiguration is necessary as at time instant 3 the updated required number of RMTs is identical to the value at time instant 0. The system becomes much more complicated with the resulting relevant effects derived from the reconfiguration.

In the above, we consider the WIP as the state variable. Alternatively, we could use a coordinate transformation and set the total input orders  $x^i$  and output orders  $x^o$  as state



**Algorithm 1** Reconfiguration Rule in the General Case

---

**Require:** Time instant  $n$ , planned input  $u_j(n)$  and previously applied input  $u_j(n-1)$ , reconfiguration delay  $r_d$  and buffer  $\tau_j(\cdot)$

- 1: **if**  $u_j(n-1) = \emptyset$  or  $\tau_j(\cdot) = \emptyset$  for some  $j \in \{1, \dots, p\}$  **then**
- 2:   **for**  $j = 1$  to  $p$  **do**
- 3:     Set  $u_j(n-1) = 0$
- 4:     **for**  $k = 1$  to  $r_d$  **do**
- 5:       Set  $\tau_j(k) = 0$
- 6:     **end for**
- 7:   **end for**
- 8: **else**
- 9:   **for**  $j = 1$  to  $p$  **do**
- 10:    **if**  $u_j(n) > u_j(n-1)$  **then**
- 11:      $\tau_j(r_d) = u_j(n)$
- 12:    **else**
- 13:     **for**  $k = 0$  to  $r_d$  **do**
- 14:        $\tau_j(n+k) = u_j(n)$
- 15:     **end for**
- 16:    **end if**
- 17:    Set  $u_j(n) = \tau_j(0)$
- 18:    **for**  $k = 0$  to  $r_d - 1$  **do**
- 19:      $\tau_j(k) = \tau_j(k+1)$
- 20:    **end for**
- 21:   **end for**
- 22: **end if**

**Output:** Applied input  $u_j(n)$  and updated buffer  $\tau_j(\cdot)$

---

variables and derive the associated state space model (4.5).

$$\begin{pmatrix} \mathbf{x}^i(k+1) \\ \mathbf{x}^o(k+1) \end{pmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{pmatrix} \mathbf{x}^i(k) \\ \mathbf{x}^o(k) \end{pmatrix} + \begin{bmatrix} \mathbf{P}^\top \\ \mathbf{I} \end{bmatrix} \mathbf{u}(k) + \begin{bmatrix} \mathbf{i}_0(k) \\ \mathbf{0} \end{bmatrix} \quad (4.5)$$

$$\mathbf{WIP}(k) = \begin{bmatrix} \mathbf{I} & -\mathbf{I} \end{bmatrix} \begin{pmatrix} \mathbf{x}^i(k) \\ \mathbf{x}^o(k) \end{pmatrix} + \mathbf{d}(k) \quad (4.6)$$

In doing so, we could intuitively compute the number of total input and output orders along with the calculation of lead time by finding the integer  $\iota$  that make  $x^i(k-\iota) \leq x^o(k) < x^i(k-\iota+1)$  hold [165]. The details of this modelling can be found in [2].

## 4.4. SUMMARY

In order to better understand the fundamental dynamic behavior in both transient and steady-state phases, we use an analytical approach and built a discrete time state space

model of a job shop system with RMTs to maintain WIP to the planned level. Taking reconfiguration delays into account, we formulate a general reconfiguration rule (Algorithm 1) to reflect the dynamics with the implemented control actions. Note that this scheme is typically applicable for those controllers in which the control action is based on the past and current informations only (e.g., PID). As a consequence, the closed-loop performance may deteriorate with the lagged control due to reconfiguration delays. Such chain reactions caused by reconfiguration delays can be naturally considered in the framework of MPC via prediction of future trajectory, which allows us to make appropriate decisions in advance to reduce the negative effects. In next section, we will show how to deal with such delays in the framework of MPC.

# 5

## OPTIMIZATION AND PLANNING TO MAINTAIN WIP VIA RMTs

In Chapter 4, we introduced a job shop model with RMTs for maintaining WIP. Yet, the re-configuration delays render the dynamics to be complex. Controlling such reconfiguration problems is a building block for feedback control. This is similar to fault-tolerant control, which also refers to reconfiguration problems that can be addressed by the respective re-configuration approaches, e.g., fault hiding, optimization based control schemes, learning control, etc. [166]. MPC is a control method based on optimization, which allows to directly integrate the complex dynamics, constraints and performance indexes through solving an optimization problem. Hence, it is readily applicable to balance capacity and loads in job shop systems via RMTs in the presence of demand fluctuations. The theoretical part of this algorithm was introduced in Section 3.4.3. The aim of this chapter is to integrate MPC with integer programming for resolving the assignment of RMTs in accordance with the formulated reconfiguration rules. This chapter is structured as follows: In Section 5.1, based on MPC, we propose three integer control strategies to solve the integer assignment – floor operator, B&B and GA. Further, in Section 5.2, we formulate a tailored reconfiguration rule for controlling WIP (Algorithm 3) in the framework of MPC, which can explicitly deal with constraints including reconfiguration delays through solving an open-loop optimization problem. Also, we present a rule (Algorithm 4) in Section 5.2.2 for truly counting times of reconfiguration, which is helpful when production cost is considered as one of the performance indicators.

Parts of this chapter have been published in [6].

### 5.1. INTEGER PROGRAMMING WITHIN MODEL

#### PREDICTIVE CONTROL

In the framework of MPC, the reconfiguration delay outlined in Algorithm 1 needs to be integrated in the optimization problem accordingly to truly reflect the dynamics. This can be achieved by applying Algorithm 1 in each step of the optimizer to the entire sequence

$u(\cdot)$ . Details will be presented after the introduction of the following integer operators. Combined, we obtain the modified MPC method outlined in Algorithm 2. Note that if the state  $x$  is not available through direct measurement or the measurement is inaccurate due to random noise, we can use its estimated value  $\hat{x}$  derived from Kalman filter or moving horizon estimation [167] as the actual initial value in practice.

---

**Algorithm 2** Basic model predictive control method
 

---

- 1: **Given**  $N \in \mathbb{N}$ .
  - 2: **for**  $n = 0, \dots$  **do**
  - 3:   Measure current WIP levels  $\hat{x}(n)$  and set  $x_0 := \hat{x}(n)$
  - 4:   Compute control inputs  $u(n)$  by solving optimal control problem (3.25)
  - 5:   Apply  $\kappa_N(\hat{x}(n)) = u^*(0)$  to workstations
  - 6: **end for**
- 

While Algorithm 2 is correct from a control point of view, we still need to ensure that the number of RMTs in each workstation is a positive integer and the sum of number of RMTs in the system is limited and fixed. Since a discrete time optimal control problem with constraints can be transformed into a standard nonlinear programming problem by discretization, most MPC solvers utilize continuous optimization routines such as SQP or IP methods. However, problem (3.25) requires a mixed integer routine [115]. This simple additional constraint renders the optimization problem to be NP-hard, and the computational burden will be increased exponentially along with the increase of the prediction horizon and the system dimension.

Integer programming is a kind of optimization problem that requires all of the variables to be integers. This type of problem has a significance in reality, e.g., to optimally allocate available resources. To address the integer MPC control problem, we propose three different approaches. The straightforward way is to completely neglect the integer property within the optimization problem, i.e., to consider  $\mathbb{U} \subset \mathbb{R}_{\geq 0}^p$ . This change allows us to utilize fast continuous optimization approaches which typically results in a non-integer optimal control sequence  $u^*(\cdot)$ . To ensure the integer property, the floor operator can be employed after the line 4 of Algorithm 2 to round the solutions down to the nearest integers, i.e.,  $u^*(0) = \lfloor u^*(0) \rfloor$ . Note that due to its simplicity this method is practicable and widely applied in the design of controllers. Yet, due to the truncated control effort, the derived solution may significantly differ from the optimal one. As a result, this method may induce performance deterioration, tardy convergence and even render the system to be unstable.

To address the latter issues, we combine integer programming within the MPC to resolve the assignment of RMTs. Typical examples of such methods are of deterministic (e.g., B&B) and meta-heuristic nature (e.g., GA).

### 5.1.1. BRANCH AND BOUND

Branch and bound (B&B) is a widespread deterministic method to address combinatorial optimization problems, cf., e.g., [168]. The integer property of variables is firstly relaxed to obtain a continuous one, which can be solved by using standard techniques, e.g., SQP. Then, the variables are fixed one by one generating a solution tree. For each problem, upper and lower bounds are derived, which allows to cut branches and speedup the solution process. Due to cutting, the method is much faster than full enumeration. In contrast to the floor operator, it allows to integrate both a performance criterion and constraints and yields an optimal solution. In [169], the authors integrated an SQP solver with a basic branch and bound to solve the integer nonlinear programming problems. The core idea is to possibly early branch the search-tree after a single QP iteration of the SQP solver. In the MPC context, this optimality property allows to employ standard techniques for showing stability of the closed loop system [107]. The combination of MPC with B&B was applied on a variety of applications, such as collision avoidance, multilevel cascaded H-Bridge static synchronous compensator, building cooling supply control, etc. [170, 171, 172]. Due to the branching and cutting approach, it is less sensitive to the initial guess than standard solvers for continuous problem. Yet, it still suffers from the curse of dimensionality, which leads the method to be computationally intractable for large scale problems.

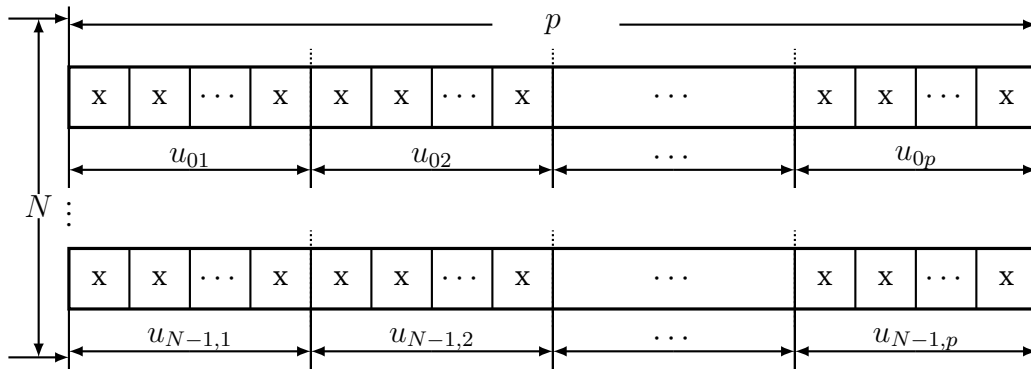
### 5.1.2. GENETIC ALGORITHM

To overcome the scalability issue, different types of meta-heuristics were devised such as particle swarm optimization (PSO) and genetic algorithm (GA) [123]. In contrast to analytical methods, meta-heuristics utilize techniques typically based on random search and educated guesses. As a result, the results heavily depend on their configuration and may not be optimal and even vary when applied repeatedly. Nevertheless, heuristics methods were widely applied in practice, especially in the case that an optimal solution cannot be found in a reasonable time by deterministic methods while solving a large class of optimization problems. Hence, it is necessary to balance the trade-off between optimality and computational burden, cf., [173] that combine exact method with meta-heuristics to solve an NP-hard optimization problem.

Due to its algorithm simplicity and fast search capability, PSO was applied for integer programming in [174] and combined with NMPC on an FPGA for fast optimization [175]. Yet, this algorithm merely considers the individual particles in the specified range in the optimization routine but neglects the relationship of mutual restraints. Although this disadvantage can be addressed by adding a penalty function, it may easily fall into a local optimum. Here, we consider using GA, which starts with a set of initial random solutions called population, cf., Figure 5.1. Then, it applies a process, which is analog to biological evolution, to find an approximately optimal solution through selection, crossover and mu-

tation operators iteratively [176]. In particular, we consider the implementation presented in [177]. We like to point out that :

- The length of a chromosome is depends on the length of the prediction horizon and system dimension, cf., Figure 5.1. Note that  $u_{N-1,p}$  represents the  $p_{th}$  element in the  $N - 1_{th}$  column of open-loop control sequence.
- The reconfiguration rule – Algorithm 3 will be incorporated in the evaluation of fitness function.



**Figure 5.1.:** Illustration of chromosome coding of GA within MPC setting

### 5.1.3. STRATEGIES FOR REDUCING COMPUTATIONAL COST

Since machine configuration involves solving an NP-hard optimization problem, this aggravates the online computational burden and may be computationally intractable. Considering the exponential computational load for large prediction horizons and/or system dimension, we may not obtain the best solution in a reasonable time. In case of fast NMPC implementation, the exact optimal solution from the nonlinear programming is not necessary and can be replaced with a suboptimal solution [178].

To reduce the computation load, there are two strategies.

- A.1 Without improving the algorithm itself.
- A.2 Improve the computational algorithm.

A.1. This category use the idea of stopping criteria (i.e., terminating the iterations of the optimization routine by incomplete optimization based on, e.g., a priori desired  $\alpha \in (0, 1]$  [179], which is related to total closed-loop performance. This strategy will be discussed with Algorithm 7 in Section 6.2.4. In this case, we do not need to find the global or even local minimum. We only aim for a feasible solution that provides a sufficient decrement in the cost function [180]. This situation is also commonly applied in embedded systems, which

requires a fast solution [175]. Another way to reduce the computation burden is merely forcing  $u^*(0)$  to be integer. That is, compared to the exhaustively search all the elements within the horizon  $u^*(k) \in \mathbb{Z}, k = \{0, 1, \dots, N - 1\}$ , we only consider the first elements  $u^*(0) \in \mathbb{Z}$  in the optimization routine and allow the other elements to be continuous,  $u^*(k) \in \mathbb{R}, k = \{1, \dots, N - 1\}$ . Obviously, in doing so, it would drastically reduce the computational load and an acceptable solution may be derived. The performance may not be significantly deteriorated. Yet, the optimality and feasibility may be lost. We would like to point out that since GA adopt a random search technique regardless of discrete or continuous variables, thus, this method is preferred to be used in B&B, cf., Table 7.4, which you can observe that the computational load from MPC-B&B is significantly decreased.

A.2 The orientation of this thesis is not immediately concerned with computational techniques but demonstrate the effectiveness of the proposed method. However, the following representative papers provide fruitful results to be learned for future research with large system dimensions or/and long prediction horizon in practice. Taken the online computational burden into account, an explicit MPC strategy was investigated for linear systems and also mixed logical dynamical systems [113, 181]. The core of this algorithm is that the explicit state-feedback law can be computed offline with the polyhedral partition of the feasible states, which allow for less online computation time to reach the optimal solution. However, the required storage space would dramatically increase with the system dimension and complexity of the partition. Considering the limitations from both of explicit MPC and online computation, [182] introduced a new approach that combines both of them for linear systems. In their work, a piecewise affine approximation of the optimal solution was computed offline to be used to warm-start an active set linear programming method. The results indicated that the warm-start outperforms either a pure offline or online method. A fast mixed nonlinear model predictive control algorithm was implemented by Outer Approximation instead of B&B in [121]. In [183], the authors proposed a multi-stage robust NMPC strategy, which could provide a fast and sustainable solution in real-time while considering uncertainty. Last, a computationally efficient MPC method was pro-

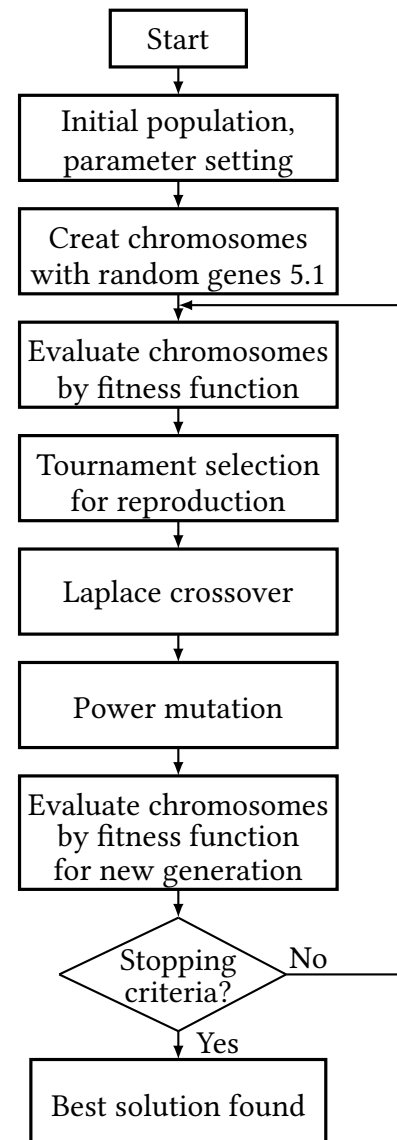


Figure 5.2.: Flow chart of MPC-GA

posed for linear systems with integer inputs or finite input sets. They formulated the optimization problem to be an integer least-squares problem and then solved with B&B in the form of a modified sphere decoding [184, 185, 186, 187].

## 5.2. RECONFIGURATION RULE WITHIN INTEGER PREDICTIVE CONTROL SETTING

In Section 4.3, we introduced a reconfiguration rule in a general setting (Algorithm 1), which is applicable to those control algorithms without prediction functionality, e.g., PID. In this case, the control actions  $u$  is executed after the determination of the uncertain reconfiguration delay  $r_d$ . However, this would become much more complicated when combining these reconfiguration cases with MPC. We have to deal with the relationship between prediction horizon with the step of reconfiguration delay, i.e, the variables of the entire control sequence in the optimizer needs to be updated based on the comparison of adjacent states within the horizon of reconfiguration delay. Furthermore, due to the limited available RMTs, we propose a priority strategy based on our primary goal—maintaining WIP for each workstation.

### 5.2.1. RECONFIGURATION RULE BASED ON PRIORITY OF WIP TRACKING

Respective details are given in Algorithm 3 below. Note that the reconfiguration delay is typically less than 2 hours. For simplicity without loss of generality, thus here we assume  $r_d = 1$  and use the following abbreviations in Algorithm 3:

- $u_j(n)$  is the first element of control sequence  $u_j(\cdot)$  of the MPC framework.
- $u_j(n - 1)$  is the previously applied control element, which will be updated relying on the comparison of adjacent control states in the scope of prediction horizon  $N$ . This update is necessary and significant concerning the real system dynamics and implementing the real value to the system. It may decrease the influence caused by the reconfiguration delay.
- If some idle RMTs exist, then we sort the required increases of RMTs  $req_{rmt}$  for all workstations and determine a priority.

Note that the line 27-30 of the Algorithm 3 concerning the triggered RMTs means that when there is a requirement for increasing the number of RMTs, there must exist a delay and therefore the current control value is not changed but the previous value is used until the reconfiguration is completed. Yet, this is too conservative. To explain it, we sketch an



**Algorithm 3** Reconfiguration rule within solving problem (3.25) in the MPC setting with integer operators

**Require:** Time instant  $n$ , planned input  $u_j(\cdot)$ , previously applied input  $u_j(n-1)$ , and the maximum value of RMTs  $m$ , prediction horizon  $N$ , and system dimension  $p$ .

```

1: if  $u_j(n-1) = \emptyset$  for some  $j \in \{1, \dots, p\}$  then
2:   for  $j = 1$  to  $p$  do
3:     Set  $u_j(n-1) = 0$ 
4:   end for
5: end if
6: for  $\iota = 0$  to  $N-1$  do
7:    $req_{rmt}(\iota) = 0$ 
8: end for
9: for  $\iota = 0$  to  $N-1$  do
10:  for  $j = 1$  to  $p$  do
11:    if  $u_j(n+\iota) > u_j(n-1)$  then
12:       $\Delta u_j(\iota) = u_j(n+\iota) - u_j(n-1)$ 
13:       $req_{rmt}(\iota) = req_{rmt}(\iota) + \Delta u_j(\iota)$ 
14:    end if
15:  end for
16:   $total_{previous} = \sum_{j=1}^p u_j(n-1)$ 
17:  for  $j = 1$  to  $p$  do
18:    if  $u_j(n+\iota) > u_j(n-1)$  then
19:      if  $total_{previous} == m$  then
20:         $u_j^{real}(\iota) = u_j(n-1)$ 
21:      else if  $m - req_{rmt}(\iota) < total_{previous} < m$  then
22:         $[req_{rmt}, index] = \text{sort}(\Delta u(:, \iota), 'descend')$ 
23:         $req_{rmt}_{rank} = 0$ 
24:        for  $j = 0$  to  $p-1$  do
25:           $req_{rmt}_{rank} = req_{rmt}_{rank} + \Delta u(index(j+1))$ 
26:          if  $req_{rmt}_{rank} \geq m - total_{previous}$  then
27:            for  $\gamma = 1$  to  $j$  do
28:               $u_{index(\gamma)}^{real_1}(\iota) = u_{index(\gamma)}(n-1)$ 
29:               $u_{index(\gamma)}^{real}(\iota) = u_{index(\gamma)}(n-1) + \Delta u_{index(\gamma)}$ 
30:               $u_{index(\gamma)}(n-1) = u_{index(\gamma)}^{real}(\iota); u_{index(\gamma)}^{real}(\iota) = u_{index(\gamma)}^{real_1}(\iota)$ 
31:            end for
32:            for  $\gamma = j+1$  to  $p$  do
33:               $u_{index(\gamma)}^{real}(\iota) = u_{index(\gamma)}(n-1)$ 
34:            end for
35:          end if
36:        end for
37:      else
38:         $u_j^{real_1}(\iota) = u_j(n-1), u_j^{real}(\iota) = u_j(n-1) + \Delta u_j(\iota)$ 
39:         $u_j(n-1) = u_j^{real}(\iota), u_j^{real}(\iota) = u_j^{real_1}(\iota)$ 
40:      end if
41:    else
42:       $u_j^{real}(\iota) = u_j(\iota)$ 
43:       $u_j(n-1) = u_j^{real}(\iota)$ 
44:    end if
45:  end for
46: end for

```

**Output:** Applied input  $u_j^{real}(\cdot)$  and updated  $u_j(n-1)$  for next iteration

example via Figure 5.3 where  $u(n - 2)$  stands for the previous previous value,  $u(n - 1)$  represents the previous value,  $u(n + 0)$  is the current value, the others are the rest of the optimal control sequence within prediction horizon  $N$ . In the case 2), we find  $u(n + 0) > u(n - 1)$ , it means that there exists an increase request with respect to usage of RMTs, but actually here no delay occurred due to  $u(n + 0) \leq u(n - 2)$ . Hence, the required increase in RMTs could be immediately compensated by the workstation itself. In contrast to that, in the case 1),  $u(n + 0) > u(n - 1) \& u(n + 0) > u(n - 2)$ , that means the reconfiguration is indeed triggered and the required RMTs have to wait until the reconfiguration is completed from other workstations.

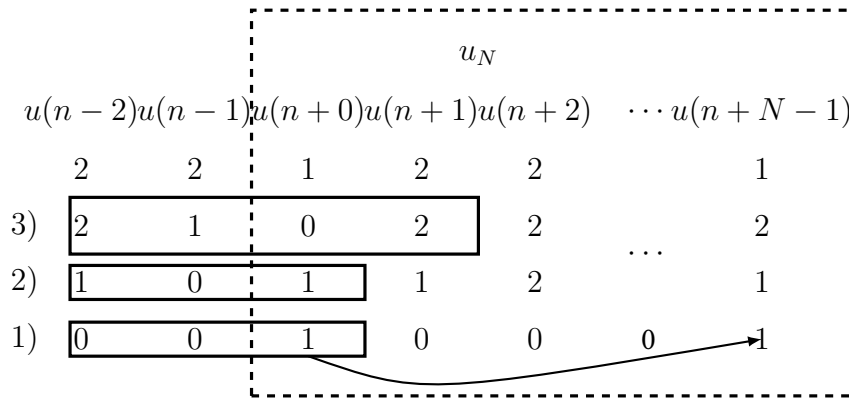


Figure 5.3.: Comparison of adjacent states for the triggered reconfiguration

**Remark 5.1**

Note that <sup>2</sup>  $u(n + 0), u(n - 1), u(n - 2)$  needs to be iteratively updated for an interactive comparison along with the shifted control step  $i$  within the prediction horizon  $N$ ,  $i = 0, 1, \dots, N - 1$ . Specially,  $u(n - 2)$  is actually the historical maximum value that among all control sequence within horizon  $N$  and also the already visited values. That is, for each iteration, with the shift of step  $i$  within prediction horizon  $N$ , the historic maximum previous value  $u(n - 2)$  at  $i_{th}$  step within  $N$  is given by  $\max\{u(l)\}$ ,  $n = 2, 3 \dots, i = 0, 1, \dots, N - 1$ . In order to better understand this situation, we illustrate case 3) in Figure 5.3 as an example with  $N = 2$ : Here, we observe that now the “current value  $u(n + 1)$ ”  $>$  “ $u(n + 0)$  - previous value” and also  $u(n + 1) > u(n - 1)$  - “previous previous value”, but  $u(n + 1) = u(n - 2)$  “historic previous value”. In this case, no reconfiguration occurs.

**5.2.2. COUNTING TIMES OF RECONFIGURATION FOR PRODUCTION COST**

Considering these complicated cases, we extend the Algorithm 3 to Algorithm 4, which may improve the closed-loop performance with a less conservative control input. Another

<sup>2</sup>Actually,  $u$  as a vector consists of all control inputs from  $p$  workstations. For simplicity, we use  $u$  instead of  $u_j, j = 1, \dots, p$  for adjacent state comparison of each workstation.

advantage is that it allows us to figure out how many times of a reconfiguration is actually triggered to achieve tasks. To this end, we modify Algorithm 3 which mainly focus on the cases with increasing requirement. That is, line 27-33 is the situation where that there exists some available RMTs but not enough to respond to all increase requirements. Based on priority rule, if those workstations acquiring RMTs from themselves, then there is no reconfiguration delay. <sup>3</sup>The same analysis goes to the line 38-39 where there exists enough RMTs available for the increasing requirements but we still should figure out the assignment direction. That is, increasing requirements within a workstation comes without delay, but assigned to other workstation with delay. Regarding the last case with increase requirement (i.e., line 19-20), there must exists a delay due to previously fully utilizing RMTs. This systematical analysis is important if the reconfiguration cost is considered as one of the performance indicators. We like to highlight that once an RMT is assigned to another workstation, then the historic maximum previous value for this workstation will be reset to the current value which as the new historic maximum previous value. For instance, in Figure 5.3, the arrow means that although “current value  $u(n+N-1) \leq u(n+0)$  – historic maximum previous value”, the reconfiguration delay indeed exists because this RMT was already assigned to another workstation. In this case, the historic maximum previous value is 0 instead of 1.

---

**Algorithm 4** Improvement of Algorithm 3 for controlling WIP along with counting times of reconfiguration

---

**Require:** The rolling updated historic maximum previous value  $u(n-2)$ .

```

1:  ⋮
2:  for  $\gamma = 1$  to  $j$  do
3:    if  $u_{index(\gamma)}^{real}(t) > u_{index(\gamma)}(n-2)$ , % There is a delay, has to wait until reconfiguration
        completed form other workstations, count reconfiguration times then
4:       $u_{index(\gamma)}^{real_1}(t) = u_{index(\gamma)}(n-1)$ ;
5:       $u_{index(\gamma)}(n-2) = u_{index(\gamma)}(n-1)$ , % Update historic maximum previous value
6:       $u_{index(\gamma)}^{real}(t) = u_{index(\gamma)}(n-1) + \Delta u_{index(\gamma)}$ 
7:       $u_{index(\gamma)}(n-1) = u_{index(\gamma)}^{real}(t)$ ; % Update previous value for next comparison
         $u_{index(\gamma)}^{real}(t) = u_{index(\gamma)}^{real_1}(t)$  % Actual applied value
8:    else
9:       $u_{index(\gamma)}^{real}(t) = u_{index(\gamma)}(n-1) + \Delta u_{index(\gamma)}$ 
10:      $u_{index(\gamma)}(n-1) = u_{index(\gamma)}^{real}(t)$ ;
11:    end if
12:  ⋮
13: end for

```

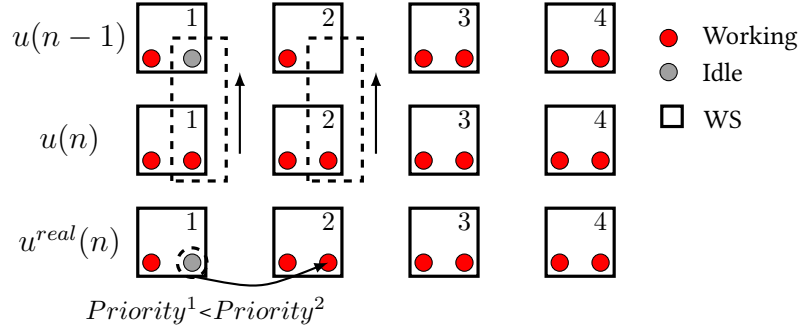
**Output:** Applied input  $u^{real}$

---

<sup>3</sup>The modification for line 38-39 is similar to the former, it is omitted in Algorithm 4.

### 5.2.3. POSSIBLE UNNECESSARY TRIGGERED RECONFIGURATION

Algorithms 3 and 4 are designed to control WIP as the primary goal. Yet, due to the limited available RMTs and priority, the temporary unused RMTs may be assigned to other workstations although the current workstation has an increased requirement. In this case, the reconfiguration cost will increase. For instance, in Figure 5.4, due to a priority rule, the unused one RMT in WS1 is assigned to WS2 although WS1 also needs one more RMT.



**Figure 5.4.:** Triggered reconfiguration based on WIP priority with possibly increased cost

Now, we summarize reconfiguration rules in kinds of scenarios.

- From view of customer
  - Reconfiguration is triggered based on maintaining WIP by a priority rule to reduce lead time and improve the reliability of delivery time. It may cause frequent but necessary reconfiguration which may lead to an increase of reconfiguration cost.
- From view of manufacturer
  - The determination of reconfiguration is based on considering production cost including inventory cost and reconfiguration cost. In this case, one simple way is to add  $\Delta u$  in the design of the stage cost with a penalty for the reconfiguration  $\lambda_1$ , i.e.,

$$\ell(x(k), u(k), \Delta u(k)) = \|x(k) - x^*\|_2^2 + \lambda \cdot \|u(k) - u^*\|_2^2 + \lambda_1 \cdot \|\Delta u(k)\|_2^2.$$

- From view of both of customer and manufacturer
  - Since this refers to multi-objective optimization which typically including conflicting objectives, we refer to multi-objective MPC [188] and multi-objective GA [189].

We would like to point out that under global competition, manufacturers in general prefer to meet the customer requirements at the expense of additional cost. Also, if the due date

is delayed due to a bad shop floor control performance, then manufacturers have to compensate consumers with a certain discount. This is why we focus on controlling WIP to the desired value in case of bottlenecks as a primary goal for the improvement of shop floor control performance, especially in the presence of frequent fluctuations in demand.

#### 5.2.4. PREDICTION HORIZON AND STEP OF RECONFIGURATION DELAY

The control interval  $T$  (sampling time) plays an important role in MPC which relates to closed-loop performance and computational cost. In this thesis, the control interval and the reconfiguration delay are assumed to be one hour. Based on such a setting, the reconfiguration rule is formulated while considering the relationship between prediction horizon with reconfiguration delay. We would like to point out that since in the reconfiguration stage the capacity of whole system along with those reconfigured machines is decreasing, it is crucial to consider the chain reaction caused by such a delay and make appropriate decisions with respect to the triggered reconfiguration. Intuitively, a longer prediction horizon would be preferable. Yet, as we mentioned before, this would dramatically aggravate the computational burden for an NP-hard optimization problem. In some cases, improving the prediction horizon may not lead to an improved closed-loop performance [190, 191]. Thus, the “minimal” prediction horizon should be proportional to the reconfiguration delay such that

$$N_{min} = r_d/T + 1 \quad (5.1)$$

holds. But we like to point out that both parameters in (5.1) impact the closed-loop performance. Given reconfiguration delay  $r_d$ , the shorter the control interval  $T$  is, the longer  $N$  needs to be chosen to include  $r_d$ . Given  $r_d$  and  $T$ , this constructed  $N_{min}$  is obtained but also needs to be satisfied with stability conditions (3.42) requiring a sufficient large  $N$ . To our best of knowledge, there are no related techniques available that allow us to analytically analyze the stability incorporating all these three parameters. Nevertheless, we can compute the degree of suboptimality online for implicitly showing stability. In particular, extending the one step relaxed Lyapunov inequality (3.34) to  $k$  steps, we may keep the stabilizing minimal  $N$  unchanged with the concept of a flexible Lyapunov function that allow local increases but with a decrease in future steps. This part will be discussed in Chapter 6.

### 5.3. SUMMARY

In order to effectively control WIP via RMTs under unpredictable events, we employ MPC to achieve a sustainable capacity allocation due to the capability of explicitly handling constraints. In addition, we formulate a set of reconfiguration rules to trigger determination of RMTs in the framework of MPC, which allows us to reduce the negative effects caused by

reconfiguration delay through the usage of predictive future information, and help manufacturers to make appropriate decisions in advance to avoid performance deterioration. To address the integer assignment of RMTs on the shop floor, we propose three control strategies – MPC-floor, MPC-B&B and MPC-GA. The first solution is easy to implement but may destabilize the system due to the truncated control effort. MPC-B&B is able to guarantee optimality of the derived solution but at the expense of computational cost. MPC-GA may provide a non-optimal but acceptable solution in a reasonable time. With the increase of system dimension and prediction horizon, the computational load would exponentially increase and be predominant, rendering MPC-GA to be the preferable option. In next chapter, we will show conditions such that the controlled system is stable despite possibly frequent reconfigurations.

# 6

## STABILITY OF A JOB SHOP SYSTEM WITH RMTs FOR WIP CONTROL

A digitalized manufacturing process is a core assumption in the context of Industry 4.0 [20]. Stability, accuracy and rapidity are the three requirements for a control system design. Among them, stability is the prerequisite and is of utmost importance to steer a production process as stable as possible in a long term consideration. In [192], the authors studied the classical automatic pipeline, inventory and order-based production control system in the form of a discrete time state space model. The stability analysis was conducted and results indicated that the system may be unstable without a closed-loop feedback of the WIP. In [193], the robustness and stability were considered for a dynamic job shop scheduling problem subject to machine breakdowns by means of a hybrid GA. In [194], an intelligent control system was designed for tuning the process to mitigate chattering.

Typically, stability is discussed in terms of the Lyapunov stability or strictly asymptotic stability. However, these concepts are unrealistic in a variety of real time applications, e.g., temperature control, inventory control, power electronics control, etc., where strict asymptotic stability may be lost. Practical asymptotic stability extends the concept of asymptotic stability and has a significance for control system with quantization errors [127], sampling control system [130] as well as investigations of interconnections of nonlinear system by small-gain theorems [195].

In this thesis, the process stability we considered refers to performance indicators (e.g., WIP) which converge to desired values or acceptable stability regions. Once a trajectory enters a neighborhood of the equilibrium, it will remain there. In [19], a comparison regarding stability regions was conducted from both perspectives (macroscopic and microscopic), i.e., continuous modeling by mathematical theory and simulation results from DES. The authors indicated that such an approximation made by a mathematical model is suitable and effective for stability analysis. According to the specification of the arrival rate, this method allows to determine stability parameters for a production network with less time consumption compared to a repeated trial and error approach. However, only state stability information was discussed, the tracking control problem was not taken into account. In [196], the authors adopted the classical PID control into a disturbed and time-delayed job

shop system with RMTs for capacity control. The tracking performance was achieved and stability was guaranteed and analyzed by Nyquist criterion. However, the constraints were not truly treated but a fractional approach was considered, which may lead to non-optimal solutions and performance deterioration.

In this thesis, we employ MPC to maintain a predefined WIP level by usage of RMTs in the presence of disturbance along with the possible reconfigurations and to analyze the closed-loop system stability. To this end, we qualitatively and quantitatively analyze and guarantee stability via two types of stabilizing controllers – using MPC with and without terminal conditions. More specifically, we first qualitatively analyze the asymptotic stability property in case of continuous variables without considering integer constraints and reconfiguration delays with and without endpoint constraints, respectively. Further, we quantitatively analyze practical stability using terminal cost and terminal region and without such terminal conditions. Last, considering that the existence of reconfiguration delay may lead to local instability, we propose to combine adaptive MPC method with the principle of flexible Lyapunov functions that allow the optimal value function to increase locally but with an ensured decrease in future steps.

Parts of this chapter have been published in [32, 162].

## **6.1. ASYMPTOTIC STABILITY WITHOUT INTEGER CONSTRAINTS AND DELAYS**

We compared three stability methods in terms of design, controller performance, feasibility and numerical effort in Section 3.5. In terms of applicability in practice, the terminal endpoint constraints and unconstrained MPC scheme are easy understandable and preferable for practitioners because only a cost function and a desired equilibrium need to be designed [47, 147, 197]. For example, in [198], the authors firstly used the zero terminal state constraint to guarantee the stability of a batch process which is formulated as a two dimensional system. In [199], the authors studied a supply chain network by means of sequential distributed MPC. The terminal equality constraints imposed in the local MPC inherently recursively satisfied the overall system. In [200], the authors used MPC with endpoint constraints to ensure the validity of the derived solutions in the operation region under the variation of initial measurable conditions of an industrial batch process. In [201], an adaptive MPC with zero terminal state constraint was designed for a production-inventory system to be capable of identifying the process dynamics in real time. This method was easy to implement with the routinely collected data from manufacturing companies and was a very powerful tool for managers to make quick responses to demand variations. In [202], the authors applied MPC with endpoint constraint in a industrial machine tool servo drive system for reference tracking. The tracking performance and stability were guaranteed and



the results outperformed cascaded PID control in terms of tracking accuracy.

The unconstrained MPC, which follows the original attribute of the control objective without any additional imposing terminal conditions, is commonly applied in industry. In [203], regulation of nonholonomic mobile robots to the desired position and orientation was performed with MPC. The asymptotic stability was guaranteed by a tailored non-quadratic stage cost and a sufficient prediction horizon. In [130], a arm/rotor/platform (ARP) model was conducted by MPC without using stabilizing constrain or cost. The tracking performance was ensured and the stability was implicitly guaranteed by a characteristic value  $\alpha$  along with the visited closed-loop trajectory.

Hence, due to its simplicity, we first adopt the simple terminal endpoint constraint method in case of without integer constraints and reconfiguration delays. Our goals include the following points: first, in absence of integer constraints and reconfiguration delays, the stability will be analyzed qualitatively. Thereafter, we aim to show effectiveness of the proposed method in the basic case, which serves as a cornerstone and will be extended for analyzing the system performance in the presence of integer constraints and reconfiguration delays. Third, we additionally compare its performance with the unconstrained MPC scheme, especially as the latter paves the way for analytically approximately estimating the practical stability region in Section 6.2.5.

### 6.1.1. TERMINAL ENDPOINT CONSTRAINTS

#### Definition 6.1

Suppose a system (4.3), a predefined reference value  $x^*$  and a control  $u(\cdot)$  to be given such there exists a forward invariant set  $Y \subset \mathbb{X}$ . If there exists a function  $\beta \in \mathcal{KL}$  such that

$$\|x(n) - x^*\| \leq \beta(\|x_0 - x^*\|, n) \quad (6.1)$$

holds for all  $x_0 \in Y$  and all  $n \in \mathbb{N}_0$ , then the control  $u(\cdot)$  asymptotically stabilizes  $x^*$ .

To achieve the goal of asymptotic stability, we propose to utilize MPC introduced in Chapter 5. Here, we impose and employ the standard assumptions presented in Assumption 3.23 and 3.24. Our goal is to steer the WIP level of each workstation to the desired value  $x^*$  while considering the state and control constraints

$$x(N) = \{x^*\}, \quad 0 \leq u_j(n) \quad \text{and} \quad \sum_{j=1}^p u_j(n) \leq m \quad (6.2)$$

To this end, we imposed the stage cost function (6.4), which satisfies Assumption 3.23 with

$$u^* = \frac{-P^{-1} \cdot d - n^{DMT} \cdot r^{DMT}}{r^{RMT}}. \quad (6.3)$$

For technical reasons, we require

**Assumption 6.2**

The flow probability matrix  $P$  between the workstations is invertible.

Then we can utilize our dynamics (4.3) together with Assumption 6.2 to show the following:

**Proposition 6.3**

Consider Problem 3.25 for the job shop system (4.3) together with the stage costs

$$\ell(x(k), u(k)) = \|x(k) - x^*\|_2^2 + \lambda \cdot \|u(k) - u^*\|_2^2 \quad (6.4)$$

and  $\lambda > 0$  for some predefined desired equilibrium  $(x^*, u^*)$ . Then Theorem 3.25 holds for  $N = 2$  with

$$\begin{aligned} \alpha_1(s) &= \alpha_3(s) = s^2 \\ \alpha_2(s) &= (1 + \lambda\|\theta_1\|_2^2 + \|\theta_2\|_2^2 + \lambda\|\theta_3\|_2^2)s^2 \end{aligned}$$

where

$$\begin{aligned} \theta_1 &= (4\lambda \cdot \mathbf{Id} + 2(r^{RMT})^2 P^\top P)^{-1} (-2r^{RMT} P^\top + \frac{-P^{-1}}{r^{RMT}} \cdot 2\lambda) \\ \theta_2 &= \mathbf{Id} + r^{RMT} P \theta_1 \\ \theta_3 &= \frac{P^{-1} + r^{RMT} \theta_1}{r^{RMT}}. \end{aligned}$$

Moreover, the stabilizing MPC feedback is given by

$$\kappa_2(x) = \theta_1(x - x^*) + u^*.$$

The weighting coefficient  $\lambda$  in stage cost (6.4) represents the penalty for control effort. Obviously, the smaller  $\lambda$  is, the more quick convergence is reached. In this thesis, the weighting coefficients concerning the WIP tracking error for each workstation are assumed to be equal. If one specific workstation has a high priority and expect to be steered to the desired value quickly, it can be achieved by enhancing the associated coefficient.

**Proof of Proposition 6.3:** [Terminal endpoint constraint]

Given the running cost (6.4),  $\alpha_1(s) = \alpha_3(s) = s^2$  satisfy asymptotic Lyapunov function conditions (3.18) and (3.19). Hence, only the bound  $\alpha_2(\|x - x^*\|) \geq V_N(x)$  needs to be established. Based on Lemma 6.4, we conclude that if  $V_2(x) \leq \alpha_2(\|x - x^*\|)$  holds, then  $V_N(x) \leq \alpha_2(\|x - x^*\|)$  holds for any  $N \geq 2$ . Through  $f(x, \kappa_1(x)) = x^*$ , we obtain

$$\kappa_1(x) = \frac{P^{-1}(x^* - x) - n^{DMT} \cdot r^{DMT} - P^{-1}d}{r^{RMT}},$$

then with (6.3), we have

$$V_1(x) = \|x - x^*\|_2^2 + \lambda \cdot \left\| -\frac{P^{-1}(x - x^*)}{r^{RMT}} \right\|_2^2. \quad (6.5)$$

Based on the dynamic programming principle, we get  $V_2(x) = \ell(x, \kappa_2(x)) + V_1(f(x, \kappa_2(x)))$ .

Now, combine with system dynamic (4.3) and (6.5), we get

$$\begin{aligned} \kappa_2(x) &= \operatorname{argmin}_{u \in \mathbb{U}} (\ell(x, u) + V_1(f(x, u))) \\ &= \operatorname{argmin}_{u \in \mathbb{U}} \|x - x^*\|_2^2 + \lambda \cdot \|u - u^*\|_2^2 \\ &\quad + \|(x - x^* + P \cdot n^{DMT} r^{DMT}) + d + r^{RMT} \cdot P \cdot u\|_2^2 \\ &\quad + \lambda \cdot \left\| -\frac{P^{-1}(x - x^* + P \cdot n^{DMT} r^{DMT} + d)}{r^{RMT}} - u \right\|_2^2. \end{aligned}$$

Next, we set

$$\begin{aligned} a_1 &:= r^{RMT} \cdot P \\ a_2 &:= x - x^* + P \cdot n^{DMT} r^{DMT} + d \\ a_3 &:= -\frac{P^{-1}(x - x^* + P \cdot n^{DMT} r^{DMT} + d)}{r^{RMT}} \end{aligned} \quad (6.6)$$

and obtain

$$\begin{aligned} \kappa_2(x) &= \operatorname{argmin}_{u \in \mathbb{U}} [\|x - x^*\|_2^2 + \lambda \cdot \|u - u^*\|_2^2 \\ &\quad + \|a_2 + a_1 \cdot u\|_2^2 + \lambda \cdot \|a_3 - u\|_2^2]. \end{aligned} \quad (6.7)$$

Hence, we have

$$\begin{aligned} \frac{\partial \kappa_2(x)}{\partial u} &= 2\lambda(u - u^*) + 2a_1^\top (a_2 + a_1 \cdot u) - 2\lambda(a_3 - u) \\ &= (4\lambda \cdot \mathbf{Id} + 2a_1^\top a_1)u + 2a_1^\top a_2 - 2\lambda(a_3 + u^*) = 0 \end{aligned}$$

Since  $a_3 = -\frac{P^{-1}(x-x^*)}{r^{RMT}} + u^*$ , then

$$\begin{aligned}
u &= (4\lambda \cdot \mathbf{Id} + 2a_1^\top a_1)^{-1} (-2a_1^\top a_2 + 2\lambda(a_3 + u^*)) \\
&= (4\lambda \cdot \mathbf{Id} + 2a_1^\top a_1)^{-1} (-2a_1^\top (x - x^* + P \cdot n^{DMT} r^{DMT} + d) + 2\lambda(-\frac{P^{-1}(x-x^*)}{r^{RMT}} + 2u^*)) \\
&= \overbrace{(4\lambda \cdot \mathbf{Id} + 2a_1^\top a_1)^{-1} (-2a_1^\top + \frac{-P^{-1}}{r^{RMT}} \cdot 2\lambda)}^{\theta_1} (x - x^*) + \\
&\quad (4\lambda \cdot \mathbf{Id} + 2a_1^\top a_1)^{-1} (4\lambda u^* - 2a_1^\top \overbrace{(P \cdot n^{DMT} r^{DMT} + d)}^{-a_1 u^*}) \\
&= \theta_1(x - x^*) + u^*
\end{aligned} \tag{6.8}$$

As the problem is convex,  $u$  is the unique optimal solution according to the MPC scheme, which in turn allows us to set  $\kappa_2 = u$ . Utilizing the closed loop system dynamics via (4.3) and  $\kappa_2$ , we have

$$f(x, \kappa_2(x)) = x + P \cdot n^{DMT} r^{DMT} + d + r^{RMT} P(\theta_1(x - x^*) + u^*). \tag{6.9}$$

Then we substitute (6.9) into (6.5) and utilize (6.3) to obtain

$$\begin{aligned}
V_1(f(x, \kappa_2(x))) &= \|(\mathbf{Id} + r^{RMT} P \theta_1)(x - x^*) + \overbrace{(P n^{DMT} r^{DMT} + d + r^{RMT} P u^*)}^{=0}\|_2^2 + \\
\lambda \cdot &\| \frac{(P^{-1} + r^{RMT} \theta_1)(x - x^*)}{r^{RMT}} + \overbrace{\frac{n^{DMT} r^{DMT} + P^{-1} d + r^{RMT} u^*}{r^{RMT}}}_{=0} \|_2^2.
\end{aligned} \tag{6.10}$$

Last, we substitute (6.3) into (6.10), which then reads

$$\begin{aligned}
V_2(x) &= \ell(x, \kappa_2(x)) + V_1(f(x, \kappa_2(x))) \\
&= \|x - x^*\|_2^2 + \lambda \|\theta_1(x - x^*)\|_2^2 + \|\overbrace{(\mathbf{Id} + r^{RMT} P \theta_1)}^{\theta_2}(x - x^*)\|_2^2 \\
&\quad + \lambda \cdot \|\overbrace{\frac{(P^{-1} + r^{RMT} \theta_1)}{r^{RMT}}(x - x^*)}_{\theta_3}\|_2^2 \\
&\leq \|x - x^*\|_2^2 + \lambda \|\theta_1\|_2^2 \|x - x^*\|_2^2 + \|\theta_2\|_2^2 \|x - x^*\|_2^2 + \lambda \|\theta_3\|_2^2 \|x - x^*\|_2^2 \\
&= (1 + \lambda \|\theta_1\|_2^2 + \|\theta_2\|_2^2 + \lambda \|\theta_3\|_2^2) \|x - x^*\|_2^2
\end{aligned} \tag{6.11}$$

Therefore, we may define the bound

$$\alpha_2(s) = (1 + \lambda \|\theta_1\|_2^2 + \|\theta_2\|_2^2 + \lambda \|\theta_3\|_2^2) s^2. \tag{6.12}$$

Hence,  $V_2(x)$  is a Lyapunov function and the assumptions of Theorem 3.25 hold.

**Lemma 6.4**

Consider the optimal control problem (3.25) with prediction horizon  $N \in \mathbb{N}$  and the additional terminal condition  $x(N) = x^*$  and suppose that Assumptions 3.23 and 3.24 hold. Then for each  $N \geq 2$  and each  $x \in \mathbb{X}_{N-1}$  we have  $V_{N-1}(x) \geq V_N(x)$ .

Theorem 3.25 provides the general background for our task of asymptotically stabilizing a job shop system with RMTs. Hence, our aim now is to prove that the terminal endpoint constraints method applies to our case. The particular difficulty with MPC is that a suitable prediction horizon  $N$  is typically unknown, yet if  $N$  reveals a stabilizing control, then also the feedback with  $N + j$  for  $j \in \mathbb{N}$  stabilizes the closed-loop [Lemma 6.4, [148]]. Here, we show that for  $N = 2$ , the solution of problem (3.25) can be computed explicitly without the requirement of an optimization routine. Therefore, also all feedbacks with  $N \geq 2$  asymptotically stabilize the closed-loop, which renders the method to be applicable in general.

Due to the additional terminal endpoint constraints, recursive feasibility is guaranteed automatically, i.e., if the initial state and initial control sequence within prediction horizon of the job shop system adhere all constraints, then there always exists a solution to problem (3.25) and the MPC procedure can be applied without running into a dead end [107].

**Remark 6.5** (Stage cost and closed-loop stability)

*In practice, a typical choice for stage cost is the quadratic form which penalizes the distance to some desired equilibrium, e.g., (6.4). Note that the designed stage cost for stabilization may not necessarily reflect perfect performance [156], e.g. low energy consumption. Thus, there is a trade off between closed-loop stability and good performance. For instance, in our case, if we use terminal endpoint constraints method and the designed stage cost (6.4) without  $u^*$ , then the stabilization at equilibrium point  $(x^*, u^*)$  will be impossible because of that the system is affine, also see [Example 6.31 [107]] in which reducing the overshoot constant by a “good” stage cost can reduce the required  $N$  for stability. Note that the stage cost designed here with  $u^*$  is to achieve asymptotic stability via terminal equilibrium constraints. Later as we take integer constraints into account, asymptotic stability cannot be achieved using this method, so the stage cost needs be designed without  $u^*$ . We like to note that  $\ell$  may represent any performance indicator or a scalarized combination of several indicators. For multi-objective extensions, we refer to [188].*

**6.1.2. UNCONSTRAINED MPC SCHEME**

In this section, we additionally employ unconstrained MPC for comparison with MPC with terminal endpoint constraints. Also, this will be as a cornerstone for further steps to analyze practical stability in the presence of integer constraints and reconfiguration delays. Now,

recall the system dynamics again

$$x(n+1) = x(n) + r^{RMT} \cdot P \cdot u(n) + P \cdot n^{DMT} r^{DMT} + d$$

with the same stage cost defined in (6.4). Since the computation burden grows rapidly along with the increase of prediction horizon, it is not desirable to apply larger ones than necessary to guarantee the relaxed Lyapunov inequality (3.34) hold. Therefore, in the case of unconstrained MPC, we still first consider the shortest possible prediction horizon  $N = 2$ . Assume  $u^* \in \mathbb{U}$ ,  $\ell^*(x) = \inf_{u \in \mathbb{U}} \ell(x, u) = \|x - x^*\|_2^2$  satisfies (3.44). Also, we obtain the feedback law via without terminal conditions

$$\begin{aligned} \kappa_2(x) &= \underset{u \in \mathbb{U}}{\operatorname{argmin}} (\ell(x, u) + V_1(f(x, u))) \\ &= \underset{u \in \mathbb{U}}{\operatorname{argmin}} (\|x - x^*\|_2^2 + \lambda \cdot \|u - u^*\|_2^2 + \|a_2 + a_1 \cdot u\|_2^2). \end{aligned}$$

Hence, we have

$$\frac{\partial \kappa_2(x)}{\partial u} = 2\lambda(u - u^*) + 2a_1^\top (a_2 + a_1 \cdot u).$$

Then we obtain

$$\begin{aligned} u &= (2\lambda \cdot \mathbf{Id} + 2a_1^\top a_1)^{-1} \left( \overbrace{\frac{\partial \kappa_2(x)}{\partial u}}^0 + 2\lambda u^* - 2a_1^\top a_2 \right) \\ &= (2\lambda \cdot \mathbf{Id} + 2a_1^\top a_1)^{-1} (2\lambda u^* - 2a_1^\top (x - x^* - Pr^{RMT} u^*)) \\ &= (2\lambda \cdot \mathbf{Id} + 2a_1^\top a_1)^{-1} \left( (2\lambda \cdot \mathbf{Id} + 2a_1^\top \overbrace{Pr^{RMT}}^{a_1}) u^* - 2a_1^\top (x - x^*) \right) \\ &= \overbrace{-(2\lambda \cdot \mathbf{Id} + 2a_1^\top a_1)^{-1} \cdot 2a_1^\top}^{\Phi_1} (x - x^*) + u^* \end{aligned} \quad (6.13)$$

Similar to terminal endpoint constraints,  $u(n) = \Phi_1(x(n) - x^*) + u^* \in \mathbb{U}$  may not hold with constraints, (e.g., at initial high WIP). To this end, we inductively define a feedback  $\Phi_2 := -q \cdot a_1^{-1} \cdot \mathbf{Id}$ ,  $q \in (0, 1)$  such that  $u(n) = \Phi_2(x(n) - x^*) + u^* \in \mathbb{U}$  hold. Then we set  $\|\mathbf{Id} + a_1 \Phi_1\|_2 := \infty$  if  $u(\Phi_1(x)) \notin \mathbb{U}$  and define

$$\Phi := \underset{\Phi_1, \Phi_2}{\operatorname{argmin}} \{ \|\mathbf{Id} + a_1 \Phi_1\|_2, \|\mathbf{Id} + a_1 \Phi_2\|_2 \}. \quad (6.14)$$

The closed-loop trajectory in the case of unconstrained MPC is then given by

$$x(n) = ((\mathbf{Id} + Pr^{RMT} \Phi))^n (x_0 - x^*) + x^*. \quad (6.15)$$

Consequently, we use the feedback law (6.13) and (6.15)

$$\begin{aligned}
\ell(x(k), u(k)) &= \|x(k) - x^*\|_2^2 + \lambda \|u(k) - u^*\|_2^2 \\
&= \|x(k) - x^*\|_2^2 + \lambda \|\Phi(x(k) - x^*) + u^* - u^*\|_2^2 \\
&\leq (1 + \lambda \|\Phi\|_2^2) \|x(k) - x^*\|_2^2 \\
&\leq \overbrace{(1 + \lambda \|\Phi\|_2^2)}^C \overbrace{(\|\mathbf{Id} + r^{RMT} \cdot P \cdot \Phi\|_2^2)}^\sigma)^k \overbrace{\|x(0) - x^*\|_2^2}^{\ell^*(x_0)} \quad (6.16)
\end{aligned}$$

If overshoot  $C$  is sufficiently close to 1, then no matter what the decay rate  $\sigma \in (0, 1)$  is, we can always ensure stability with the shortest possible prediction horizon  $N = 2$  [107]. Also, based on the controllability condition regarding the stage cost (6.16), we do not need to compute and check Assumption 3.31. Instead, utilizing the result presented in the Lemma 3.37, we can compute  $\gamma_N$  from (3.46) and then obtain the associated performance index  $\alpha_N$  depending on  $N$  via (3.47) [158]. Here, we choose  $N = 2$ , then we have  $\gamma_2 = C(1 + \sigma)$  and  $\alpha_2 = 1 - (\gamma_2 - 1)^2$ . To verify  $0 < \alpha_2 \leq 1$ , it is equivalent to ensure  $1 \leq \gamma_2 < 2$ . Now we set  $\Phi_2 := \Phi$  and put it into (6.17). Given the system parameters  $a_1$  which consists of flow matrix  $P$  and production rate of RMT  $r^{RMT}$ , through choosing  $\lambda \in (0, 1)$  and  $q \in (0, 1)$ , we can show

$$\begin{aligned}
\gamma_2 = C(1 + \sigma) &= (1 + \lambda \|\Phi_2\|_2^2)(1 + \|\mathbf{Id} + a_1 \cdot \Phi_2\|_2^2) \\
&= (1 + \lambda \|\mathbf{-}q \cdot a_1^{-1} \cdot \mathbf{Id}\|_2^2)(1 + \|\mathbf{Id} + a_1 \cdot \mathbf{-}q \cdot a_1^{-1} \cdot \mathbf{Id}\|_2^2) \quad (6.17)
\end{aligned}$$

and therefore

$$1 \leq (1 + \lambda \cdot q \|a_1^{-1}\|_2^2)(1 + (1 - q)^2) < 2 \quad (6.18)$$

to hold.

The current case study represents a straightforward proof of concept, which will be extended to incorporate reconfiguration delays and integer programming methods for addressing the assignment of RMTs. Note that in practice the perfect tracking (Definition 6.1) is almost surely impossible, but needs to be extended to practical stability, cf. [107, Chapter 2]. Apart from operators who may interfere within the processes, the latter is due to two facts: For one, any reconfiguration cannot immediately completed but with a delay time (e.g., installation time). And secondly, the assignment of RMTs for each workstation is inherently to be integer, whereas typical feedbacks consider convex sets. To cope with both problems in a long term, we propose to employ MPC as a control scheme, which is able to explicitly handle constraints through solving an optimization problem. To make the first step into this direction and show effectiveness of the proposed method, in this section, we consider the basic system dynamics in case of continuous variables and compare MPC

to the standard PID implementation, the comparison results of which will be presented in Section 7.1.

In our case, the cost function we used is of quadratic form and we show that stability can be guaranteed even with the shortest prediction horizon by unconstrained MPC. This is very significant for analyzing practical stability in the context of integer constraints and reconfiguration delays. Particularly, the latter may lead to a locally increasing Lyapunov function even before reaching the practical stability region, cf., Remark 3.18. Also, we like to point out that in some cases, the quadratic cost may fail in MPC since a sufficient prediction horizon for stability does not exist even if the system is finite time controllable, cf., [203, 204].

## 6.2. PRACTICAL STABILITY WITH INTEGER CONSTRAINTS AND RECONFIGURATION DELAYS

Since strict asymptotic stability cannot be achieved with integer constraints in general, the best we can hope for is that the closed-loop trajectory will converge to a neighborhood of the equilibrium, which is termed as the acceptable or practical stability region [18, 130]. This concept is similar to robust control invariant sets, in which the system state is confined in the presence of bounded uncertainties or disturbances [125, 205]. There are two categories of techniques for addressing the robust stability issue. The first class is the so-called robust MPC method, which mainly includes min-max approaches [206, 207], tube-based methods [208, 209] and multi-stage strategies [183]. In the min-max approach MPC, the worst case costs for all possible disturbances are taken into account. For tube-based method, the initial state is also considered as a decision variable together with the control sequence to ensure satisfaction of the constraints subject to a set of bounded disturbances. In the multi-stage MPC, the uncertainty is explicitly taken into account via using a scenario tree, which leads to a non-conservative robust control. The other category is called stochastic MPC, in which chance constraints are included to enable the systematic trade-off between achieving control objectives and an admissible level of closed-loop constraint violation in a probabilistic sense [210, 211].

In this thesis, since the assignment of RMTs is limited to be integer, the controlled WIP trajectory cannot perfectly track a reference but instead may show oscillations around such an equilibrium. Compared to continuous control, the discretization of inputs can be viewed as a bounded “disturbance”, which is inherently connected to the system model. In this way, we utilize the inherent robustness of MPC relying on its internal receding optimization strategy without any modification to the nominal controller instead of the aforementioned robust MPC approach, unless the effect caused by external disturbance is more significant than the “disturbance” caused by integer constraints [212]. Thus, practical stability has



the advantage of executing an MPC scheme which is robust against a certain extend of perturbations [213]. The inherent robustness had been verified in the case that the optimal value function is continuous which may fail in the presence of state constraints [214, 215].

To the best of our knowledge, regarding the practical stabilization property of MPC, there is a lack of an explicit characterization of the stability region, which is typically handled by extensive closed-loop simulations. Using terminal endpoint constraints may be intractable and impracticable with integer constraints to allow  $x^* = f(x^*, u^*)$  to hold [213, 216]. As a remedy, there are two solutions available. One is to replace the terminal endpoint constraint with a terminal region so that the feasibility of the optimization problem is guaranteed, i.e., the controlled WIP trajectory would be confined within an acceptable region rather than an exact target point, which is termed as practical stability region and will be compassed in the designed terminal region. A good interpreted example is the application of finite control inputs MPC in power electronics in which the candidate control variables are fixed to be from a finite set of real numbers [127, 212, 217]. Asymptotic practical stability is guaranteed by using terminal cost and the associated terminal region with  $N = 1$ . Yet, as the worst case derived from the bounded quantization error of control input sets is considered, the estimate of practical stability region is conservative.

The other solution is still in the framework of unconstrained MPC scheme. More specifically, we follow the method proposed in [130] and adopt the trajectory dependent approach to quantitatively estimate the degree of suboptimality  $\alpha$  online for stability. Although this method cannot rigorously guarantee stability, it is still a powerful tool to analyze closed-loop stability and performance of a feedback controller for a specific closed-loop trajectory. That is, we only consider the points  $x(n) \in \mathbb{X}$  visited by the closed-loop trajectory rather than all  $x \in \mathbb{X}$ . Using this method provides a chance to adaptively change the prediction horizon  $N$  based on a given stability and suboptimality estimate. In particular, since the reconfiguration delays may render the Lyapunov function to increase locally before the practical stability region is reached, we can combine adaptive horizon MPC with the principle of a flexible Lyapunov function.

### 6.2.1. TERMINAL COST AND TERMINAL REGION

Before we introduce the terminal cost and terminal region method, we would like to present some results with respect the discrete LQR optimal control problem, because the construction of the terminal cost (additional Lyapunov function) is based on the solution of discrete Algebraic Riccati Equations (DARE). Although there are a lot of references that used terminal cost and terminal regions within the stability analysis is, the systems they studied are almost all linear, without additive terms, and with continuous variables [127, 141, 147]. Yet, in our case, the system considered is with additive terms, and required the control variables

are integer. Now, we consider a class of discrete affine linear time-invariant systems :

$$x^+ = f(x, u) = Ax + Bu + c \quad (6.19)$$

where  $x \in \mathbb{R}^n$ ,  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$ ,  $u \in \mathbb{R}^m$ ,  $c \in \mathbb{R}^n$ ,  $n \in \mathbb{N}_0$ , and  $x^+$  is the successor state. Set  $\tilde{x} := x - x^*$ ,  $\tilde{u} := u - u^*$  where a pair  $(x^*, u^*)$  is an equilibrium point. We assume the following:

**Assumption 6.6**

There exists  $u^* \in \mathbb{R}^m$  such that  $x^* = f(x^*, u^*)$  hold, i.e.,  $c + (A - \mathbf{Id})x^* + Bu^* = 0$ .

Then through (6.19), we obtain:

$$\begin{aligned} x^+ - x^* &= A(x - x^*) + B(u - u^*) + c + Ax^* - x^* + Bu^* \\ \tilde{x}^+ &= A\tilde{x} + B\tilde{u} + c + (A - \mathbf{Id})x^* + Bu^* \\ \tilde{x}^+ &= A\tilde{x} + B\tilde{u} \end{aligned} \quad (6.20)$$

Consider the discrete infinite horizon optimal control problem (6.21)

$$J_\infty(\tilde{x}(k), \tilde{u}(k)) := \sum_{k=0}^{\infty} \ell(\tilde{x}(k), \tilde{u}(k)) := \sum_{k=0}^{\infty} \tilde{x}(k)^\top Q \tilde{x}(k) + \tilde{u}(k)^\top R \tilde{u}(k) \quad (6.21)$$

with the system dynamics (6.20), where  $Q$  is semi-positive definite matrix  $Q \geq 0$ ,  $R$  is positive definite matrix and  $R > 0$ . Assume the pair  $(A, B)$  is stabilizable. Then it can be solved via the discrete Riccati equation

$$P = A^\top P A - A^\top P B K + Q. \quad (6.22)$$

The static feedback control law  $\tilde{u} = -K\tilde{x}$  is obtained with the derived  $K = (B^\top P B + R)^{-1} B^\top P A$ . Since  $u$  is restricted to be integer, we set the terminal control law  $u_F = \mathcal{L}(u) = \overbrace{\tilde{u} + u^*}^u - e$ ,  $\mathcal{L} : \mathbb{R}^n \rightarrow \mathbb{Z}^n$ , where  $e$  is regarded as a discretization error caused by integer constraints and reconfiguration delays. Then the system dynamic (6.20) is expressed as:

$$\tilde{x}^+ = A\tilde{x} + Bu_F = A\tilde{x} + B(\tilde{u} + u^* - e) = \overbrace{(A - BK)}^{A_k} \tilde{x} - B(e - u^*) \quad (6.23)$$

In addition, the final terminal cost is defined as  $F(\tilde{x}) = \|\tilde{x}\|_P^2$ , where  $P$  is positive definite.

Then considering Assumption 3.26, we have

$$\begin{aligned}
 & F(\tilde{x}^+) - F(\tilde{x}) + \ell(\tilde{x}, u_F) \\
 &= \|A\tilde{x} + Bu_F\|_P^2 - \|\tilde{x}\|_P^2 + \|\tilde{x}\|_Q^2 + \|u_F\|_R^2 \\
 &= \|A_k\tilde{x} - B(e - u^*)\|_P^2 - \|\tilde{x}\|_P^2 + \|\tilde{x}\|_Q^2 + \|-K\tilde{x} + u^* - e\|_R^2 \\
 &= (A_k\tilde{x} - B(e - u^*))^\top P(A_k\tilde{x} - B(e - u^*)) - \tilde{x}^\top P\tilde{x} + \tilde{x}^\top Q\tilde{x} + \\
 &\quad (-K\tilde{x} + u^* - e)^\top R(-K\tilde{x} + u^* - e) \\
 &= \tilde{x}^\top A_k^\top P A_k \tilde{x} - \tilde{x}^\top A_k^\top P B(e - u^*) - (e - u^*)^\top B^\top P A_k^\top \tilde{x} + (e - u^*)^\top B^\top P B(e - u^*) - \\
 &\quad \tilde{x}^\top P \tilde{x} + \tilde{x}^\top Q \tilde{x} + \tilde{x}^\top K^\top R K \tilde{x} - \tilde{x}^\top K^\top R(u^* - e) - (u^* - e)^\top R K \tilde{x} + (u^* - e)^\top R(u^* - e) \\
 &= \tilde{x}^\top (A_k^\top P A_k - P + Q + K^\top R K) \tilde{x} + \tilde{x}^\top (K^\top R - A_k^\top P B)(e - u^*) + \\
 &\quad (e - u^*)^\top (R K - B^\top P A_k) \tilde{x} + (e - u^*)^\top (B^\top P B + R)(e - u^*) \tag{6.24}
 \end{aligned}$$

Since  $K = (B^\top P B + R)^{-1} B^\top P A$ , then <sup>4</sup>

$$R K - B^\top P A_k = 0 \tag{6.25}$$

$$K^\top R - A_k^\top P B = 0 \tag{6.26}$$

$$A_k^\top P A_k - P + Q + K^\top R K = 0 \tag{6.27}$$

Now (6.24) can be expressed as:

$$F(\tilde{x}^+) - F(\tilde{x}) + \ell(\tilde{x}, u_F) \leq \|B^\top P B + R\|_2 \cdot \xi^2 \quad \forall x \in \mathbb{X}_F. \tag{6.28}$$

Here,  $\|e - u^*\|_2 \leq \xi$ . Since  $\lambda_{\min}(Q)\|\tilde{x}\|^2 \leq \|\tilde{x}\|_Q^2 = \tilde{x}^\top Q \tilde{x} \leq \lambda_{\max}(Q)\|\tilde{x}\|^2$  and  $\ell(\tilde{x}, u_F) \geq \alpha_3(\|\tilde{x}\|^2) = \lambda_{\min}(Q)\|\tilde{x}\|^2$ , we have

$$\Delta F(x) = F(\tilde{x}^+) - F(\tilde{x}) \leq -\lambda_{\min}(Q)\|\tilde{x}\|^2 + \|B^\top P B + R\|_2 \cdot \xi^2. \tag{6.29}$$

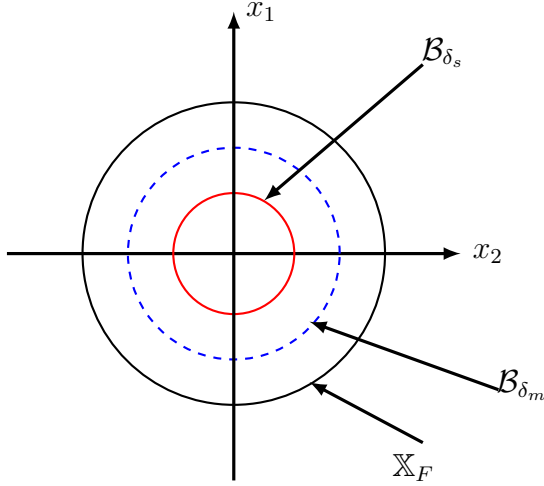
According to  $\Delta F(x) \geq 0$ , we can define the smallest invariant  $\mathcal{B}_{\delta_s} = \{\tilde{x} \in \mathbb{X}_F \mid \|\tilde{x}\|_2 \leq \delta_s = \sqrt{\frac{\|B^\top P B + R\|_2}{\lambda_{\min}(Q)}} \cdot \xi\}$ , which is marked in red in Figure 6.1. It means that the Lyapunov function inside  $\mathcal{B}_{\delta_s}$  is monotonically increasing, e.g.,  $F(x(7)) < F(x(8))$  in Figure 6.2. Consequently, the trajectory may move out of  $\mathcal{B}_{\delta_s}$  but is still confined within  $\mathcal{B}_{\delta_m}$  based on (6.30), e.g.,  $x(8) \in \mathcal{B}_{\delta_s}$  and  $x(9) \in \mathcal{B}_{\delta_m}$  in Figure 6.2. According to (6.23), such maximal outward region is defined as

$$\begin{aligned}
 \delta_m &\leq \|A_k \delta_s - B(e - u^*)\|_2 \leq \|A_k\|_2 \delta_s + \|B\|_2 \|e - u^*\|_2 \\
 &\leq (\|A_k\|_2 \sqrt{\frac{\|B^\top P B + R\|_2}{\lambda_{\min}(Q)}} + \|B\|_2) \xi \tag{6.30}
 \end{aligned}$$

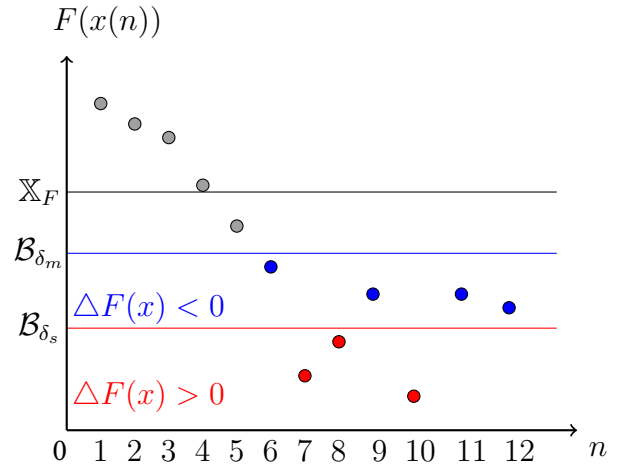
<sup>4</sup>Induction process is in Appendix A.4.

Note that in  $\mathcal{B}_{\delta_m} \setminus \mathcal{B}_{\delta_s}$ , the Lyapunov function is monotonically decreasing, e.g.,  $F(x(12)) < F(x(11))$ . The control invariant is finally set as  $\mathcal{B}_{\delta_m} = \{\tilde{x} \in \mathbb{X}_F \mid \|\tilde{x}\|_2 \leq \delta_m\}$ , where the terminal region is defined as

$$\mathbb{X}_F := \left\{ x \in \mathbb{R}^n : \|x - x^*\|_2 \leq \delta_F = \frac{\max_{u \in \mathcal{U}} (\|u - u^*\|_2)}{\|K\|_2} \right\}$$



**Figure 6.1.:** Minimal and maximum invariant sets within terminal region

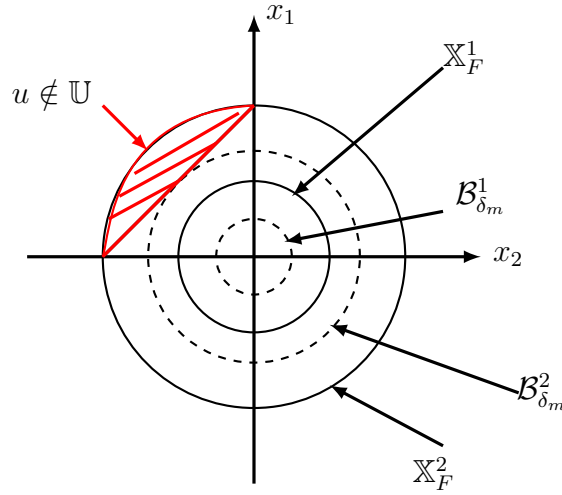


**Figure 6.2.:** Relationship among  $\mathcal{B}_{\delta_s}$ ,  $\mathcal{B}_{\delta_m}$  and  $\mathbb{X}_F$

### Remark 6.7

The designed local controller is not implemented but only required to show practical stability along with the constructed terminal region and ultimate invariant sets. Also,  $\mathcal{B}_{\delta_m} \subset \mathbb{X}_F$  must be guaranteed by the chosen of parameters in (6.30) which influences the size of the practical stability region. We would like to point out that in the FCS-MPC, the discretization error  $e$  can be approximated by the bound  $\xi$  based on a finite set of real numbers [212]. If we use this method and consider the worst case for the bound (6.30), the result would be much more conservative. Consequently, the associated terminal region that encompasses a practical stability region is enlarged. This stability constraints may be in conflict with hard constraints, which may let the terminal control law loose effect, cf., Figure 6.3, where  $\mathcal{B}_{\delta_m}^2$  and  $\mathbb{X}_F^2$  are over conservative regions compared to  $\mathcal{B}_{\delta_m}^1$  and  $\mathbb{X}_F^1$ . In this case, choosing a set of appropriate parameters in (6.30) becomes much more complicated. Note that such a terminal region is difficult to construct in the time-varying case because linearization and the LQR technique do not lead to an Algebraic Riccati Equation.

To this end, in next section, we quantitatively analyze practical stability by the trajectory-dependent approach to estimate the degree of suboptimality in the framework of unconstrained MPC scheme.



**Figure 6.3.:** Relationship between terminal region and practical stability region

### 6.2.2. UNCONSTRAINED MPC SCHEME

#### Proposition 6.8

Consider a feedback control law:  $\kappa_N : X \rightarrow U$  and the associated closed-loop trajectory  $x(\cdot)$  according to (3.26). Assume that

$$V_N(x(n)) \geq V_N(x(n+1)) + \min\{\alpha(\ell(x(n), \kappa_N(x(n))) - \varepsilon), \ell(x(n), \kappa_N(x(n))) - \varepsilon\} \quad (6.31)$$

holds with some  $\alpha \in (0, 1]$ ,  $\varepsilon > 0$  and all  $n \in \mathbb{N}_0$ . For all time instants  $n \in [n_1, n_2]$ ,  $\ell(x(n), \kappa_N(x(n))) \geq \varepsilon$  holds. We define  $V_N(x(n_2+1)) := V_N^\varepsilon$  and  $\bar{\ell}(x(n), \kappa_N(x(n))) := \max\{\ell(x(n), \kappa_N(x(n))) - \varepsilon, 0\}$ , as well as the associated optimal value via

$$\bar{V}_N^{\kappa_N}(x(n)) := \sum_{j=n}^{n_2} \bar{\ell}(x(j), \kappa_N(x(j))), \quad n \in [n_1, n_2]$$

Then the estimate

$$\alpha \bar{V}_N^{\kappa_N}(x(n)) \leq V_N(x(n)) - V_N^\varepsilon \leq V_\infty(x(n)) - V_N^\varepsilon, \quad n \in [n_1, n_2] \quad (6.32)$$

holds for the feedback controller.

The proof has been given in [130]. Proposition 6.8 indicates that in the time interval  $n \in [n_1, n_2]$ , like  $[0, 6]$  and  $[10, 12]$  in Figure. 6.4,  $\ell(x(n), \kappa_N(x(n)))$  decreases monotonously. The procedure with respect to estimate of  $\alpha$  in an a posteriori way is given in Algorithm 5.

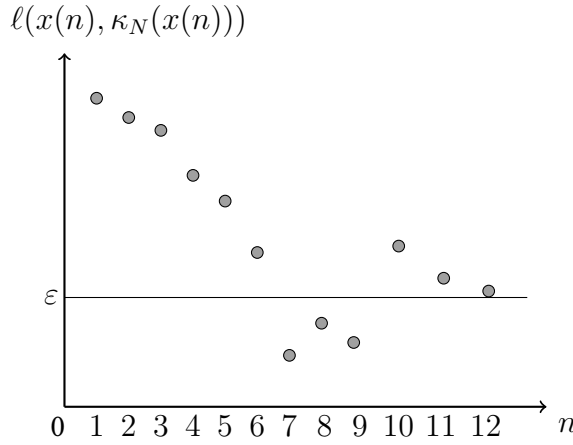
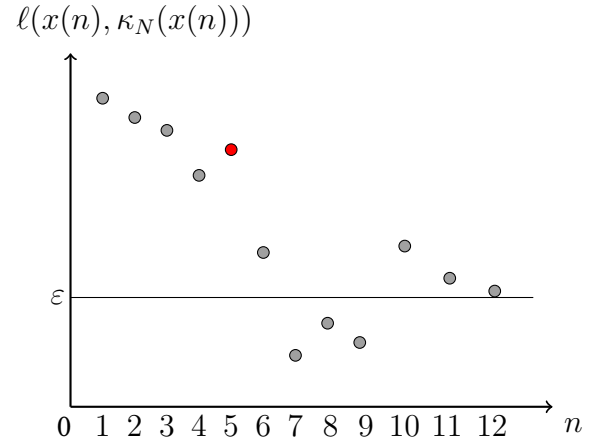


Figure 6.4.: Illustration for Proposition 6.8

Figure 6.5.: Local increase before reaching  $\varepsilon$ 


---

**Algorithm 5** Practical estimate of the degree of suboptimality  $\alpha$  in a posterior way
 

---

**Require:**  $\varepsilon$ .

- 1: **Given**  $N \in \mathbb{N}$ , set  $\alpha_{min} = 1$
- 2: **for**  $n = 0, \dots$  **do**
- 3:   Measure current WIP levels  $x(n)$  and set  $x_0 := x(n)$
- 4:   Compute control inputs  $u(n)$  by solving optimal control problem (3.25)
- 5:   Obtain the optimal control sequence  $u_n^*(\cdot)$
- 6:   Set  $V_N(x_0) = J_N(x_0, u_n^*(\cdot))$ ,  $\kappa_N(x(n)) = u_n^*(0)$
- 7:   Compute the closed-loop dynamics (3.26) to get  $x(n+1)$
- 8:   Set  $x_0 = x(n+1)$  and recompute (3.25) via step 4
- 9:   Set  $V_N(n+1) = J_N(x_0, u_{n+1}^*(\cdot))$
- 10:   **if**  $\ell(x(n), \kappa_N(x(n))) \geq \varepsilon$  **then**
- 11:     Set  $\alpha = \frac{V_N(x(n)) - V_N(x(n+1))}{\ell(x(n), \kappa_N(x(n))) - \varepsilon}$
- 12:     **if**  $\alpha > 1$  **then**
- 13:       Set  $\alpha = 1$  % The estimates in practical case become conservative and may be more than 1
- 14:     **end if**
- 15:     **if**  $\alpha < 0$  **then**
- 16:       Print "the solution may be unstable"
- 17:     **end if**
- 18:   **else**
- 19:     Set  $\alpha = 1$  % Due to reaching into the practical stability region
- 20:     Set  $\alpha_{min} = \min\{\alpha, \alpha_{min}\}$  % Update the estimate of closed-loop system
- 21:   **end if**
- 22:   Apply  $\kappa_N(x(n)) = u_n^*(0)$  to workstations
- 23: **end for**

**Output:**  $\alpha_{min}$  – The minimum of suboptimality degree of the visited states
 

---

Compared to the a posteriori estimate method which needs to compute an additional optimal control problem  $V_N(n+1)$  (i.e., line 8 in Algorithm 5), in what follows, we introduce an a priori way, which allows us to conduct such an estimate at the current time instant. Here, we only present the basic assumptions and the derived Algorithm 6 concerning the a priori estimate of  $\alpha$ . Regarding the detailed proof and comprehensive analysis, we refer to [130].

**Assumption 6.9**

Given  $N \geq 2$ ,  $N \in \mathbb{N}$ , if there exists some  $\gamma > 0$  and  $\varepsilon > 0$  such that the following inequalities

$$\begin{aligned} V_2(x_{u_N}(N-2, x(n))) &\leq \max\{V_1(x_{u_N}(N-2, x(n))) + \varepsilon, (\gamma+1)V_1(x_{u_N}(N-2, x(n))) + \\ &(1-\gamma)\varepsilon\} \text{ and} \\ V_k(x_{u_N}(N-k, x(n))) &\leq \max\{\ell(x_{u_N}(N-k, x(n))), \kappa_k(x_{u_N}(N-k, x(n))) + (k-1)\varepsilon, \\ &(\gamma+1)\ell(x_{u_N}(N-k, x(n))), \kappa_k(x_{u_N}(N-k, x(n)))\} + (k-1-\gamma)\varepsilon \\ \text{hold for all } k &= 3, \dots, N \text{ and all } n \in \mathbb{N}. \end{aligned} \quad (6.33)$$

If Assumption 6.9 holds, then the condition (6.31) is satisfied. Compared to the former, we need to additionally compute  $V_1(x_{u_N}(N-2, x(n)))$ , which is termed as one-step optimization problem and can be solved by Levenberg-Marquardt or the Gauss-Newton algorithm [218]. Note that there may be a tradeoff between  $\varepsilon$  and  $\gamma$ , cf., [156]. Using an a priori estimate based on Assumption 6.9 may yield a rather conservative estimate compared to the a posteriori estimate. In order to remedy this, the Assumption 6.9 was extended incorporating an additional parameter  $N_0$  where  $2 \leq N_0 \leq N$  to improve the accuracy of estimate, cf., [Assumption 19 [130]]. This “best”  $N_0$  is unknown which is closely related to computational cost due to solving  $N_0 - 2$  additional optimal control problem with the associated feedback law  $\kappa_j, j = 1, \dots, N_0 - 1$ . Yet, if the designed stage cost is independent of  $u$ , then it is unnecessary to compute  $u$ . In this case, different  $N_0$  can be simply evaluated for the maximization of  $\alpha$  based on the derived closed-loop trajectory [130]. The associated algorithm concerning the practical priori estimate including  $N_0$  was given in [Algorithm 3.40 [218]].

We would like to point out that since the upper bound of  $N_0$  is determined by  $N$ , the longer  $N$  is, the more opportunities obtained for achieving a “good” estimate through testing  $N_0$ . A comparison concerning the a posteriori estimate and the a priori estimate was conducted in terms of required prediction horizon for closed-loop performance and computational time (cf., [page 331 [107]]), the results showed that due to the derived more conservative estimate, using the a priori estimate typically need a large  $N$  to improve  $\alpha$  for performance guarantee (i.e.,  $\alpha > \bar{\alpha}$ ). Yet, this may be intractable if the prediction horizon itself for stability is required to be “sufficiently” large to ensure  $\alpha > 0$ , cf., [example 10.14

[107]] in which  $N = 70$  is needed at its initial value. We would like to point out this may occur in our case in which the initial WIP is at a high level. As a consequence, the short prediction horizons may result in an negative estimate of degree of suboptimality. To improve the closed-loop performance and guarantee the stability, a large  $N$  may be needed in the period of high WIP load. Yet, this would in turn drastically aggravate computational burden referring to solving an NP-hard optimization problem. Considering the computational cost along with the increase of the length of prediction horizon for stability, the strict one step decreased Lyapunov inequality condition can be extended and replaced with a flexible Lyapunov function including  $k$  steps, which may eases this tight situation and guarantee the system stability with a possibly short prediction horizon. We will discuss this in Section 6.2.6.

---

**Algorithm 6** Practical estimate of the degree of suboptimality  $\alpha$  in a priori way

---

**Require:**  $\varepsilon$

```

1: Given  $N \in \mathbb{N}$ , set  $\alpha_{min} = 1$ 
2: for  $n = 0, \dots$  do
3:   Set  $\gamma = 0$ , compute  $V_1(x_{u_N}(N - 2, x(n)))$ 
4:   if  $V_2(x_{u_N}(N - 2, x(n))) > V_1(x_{u_N}(N - 2, x(n))) + \varepsilon$  or  $V_1(x_{u_N}(N - 2, x(n))) > \varepsilon$ 
       then
5:     Set  $\gamma = \frac{V_2(x_{u_N}(N-2,x(n))) - V_1(x_{u_N}(N-2,x(n))) - \varepsilon}{V_1(x_{u_N}(N-2,x(n))) - \varepsilon}$ 
6:   end if
7:   for  $k = 3, \dots, N$  do
8:     if  $V_k(x_{u_N}(N - k, x(n))) - \ell(x_{u_N}(N - k, x(n)), \kappa_k(x_{u_N}(N - k, x(n)))) > (k - 1)\varepsilon$ 
       or  $\ell(x_{u_N}(N - k, x(n)), \kappa_k(x_{u_N}(N - k, x(n)))) > \varepsilon$  then
9:       Set  $\gamma_k = \frac{V_k(x_{u_N}(N-k,x(n))) - \ell(x_{u_N}(N-k,x(n)), \kappa_k(x_{u_N}(N-k,x(n)))) - (k-1)\varepsilon}{\ell(x_{u_N}(N-k,x(n)), \kappa_k(x_{u_N}(N-k,x(n)))) - \varepsilon}$ 
10:      Set  $\gamma = \max\{\gamma, \gamma_k\}$ 
11:    end if
12:  end for
13:  if  $\gamma = 0$  then
14:    Set  $\alpha = 1$ , Print "Practical region reached"
15:  else
16:    Set  $\alpha = \frac{(\gamma+1)^{N-2} - \gamma^N}{(\gamma+1)^{N-2}}$ 
17:    if  $\alpha < 0$  then
18:      Print "Solution may be unstable"
19:    end if
20:    Set  $\alpha_{min} = \min\{\alpha, \alpha_{min}\}$ 
21:  end if
22: end for

```

**Output:**  $\alpha_{min}$

---



Also, we would like to point out that the optimized variables must be integer in this thesis. The aforementioned one step optimization problem is similar to the one used in the finite input set MPC with horizon  $N = 1$ . Yet, the cost functional includes an additional terminal cost and terminal region constraint. In doing so, we typically need to solve an additional optimization problem although the stage cost is independent of  $u$  [127].

### 6.2.3. ESTIMATE OF $\varepsilon$

The decay rate of the tracking error represents the transient response of closed loop performance, and the size of practical stability region stands for the steady-state error. Note that the latter is determined by system parameters (e.g., production rate of RMT and flow probability matrix) together with the disturbance caused by discretization of input, cf., (6.42). The approximately selected  $\varepsilon$  typically needs to ensure that  $\ell(\cdot, \cdot)$  is monotonically decreasing in the time periods like  $[0, 6]$  and  $[10, 12]$  in Figure 6.4. We would like to point out that determination of  $\varepsilon$  is not trivial, in particular due to integer constraints and reconfiguration delays. For instance, if RMTs in the job shop system do not have to be reconfigured in the steady state phase, the practical region determined by  $\xi_m$  in (6.42) is identical to the value derived without reconfiguration delays. Otherwise, the practical region derived with reconfiguration delays is typically larger due to lagged control actions, cf., Figure 7.16. To estimate  $\varepsilon$  especially with unknown external disturbance is an extraordinary task. Currently, such an  $\varepsilon$  is constructed by extensive simulations together with analysis of designed stage cost.

### 6.2.4. INCOMPLETE OPTIMIZATION WITH NP-HARD FOR STABILITY

In Section 6.2, we discussed practical stability with integer constraints. Since the derived solutions refer to solving an NP-hard optimization problem, the latter may be intractable if a long prediction horizon is required for stability or improving performance. To this end, we consider using an incomplete optimization strategy. One way to reduce the computational cost and guarantee the stability is making use of the a priori characteristic value  $\alpha \in (0, 1]$ , cf., Algorithm 7, to search a feasible solution.

Note that in the first iteration we still insist on achieving the optimal solution as a root and then focus on (non)-optimal solutions at the next steps. The stability proof regarding such incomplete optimization is given in [Proposition 7.25 [107]]. Note that in Step 8 of Algorithm 7, the solutions satisfying the inequality may not be found at some time instants with the given  $\alpha$ . In this case, we complete the iterative optimization and use the derived optimal solution.

---

**Algorithm 7** Incomplete NP-hard optimization based on a priori desired degree of suboptimality  $\alpha$

---

- 1: **Given**  $N \in \mathbb{N}$ .  $\varepsilon$
  - 2: **if**  $n = 0$  **then**
  - 3:   Measure current WIP levels  $x(n)$  and set  $x_0 := x(n)$
  - 4:   Solve the optimal control problem (3.25) and obtain the optimal control sequence  $u_n^*(\cdot)$  and the corresponding cost value  $J_N(x_0, u_n^*(\cdot))$
  - 5:   Set  $\kappa_N(x(n)) = u_n^*(0)$  as the feedback law to workstations
  - 6: **end if**
  - 7: **for**  $n = 1, \dots$  **do**
  - 8:   Measure current WIP levels  $x(n)$  and set  $x_0 := x(n)$
  - 9:   Solve the optimal control problem (3.25) and compute control inputs  $u_n^*(\cdot)$  by only ensuring  $J_N(x_0, u_n^*(\cdot)) + \min\{\alpha(\ell(x(n-1), \kappa_N(n-1)) - \varepsilon), \ell(x(n-1), \kappa_N(n-1)) - \varepsilon\} \leq J_N(x(n-1), u_{n-1}^*(\cdot))$  with a priori  $\alpha \in (0, 1]$  by terminating the optimization routine
  - 10:   Apply  $\kappa_N(x(n)) = u_n^*(0)$  to workstations in the next sampling period and iteratively.
  - 11: **end for**
- 

Proposition 6.8 shows that with a stabilizing feedback law  $\kappa_N$ ,  $\ell(x(n), \kappa_N(x(n)))$  can be ensured to decrease monotonically before reaching the practical stability region to ensure  $\alpha \in (0, 1]$  via the constant  $\varepsilon$ . In Figure 6.5, however, we can observe that  $\ell(x(5), \kappa_N(x(5))) > \ell(x(4), \kappa_N(x(4)))$ , and this phenomenon may happen when we consider the reconfiguration delays in our case. If we still utilize the fixed  $\varepsilon$ , then  $\alpha$  may be negative. As a result, using the adaptive MPC technique, the prediction horizon is increased immediately to improve the performance and hence forth  $\alpha$ . Alternatively, we could enlarge  $\varepsilon$  to ensure  $\alpha$  to be non-negative. However, if this local instability occurs far away from the planned level, this estimate would be too conservative and may even be useless, e.g., (fake enlarged region of  $\alpha = 1$  without necessity of improving performance.) Undoubtedly, we hope such region to be as small as possible, which represents a good tracking performance with small steady-state error. In this case, one simple but applicable way is to redesign the stage cost by a trial and error procedure via numerical simulations to ensure monotonicity before reaching the practical stability region. Alternatively, we could relax the classical conservative Lyapunon fuction and use the principle of a flexible Lyapunov function [128]. The details of the latter will be discussed in Section 6.2.6. This phenomenon also had been pointed out in Remark 3.18 that the decreased cost function in the Algorithm 7 is not a necessary condition for (6.34). Regarding it, we cite the following definition with respect to attractivity and stability [107].

**Definition 6.10**

Given a set  $Y \subset \mathbb{X}$ , the closed-loop trajectory  $x^+ = f(x, \kappa_N(x))$  is said to be attractive

on  $Y$  if for each  $x \in Y$ ,  $\lim_{n \rightarrow \infty} \|x(n) - x^*\| \leq \delta$ . In addition, the closed-loop system is practically stable on  $Y$  if there exists  $\alpha_1 \in \mathcal{K}$ ,  $\xi(n) > 0$ ,  $n \in \mathbb{N}$  such that

$$\|x(n) - x^*\| \leq \alpha_1(\|x_0 - x^*\|) + \xi(n) \quad (6.34)$$

This will be verified in (6.39), where  $\beta(\cdot, n) \in \mathcal{K}$ .

### 6.2.5. ANALYTICAL APPROXIMATE ESTIMATE OF PRACTICAL REGION

In Section 6.1.2, the optimal feedback law  $u(n) = \Phi(x(n) - x^*) + u^* \in \mathbb{U}$  (6.13) was derived by unconstrained MPC with continuous variables. Taking integer constraints into account, this is not the case. Particularly, the monotonicity of the Lyapunov stability condition (3.19) may not hold under the reconfiguration delays, cf., Figure 6.5. Nevertheless, in the presence of integer constraints and reconfiguration delays, if the closed-loop trajectory can be convergent as  $n \rightarrow \infty$ , i.e., (6.34) holds, then the Lyapunov function can be used to allow local increases [128, 129]. Here, *the idea is to regard the discreteness of inputs as a disturbance compared to using continuous variables and show practical stability [219]*. Now, recalling the system dynamics (4.3) and utilizing the expression of  $u^*$  (6.3), we have

$$\begin{aligned} x(n+1) &= x(n) + r^{RMT} \cdot P \cdot (\Phi(x(n) - x^*) + u^* + \xi_{i+r}(n)) + P \cdot n^{DMT} \cdot r^{DMT} + d \\ &= x(n) + \overbrace{r^{RMT} \cdot P \cdot \Phi(x(n) - x^*)}^{a_1} + \overbrace{r^{RMT} \cdot P \cdot \xi_{i+r}(n)}^{w(n)} + \\ &\quad \underbrace{r^{RMT} \cdot P \cdot u^* + P \cdot n^{DMT} \cdot r^{DMT} + d}_{=0} \end{aligned} \quad (6.35)$$

Here,  $\xi_{i+r}(n)$  is the bounded “disturbance” caused by integer constraints and reconfiguration delays compared to the solution derived from continuous variables. Through (6.35), we have

$$x(n+1) - x^* = (\mathbf{Id} + a_1 \Phi)(x(n) - x^*) + \overbrace{a_1 \cdot \xi_{i+r}(n)}^{w(n)} \quad (6.36)$$

and proceed using induction

$$\begin{aligned}
x(1) - x^* &= (\mathbf{Id} + a_1\Phi)(x(0) - x^*) + w(0) \\
x(2) - x^* &= (\mathbf{Id} + a_1\Phi)(x(1) - x^*) + w(1) \\
&= (\mathbf{Id} + a_1\Phi)((\mathbf{Id} + a_1\Phi)(x(0) - x^*) + w(0)) + w(1) \\
&= (\mathbf{Id} + a_1\Phi)^2(x(0) - x^*) + (\mathbf{Id} + a_1\Phi)w(0) + w(1) \\
&\vdots \\
x(n) - x^* &= (\mathbf{Id} + a_1\Phi)^n(x(0) - x^*) + (\mathbf{Id} + a_1\Phi)^{n-1} \cdot \overbrace{a_1 \cdot \xi_{i+r}(0)}^{w(0)} + \\
&\quad + (\mathbf{Id} + a_1\Phi)^{n-2} \cdot \overbrace{a_1 \cdot \xi_{i+r}(1)}^{w(1)} + \cdots + (\mathbf{Id} + a_1\Phi)^{n-n} \cdot \overbrace{a_1 \cdot \xi_{i+r}(n-1)}^{w(n-1)} \quad (6.37)
\end{aligned}$$

Consequently,

$$\begin{aligned}
\|x(n) - x^*\|_2 &\leq \|(\mathbf{Id} + a_1\Phi)\|_2^n \|x(0) - x^*\|_2 + \|(\mathbf{Id} + a_1\Phi)\|_2^{n-1} \cdot \|w(0)\|_2 + \\
&\quad \|(\mathbf{Id} + a_1\Phi)\|_2^{n-2} \cdot \|w(1)\|_2 + \cdots + \|w(n-1)\|_2 \quad (6.38)
\end{aligned}$$

Set  $\xi_m = \max\{\|w(k)\|_2\}, k = 0, 1, \dots, n-1$ . Then we have

$$\begin{aligned}
\|x(n) - x^*\|_2 &\leq \|(\mathbf{Id} + a_1\Phi)\|_2^n \|x(0) - x^*\|_2 + (\|(\mathbf{Id} + a_1\Phi)\|_2^{n-1} + \cdots + \|(\mathbf{Id} + a_1\Phi)\|_2^{n-n}) \xi_m \\
&\leq \overbrace{\|(\mathbf{Id} + a_1\Phi)\|_2^n \|x(0) - x^*\|_2}^{\beta(\|x(0)-x^*\|_2, n)} + \overbrace{\frac{1 - \|(\mathbf{Id} + a_1\Phi)\|_2^n}{1 - \|(\mathbf{Id} + a_1\Phi)\|_2} \xi_m}^{\xi(n)} \quad (6.39)
\end{aligned}$$

Given  $a_1$ , we have shown the system with continuous variables would be asymptotically stabilized via (6.17). Since our aim is to show the closed-loop trajectory can be convergent as  $n \rightarrow \infty$ , the tracking error is assumed to be small enough such that  $u(\Phi_1(x)) \in \mathbb{U}$  (6.13) holds. Now, set  $\Phi_1 := \Phi = -(2\lambda \cdot \mathbf{Id} + 2a_1^\top a_1)^{-1} \cdot 2a_1^\top$ , then

$$\begin{aligned}
\mathbf{Id} + a_1\Phi &= \mathbf{Id} + a_1 \cdot (-(2\lambda \cdot \mathbf{Id} + 2a_1^\top a_1)^{-1} \cdot 2a_1^\top) \\
&= \mathbf{Id} - (2\lambda \cdot \mathbf{Id} + 2a_1^\top a_1)^{-1} (2a_1^\top a_1) \\
&= \mathbf{Id} - (2\lambda \cdot \mathbf{Id} + 2a_1^\top a_1)^{-1} (2\lambda \cdot \mathbf{Id} + 2a_1^\top a_1) + 2\lambda \cdot \mathbf{Id} (2\lambda \cdot \mathbf{Id} + 2a_1^\top a_1)^{-1} \\
&= \lambda \cdot \mathbf{Id} (\lambda \cdot \mathbf{Id} + a_1^\top a_1)^{-1} \quad (6.40)
\end{aligned}$$

Under constraints, in order to guarantee the asymptotic stability,  $\|\mathbf{Id} + a_1\Phi\|_2 \in (0, 1)$

must hold. Based on (6.40), we show

$$\begin{aligned} \|\mathbf{Id} + a_1\Phi\|_2 &= \lambda \cdot \|(\lambda \cdot \mathbf{Id} + a_1^\top a_1)^{-1}\|_2 \\ &\geq \frac{\lambda}{\|\lambda \cdot \mathbf{Id} + a_1^\top a_1\|_2} \geq \frac{\lambda}{\lambda + \|a_1^\top a_1\|_2} \end{aligned} \quad (6.41)$$

Then,  $\|\mathbf{Id} + a_1\Phi\|_2 \in (0, 1)$  holds. Then when  $n \rightarrow \infty$ , (6.39) is simplified as

$$\lim_{n \rightarrow \infty} \|x(n) - x^*\|_2 \leq \frac{\overbrace{\xi_m}^{\delta}}{1 - \|(\mathbf{Id} + a_1\Phi)\|_2} \quad (6.42)$$

### 6.2.6. LOCAL INSTABILITY WITH RECONFIGURATION DELAYS

We discussed that the existence of reconfiguration delays may lead to a non-monotonous Lyapunov function. Consequently, the estimate  $\alpha$  in (6.31) may be negative even before reaching the practical stability region. In production systems, this local instability may be acceptable depending on the requirements of manufacturers. To simplify the structure of controller and save computational cost without possibly increasing the prediction horizon too often, we introduce the concept of a flexible Lyapunov function.

#### 6.2.6.1. FLEXIBLE LYAPUNOV FUNCTION

If we are able to show  $V_N(x(n)) \leq \tau$  or  $\lim_{n \rightarrow \infty} \|x(n) - x^*\|_2 \rightarrow \delta$  as  $n \rightarrow \infty$ , cf., Theorem 3.21, then the Lyapunov function is allowed to increase locally but with an average decrease in future steps [129].

#### Definition 6.11

There exists a set of parameters  $\nu_j > 0, j = 1, \dots, k - 1$  such that

$$\begin{aligned} \overbrace{\nu_{k-1} \cdot (V_N(x(n+k)) - V_N(x(n))) + \dots + \nu_1 \cdot (V_N(x(n+2)) - V_N(x(n)))}^{<0} \\ + \overbrace{(V_N(x(n+1)) - V_N(x(n)))}^{>0} < 0 \end{aligned} \quad (6.43)$$

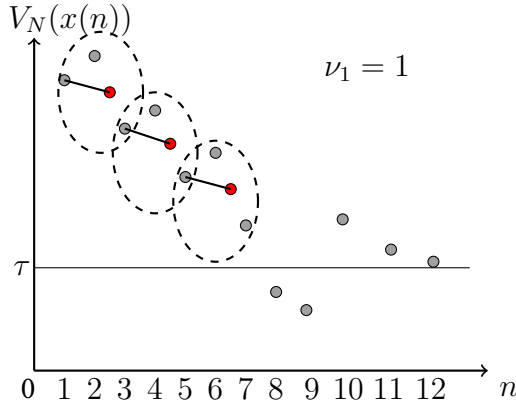
holds, then  $V_N(\cdot)$  is called flexible Lyapunov function.

For simplicity, we consider the basic three steps (6.44)

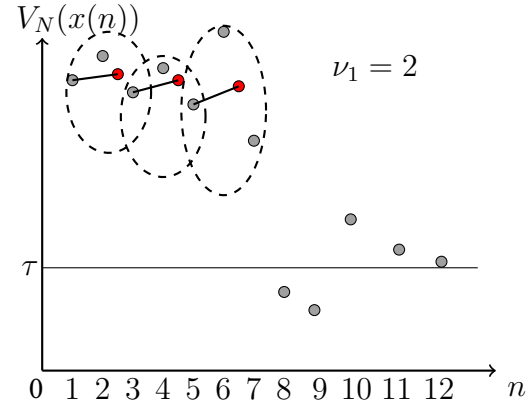
$$\nu_1 \cdot (V_N(x(n+2)) - V_N(x(n))) + (V_N(x(n+1)) - V_N(x(n))) < 0 \quad (6.44)$$

to show how condition (6.43) works with different penalties on the decrease of cost via the illustrated examples, cf., Figure 6.6 and 6.7, respectively. On the left side,  $\nu_1 = 1$  represents  $\frac{1}{2}(V_N(n+1) + V_N(n+2)) < V_N(n)$ , see the connected inclined lines in Figures

6.6. Similarly, on the right side, the penalty with  $\nu_1 = 2$  relaxes the decreased amplitude of  $V_N(n+2) - V_N(n)$  that render (6.43) to hold. Through this simple case, we can find that the flexible Lyapunov function property is closely determined by those coefficients.



**Figure 6.6.:** One step increase but with an average decrease in future two steps



**Figure 6.7.:** One step increase but with a small penalty on the decreased costs

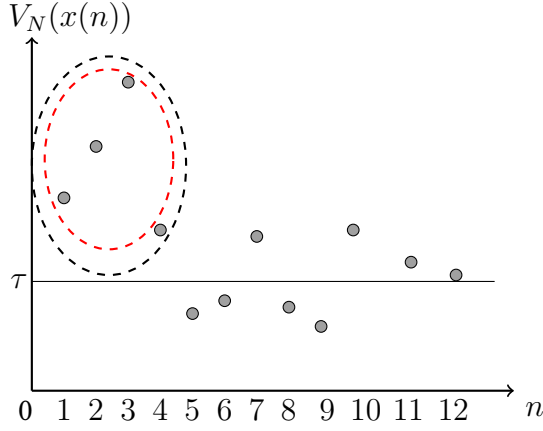
Compared to the general applied flexible Lyapunov function [128], we would like to point out that in their work either  $V(x(n+1))$  or  $V(x(n+2))$  can be less than  $V(x(n))$  such that condition (6.44) holds for searching a Lyapunov function. Here, the optimal cost functional  $V_N$  is set as Lyapunov function. In our case, we only consider condition (6.43) when  $V_N(n+1) > V_N(n)$  occurs with the resulting negative estimate  $\alpha$ . This method is also differs from [220] in which the a priori stability and suboptimality threshold  $\bar{\alpha}$  belongs to  $\bar{\alpha} \in (0, 1)$ . In this way, once  $\alpha < 0$ , the prediction horizon is instantaneously increased to guarantee the stability and improve the closed-loop performance via adaptive horizon NMPC [218].

For the general setting (6.43), given a set of parameters  $\nu_j$ , condition (6.43) may also exhibit different results with different steps  $k$ . For example, in Figure 6.8, if we choose  $k = 2$ , then condition (6.43) does not hold because of  $V_N(x(1)) < V_N(x(2)) < V_N(x(3))$ , see the points encompassed in the red ellipse. In contrast to that, if this step is extended by one to  $k = 3$ , then we observe that  $V_N(x(4)) < V_N(x(1))$ . As a result, the condition (6.43) is satisfied.

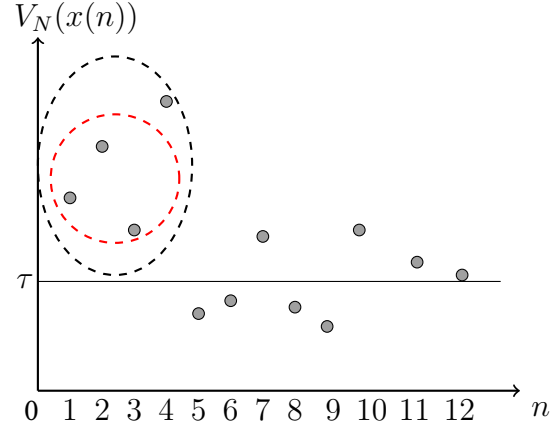
The above case we discussed is with respect to the prolongation of  $k$  for satisfying the flexible Lyapunov function condition. The right side (cf., Figure 6.9) is the inverse case that prolongation of  $k$  may be counterproductive. That is, in the red circle, condition (6.43) may hold within 3 steps under a certain of parameters but may fail due to a large increase of  $V_N(x(4))$ .

Through these two illustrated examples, the main idea we want to show is that given a fixed step  $k$  and a finite set of  $\nu_j$ , we should not verify condition (6.43) at the time instant  $n+k$  which may be too late to bring unexpected results. To cope with it, we adopt the receding strategy to iteratively check it. That is, if condition (6.43) is satisfied at any time

instants before  $n+k$ , then we immediately discard of the inspections of the next steps. This strategy is similar with the principle of MPC itself that uses a rolling time window.



**Figure 6.8.:** Prolongation of  $k$  satisfies flexible Lyapunov equation



**Figure 6.9.:** Prolongation of  $k$  may render flexible Lyapunov function fail

### 6.2.6.2. RELAXED AND FLEXIBLE LYAPUNOV INEQUALITY

Generally, the relaxed Lyapunov inequality is used for analyzing the stability via estimates of suboptimality in each step. In this thesis, we extend it to  $k$  steps through using the principle of a flexible Lyapunov function. Considering the possible local instability and computational burden, we iteratively check condition (6.43) for the further determination of increasing prediction horizon to improve the closed-loop performance. Again, we first consider the simple case (6.45) within three steps and then extend it to the general one. For simplicity of exposition, we assume  $\nu_1 = 1$

$$\overbrace{(V_N(x(n+2)) - V_N(x(n)))}^{<0} + \overbrace{(V_N(x(n+1)) - V_N(x(n)))}^{>0} < 0 \quad (6.45)$$

Note that we are only concerned with local instability before reaching the practical stability region. Therefore, the flexible Lyapunov function is only triggered in the area of  $\ell(x(\cdot), u(\cdot)) \geq \varepsilon$ . In order to be easily understood, we sketch Figure 6.10 for interpreting all possible cases. For saving space, we use  $\alpha_{n+j}^N, j = 1, \dots, k-1$  in Figure 6.10. Assume  $V_N(x(n+1)) - V_N(x(n)) > 0$  occurs with the resulting  $\alpha_n^N < 0$ . At the next time instant  $n+1$ , we have three possible situations, one for  $\alpha_{n+1}^N < 0$  and two for  $\alpha_{n+1}^N > 0$  depending on the position of  $V_N(n+2)$ . If  $\alpha_{n+1}^N > 0$ , we have

$$V_N(x(n+2)) \leq V_N(x(n+1)) - \alpha_N(n+1)\ell(x(n+1), \kappa_N(x(n+1))) - \varepsilon. \quad (6.46)$$

Note that we set  $\alpha_N(\cdot) = 1$  if  $\alpha_N(\cdot) > 1$  in case of practical stability. Now we utilize the upper bound of  $V_N(n+2)$  and substitute (6.46) into (6.45) to obtain

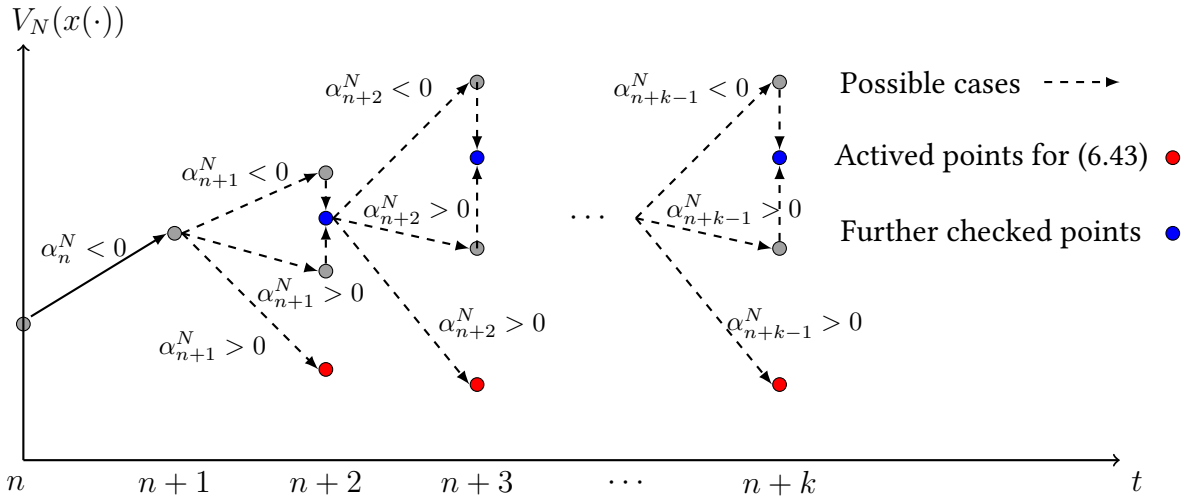
$$-\alpha_N(n+1)(\ell(x(n+1), \kappa_N(x(n+1))) - \varepsilon) + 2(V_N(x(n+1)) - V_N(x(n))) < 0. \quad (6.47)$$

Similar to the a priori estimate, we could judge the condition (6.45) with all available information (6.47) at the time instant  $n+1$ . Note that if this condition holds, then we do not need to evaluate condition (6.45) until  $n+2$  with the derived  $V_N(x(n+2))$ . Otherwise, we still need to determine the relationship between  $V_N(x(n+2))$  with  $V_N(x(n))$  at  $n+2$ , which can be considered as an a posteriori check. The detailed discussion is presented in Algorithm 8. Now we extend (6.46) and (6.47) to the general setting. If  $\alpha_N(n+k-1) > 0$ , we have

$$V_N(x(n+k)) \leq V_N(x(n+k-1)) - \alpha_N(n+k-1)\ell(x(n+k-1), \kappa_N(x(n+k-1))) - \varepsilon) \quad (6.48)$$

Substituting (6.48) into (6.43), we have

$$\begin{aligned} & -\nu_{k-1} \cdot \alpha_N(n+k-1)(\ell(x(n+k-1), \kappa_N(x(n+k-1))) - \varepsilon) + (\nu_{k-1} + \nu_{k-2}) \cdot \\ & (V_N(x(n+k-1)) - V_N(x(n))) + \dots + \nu_1 \cdot (V_N(x(n+2)) - V_N(x(n))) + \\ & (V_N(x(n+1)) - V_N(x(n))) < 0 \end{aligned} \quad (6.49)$$



**Figure 6.10.:** General case for determination of flexible Lyapunov function within  $k$  step

Using the similar principle, Algorithm 8 can be extended to the general case, cf., Figure 6.10, where red marks represent the flexible Lyapunov function to hold. For simplicity, we use blue marks to represent the two situations: One from  $\alpha_{n+j-1}^N < 0$ , it means that



$V_N(x(n+j-1)) < V_N(x(n+j))$ ). The other from  $\alpha_{n+j-1}^N > 0$  which stands for  $V_N(x(n+j-1)) > V_N(x(n+j))$ . But for both cases,  $V_N(x(n+j)) > V_N(x(n))$ ,  $j = 1, \dots, k$ , which needs to be checked further in the framework of flexible Lyapunov inequality condition (6.43).

---

**Algorithm 8** Online estimate via relaxed and flexible Lyapunov function  $k = 2$

---

```

1: Given  $N, \nu_j, k$ 
2: for  $n = 0, 1 \dots$  do
3:   Execute the Algorithm 6 for a priori online estimate
4:   if  $\alpha_N(n) < 0$  (line 17 in algorithm 6) then
5:     First keep prediction horizon  $N$  unchanged, store computed value  $V_N(n)$  and continue running algorithm 6 to the next time instant  $n + 1$ 
6:     if  $\alpha_N(n + 1) > 0$  then
7:       Inequality (6.46) holds. Check condition (6.45) through (6.47) utilizing the upper bound of  $V_N(x(n + 2))$ 
8:       if Condition (6.47) holds then
9:         Condition (6.45) is satisfied, flexible Lyapunov function works
10:      else
11:        Continue to the next time instant  $n + 2$  and compute  $V_N(x(n + 2))$  and check (6.47)
12:        if (6.47) holds then
13:          Conclude  $V_N(x(n + 2)) < V_N(x(n)) < V_N(x(n + 1))$ , flexible Lyapunov function works
14:        else
15:          Conclude  $V_N(x(n)) \leq V_N(x(n + 2)) < V_N(x(n + 1))$ , increase  $N$  to  $\bar{N}$  to improve  $\alpha_N$ 
16:        end if
17:      end if
18:    else
19:       $V_N(x(n)) < V_N(x(n + 1)) \leq V_N(x(n + 2))$ , increase  $N$  to  $\bar{N}$  to improve  $\alpha_N$ 
20:    end if
21:  end if
22: end for

```

---

$\nu_j$  and  $k$  are two factors for the determination of flexible Lyapunov function condition. Given the step  $k$ , if the computational cost is predominant and we do not want to increase  $N$  often, then one simple way is setting  $\nu_1 < \nu_2 < \dots < \nu_{k-1}$ . Yet, it is still related to the matched decreased / increased cost values in (6.43). For sake of safety and to make (6.43) hold, a strong assumption concerning flexible Lyapunov condition is given in Assumption 6.12. Additionally, since we adopt the iteratively checking strategy, in this way, the increase

of  $N$  is only triggered when  $V_N(x(n + \hat{j})) > V_N(x(n))$ , for all  $\hat{j} = 1, \dots, k$ .

**Assumption 6.12**

Given the step  $k$  and regardless of  $\nu_j$ , at least one of the decreased Lyapunov cost  $V_N(x(n + j)) - V_N(x(n))$ ,  $j = 2, \dots, k$  holds.

The prerequisite for using flexible Lyapunov function is to ensure  $V_N(x(n)) \rightarrow \tau$  as  $n \rightarrow \infty$ . It means that even though we do not change the horizon  $N$  for improving the performance, we still conclude that there exists a finite step  $\bar{k} < k = \infty$  that renders  $V_N(x(n + \bar{k})) < V_N(x(n))$  to hold through iteratively checking. Yet, this is impractical. Similarly, if we set the prediction horizon  $N$  to  $\infty$ , this local instability may not occur. It seems that there exists a relationship between  $N$  and  $k$ . The longer  $N$  is, the shorter step  $k$  can be. Inversely, the shorter  $N$  is, the longer step  $k$  may be required such that condition (6.43) holds. Considering those local instability phenomenons may be mainly attributed to the inadequate horizon with reconfiguration delays. Given a finite  $k$ , if the flexible Lyapunov function is not satisfied, then the prediction is increased to improve performance. Now, one question naturally arise. Under the increased  $\bar{N}$ , does the performance improve?. More specifically, does  $V_{\bar{N}}(x(n + \bar{i})) < V_N(x(n))$  holds, for  $\bar{N} > N$ ? Since all information in (6.43) to be checked are the visited values, the computational time will not be aggravated. However, we cannot give the accurate information regarding the new  $\bar{N}$  but have to iteratively check it. In this way, the performance cannot be guaranteed.

To cope with it, we combine the adaptive NMPC with flexible Lyapunov function that evaluates condition (6.43) at the current time instant  $n$  with future information and adaptively changes  $N$  such that this condition holds.

**6.2.6.3. ADAPTIVE NMPC WITH FLEXIBLE LYAPUNOV FUNCTION**

**Definition 6.13** (Flexible Lyapunov function)

Given a finite step  $k \geq 2$  and a set of coefficients  $\nu_j > 0$ ,  $j = 1, \dots, k - 1$ . If  $V_{N_n}(x(n + 1)) - V_{N_n}(x(n))$  occur with the resulting  $\alpha_{N_n} < 0$  before reaching the practical stability region, assume there exists  $N_n \geq 2$ ,  $N_n \in \mathbb{N}_0$  such that

$$\overbrace{\nu_{k-1} \cdot (V_{N_n}(x(n + k)) - V_{N_n}(x(n))) + \dots + \nu_1 \cdot (V_{N_n}(x(n + 2)) - V_{N_n}(x(n)))}^{<0} + \overbrace{(V_{N_n}(x(n + 1)) - V_{N_n}(x(n)))}^{>0} < 0 \quad (6.50)$$

holds, where  $x(\cdot)$  is closed-loop trajectory,  $V_{N_n} : \mathbb{N}_0 \times \mathbb{X} \rightarrow \mathbb{R}_{\geq 0}$ , then  $V_{N_n}$  is called flexible Lyapunov function.

In order to verify (6.50), the disadvantage is that we have to compute additional multiple optimal control problems at  $n$  (e.g.  $V_{N_n}(n + 1), \dots, V_{N_n}(n + k)$ ). But the advantage is that we do not need to compute all these values at once but can iteratively check

them technique. If the condition (6.50) does not hold with the current  $N_n$ , then increase  $N_n$  by one  $N_n \leftarrow N_n + 1$  and check (6.50). Note that typically under a longer horizon with the expected improved performance, the needed checked step to make (6.50) holds may be reduced, which also saves computational time in another light. After a finite step  $N_n \leftarrow N_n + i$ , condition (6.50) will be satisfied, cf., Remark 3.22. Using this strategy to adaptively change  $N$ , the closed-loop performance is guaranteed. Yet, it may suffer from computational complexity especially with a large  $N_n$ . Regarding the computational cost from increasing  $N_n$ , the main difference between this method with adaptive NMPC [218] is that when  $\alpha_{N_n}(n) < 0$  this method is used to solve mutiple additional optimal control problem with  $N_n$ , the conditions may be satisfied without increasing  $N_n$ .

### 6.3. SUMMARY

In this chapter, we systematically analyzed the stability qualitatively and quantitatively via two categories of stabilizing MPC controllers. We summarize the main results as follow:

1. With continuous variables and without integer constraints and reconfiguration delays for asymptotic stability

- Terminal endpoint constraints are easy to be implemented and recursive feasibility can be guaranteed. Yet, the additional terminal constraints may weaken the operating region and deteriorate the overall performance. Although the stability may be guaranteed for a short prediction horizon, it typically needs a large prediction horizon to guarantee a large feasible sets of initial conditions. However, forcing  $x^* = f(x^*, u^*)$  is almost surely impossible to be achieved with the discretization of inputs when we take integer constraints into account.
- Without using terminal conditions requires a controllability condition with respect to the optimal cost functional or the stage cost. Stability can be ensured by a sufficiently large prediction horizon and this “sufficient” may be achieved with the possible shortest prediction horizon (6.17). This is significant in dealing with NP-hard optimization problems where the computation burden is predominant.

2. With integer constraints and reconfiguration delays for practical (asymptotic) stability

- Using terminal cost and terminal region still typically requires a large prediction horizon to guarantee a large feasible set of initial conditions and may be impossible while solving the NP-hard optimization problem. Also, the terminal region needs to be pre-computed off-line and the practical stability region has to be compassed within the designed terminal region.
- Using a trajectory-dependent approach in the context of unconstrained MPC scheme, online estimating the degree of suboptimality allowed us to quantitatively analyze the

influence of the reconfiguration delays on key performance indicator such as, e.g., setting time, during the transient phase, and on the steady-state performance via the size of the practical stability region. Particularly, we were able to integrate the adaptive NMPC with flexible Lyapunov function for handling the possible local instability phenomenon caused by reconfiguration delays, but it may suffer from computational complexity.

# 7

## NUMERICAL RESULTS

After the development and implementation of a prototype application, a respective implementation needs to be tested thoroughly, which is basically corresponding to Chapter 6. In Section 7.1, according to the derived stability results in case of continuous assignment of RMTs for controlling WIP from Section 6.1, we first demonstrate the effectiveness of proposed method via comparison between MPC with terminal endpoint constraints and PID. After that, a sensitivity analysis is carried out which includes initial conditions, the length of prediction horizon and the dynamic flow control. Then, we additionally present results for unconstrained MPC for comparison with MPC with terminal endpoint constraints, which forms a cornerstone for considering practical stability. After the quantitative analysis with continuous variables, we quantitatively conduct the comparison concerning WIP tracking and usage of RMTs by means of different integer control strategies associated the MPC and also via PID as the benchmark in Section 7.2.1. Furthermore, for the closed-loop trajectory controlled by MPC, we show practical stability via an a posteriori estimate of the degree of suboptimality in Section 7.2.2. Based on such estimate, we quantitatively analyze the closed-loop in terms of transient and steady state performance in the cases with and without reconfiguration delays in Section 7.2.3. Similarly to the former, a sensitivity analysis is conducted in the presence of integer constraints and reconfiguration delays in Section 7.2.4. Last, in Section 7.2.4.3, through a simple case, we demonstrate that the reconfiguration delays may render the system to be unstable and show how the flexible Lyapunov function works combined with adaptive MPC and the relationship between a “good” stage cost with stability.

Parts of this chapter have been published in [6, 162].

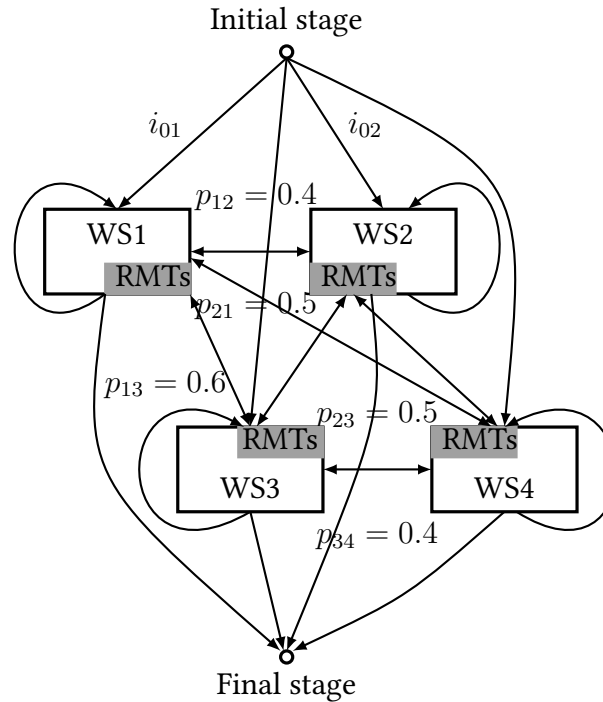
## 7.1. ASYMPTOTIC STABILITY WITHOUT INTEGER CONSTRAINTS AND DELAYS

To complement and check our theoretical findings presented in Section 6.1.1, we consider an example of a job shop manufacturing process that is sketched in Figure 7.1, which is composed of four workstations (1-turning, 2-drilling, 3-milling, 4-grinding) and each workstation contains several DMTs and RMTs. While DMTs can only operate in its own specific workstations, RMTs may be reassigned to other workstations by reconfiguration. The respective dynamics are given by (4.3) with parameters setting in Table 7.1.

**Table 7.1.:** The variables definition in the job shop system

Variable	Description
$x(0) = [40 \ 40 \ 40 \ 30]^\top$	Work in process (WIP) level
$r^{DMT} = 3$	Production rate of DMT
$r^{RMT} = 2$	Production rate of RMT
$n^{DMT} = [5 \ 4 \ 5 \ 2]^\top$	Number of DMTs for each workstation
$u = [2 \ 1 \ 2 \ 1]^\top$	Number of RMTs for each workstation
$m = 6$	Maximum value of RMTs in system
$x^* = [25, 22, 25, 16]^\top$	Planned work in process (WIP)

The production rates of machines are artificial numbers, but ensure  $r^{RMT} < r^{DMT}$ . Since



**Figure 7.1.:** Job shop manufacturing systems with RMTs

WIP items are bounded capital, manufacturers aim to keep them working at full speed by keeping WIP on a balanced level, especially in case of disturbances. To this end, the order release rate for workstation 1 is superimposed with a sin function with amplitude of 30 % and a period of 10 hours for the approximation of volatile customer demand [26]. The flow matrix  $P$  is described in the Figure 7.1 with external input rates  $d(n) := [i_{01}(n) \ i_{02}(n) \ 0 \ 0]^T$ , where  $i_{02} = 6$ ,

$$i_{01}(n) = \begin{cases} 10 + 3|\sin(0.1\pi n)|, & n \in (20 \ 30] \\ 10, & \text{else} \end{cases} \quad (7.1)$$

The stage cost is given in (6.3) and the penalty weighting coefficient  $\lambda$  is equal to 0.1.

### 7.1.1. COMPARISON BETWEEN STANDARD PID AND MPC

Given the results from Proposition 6.3 for MPC and implementation for PID [196], both are readily applicable for job shop systems with RMTs. Upon implementation, using MPC one does not have to worry about closed-loop stability along with the choose of prediction horizon length, which shows usability without expert knowledge. In contrast to stability, PID requires internal knowledge of the key performance indicators applied to evaluate the feedback as well as good command of how to appropriately adapt the PID parameters to perform well for these indexes. Manually tuning these parameters to achieve a good performance in the coupled MIMO system may be complex. To cope with it, one can apply optimization methods, e.g., PSO [221] or iterated linear matrix inequalities [222]. For MPC, no further knowledge are required as KPIs are directly used as cost criterion, and are optimized by design. As PID cannot handle the constraints on the total number of RMTs, we employed a truncation function

$$u_j(n) = \begin{cases} u_j(n), & \text{if } \sum_{j=1}^p u_j(n) \leq m \\ \frac{m}{\sum_{j=1}^p u_j(n)} u_j(n), & \text{else} \end{cases} \quad (7.2)$$

such that PID always adheres to this constraint. Through extensive simulations to reduce oscillations as much as possible, the parameters of the PID controller were tuned manually and set to  $k_p = 0.5$ ,  $k_i = 0.01$  and  $k_d = 0$ .

In order to increase the basin of attraction, we chose  $N = 16$  for terminal endpoint constraints MPC and compared the results with PID. The simulation results are sketched in Figure 7.2. As expected, in Figure. 7.2 we observe that the proposed method is capable of tracking the desired WIP value for each workstation. The allocation of RMTs is displayed in Figure. 7.3. From Figure. 7.3, we observe that for both MPC and PID, the number of RMTs

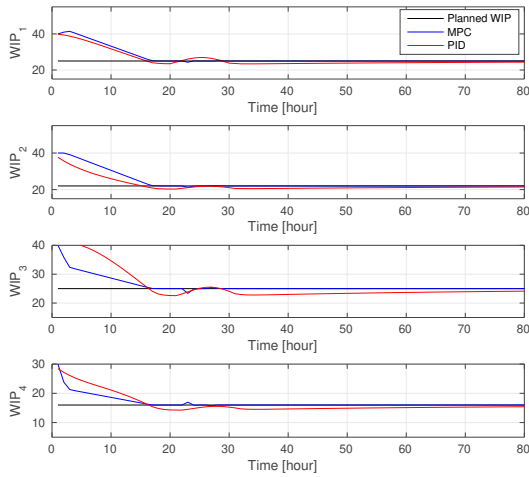
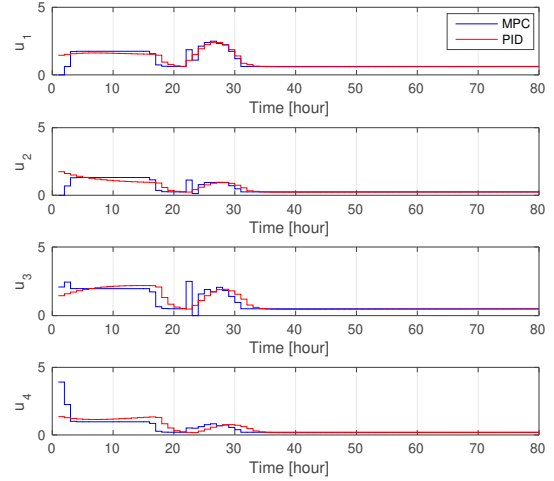
Figure 7.2.: Variation of  $WIP$ 

Figure 7.3.: Variation of usage of RMTs

assigned to particular workstation is identical from time instant  $n = 34$  onwards. Comparing this to the WIP levels in Figure 7.2, we find that MPC is tracking the desired values  $x_j^*$  nicely for each workstation  $j = 1, \dots, 4$ . PID, on the other hand, shows an offset, which we were unable to solve despite extensive tuning of the control parameters. For  $n \in [0, 33]$  in Figure 7.3, we observe that both controllers result in a very different assignment of RMTs, which however is not well reflected in standard comparison errors. Here, we considered integral absolute error (IAE), integral square error (ISE), integral absolute control (IAU) and integral square control (ISU) to analyze the results displayed in Figure 7.2 and 7.3, cf. Table 7.2 for details.

Table 7.2.: Comparison between standard PID and MPC with different performance indicators

		IAE	IAU	ISE	ISU
PID	WS1	199.40	77.41	1496.40	97.02
	WS2	161.06	40.25	1032.42	37.80
	WS3	254.92	74.51	2245.99	104.74
	WS4	162.59	37.10	866.20	31.85
MPC	WS1	155.24	76.83	1882.57	100.05
	WS2	165.45	39.73	2163.74	33.81
	WS3	83.10	73.47	622.92	103.40
	WS4	62.36	36.38	400.50	39.38

Considering the differences in the assignments, time instants  $n = 0$  and  $n = 22$  are of particular interest: At  $n = 0$ , PID chooses almost identical assignments of RMTs for all workstations, whereas MPC moves RMTs to workstations 3 and 4 only. As a result, the WIP levels for workstations 3 and 4 in the MPC case drop significantly faster than for PID, whereas WIP levels for workstations 1 and 2 are only slightly higher, cf. Figure 7.2. At



$n = 22$ , both PID and MPC reacted to the change of the input rate. The following curve is almost identical for both controllers and approximates a sin function, which is to be expected given the nature of the input rate change. In contrast to PID, however, MPC starts with a very strong peak in workstations 1, 2 and 3.

Based on these results, we conjecture that the difference between PID and MPC seen in Figures. 7.2 and 7.3 at  $n = 0$  could be reduced significantly by choosing different PID control parameters for each workstation. As this is unnecessary for MPC, we conclude that from a practitioners point of view MPC is more easily accessible. Regarding  $n = 22$  and the deviation from the desired values observed for PID, MPC – at least for this example setting – also shows improved performance. Additionally, we conjecture that this plug and play property will allow us to straight forward integrate the integer constraints and reconfiguration delays, whereas for PID these may cause serious problems in stability.

## 7.1.2. SENSITIVITY ANALYSIS

### 7.1.2.1. INITIAL CONDITIONS

To test our proposed method, we additionally simulated and implemented different initial conditions. In the Figure 7.5, the results are illustrated exemplarily for the WIP trajectory of workstation 1. In Figure 7.5, the black lines represent using an disturbance  $d_1 := [10 \ 10 \ 0 \ 0]^\top$ , which cannot be stabilized due to capacity limitations. The red lines stands for  $d_2 := [8 \ 10 \ 0 \ 0]^\top$  and the blue line represents  $d_3 := [10 \ 6 \ 0 \ 0]^\top$ . In both cases, our proposed method is able to stabilize the system. In Figure 7.4, we see that the method is capable to deal with different initial values, here 1)  $x(0) = [30 \ 30 \ 30 \ 20]^\top$ , 2)  $x(0) = [40 \ 30 \ 20 \ 30]^\top$ , 3)  $x(0) = [40 \ 40 \ 40 \ 30]^\top$ , and 4)  $x(0) = [50 \ 40 \ 50 \ 40]^\top$ , respectively, and stabilize the system accordingly.

#### Remark 7.1

*Note that the additional terminal condition  $x(N) = x^*$  has to be controllable and reachable in finite time. In the context of planning and control of manufacturing processes, it is crucial to identify the admissible set of the external input rate  $d$  which may include disturbance for manufacturers. Based on equation (6.3),  $u^* \in \mathbb{U}$  must be attained within the constraints. That means the determination of  $d$  must be resulting in a feasible solution  $u$  with the fixed given system parameter settings. If  $d$  is contained in a region for which a feasible solution exists, then any initial condition trends to a stable solution. This provides a clear understanding for manufacturers that how much the external input rate they can deal with under the limited available resource to achieve the asymptotic stability.*

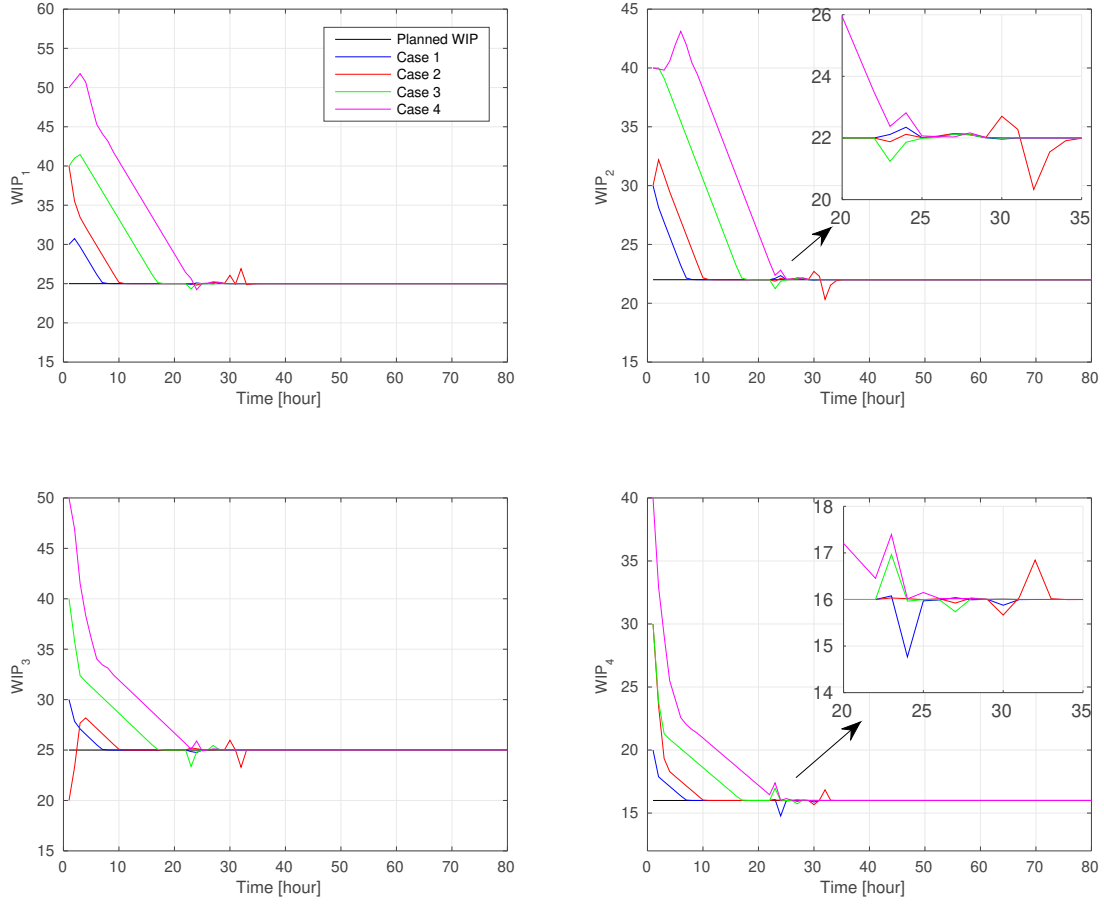


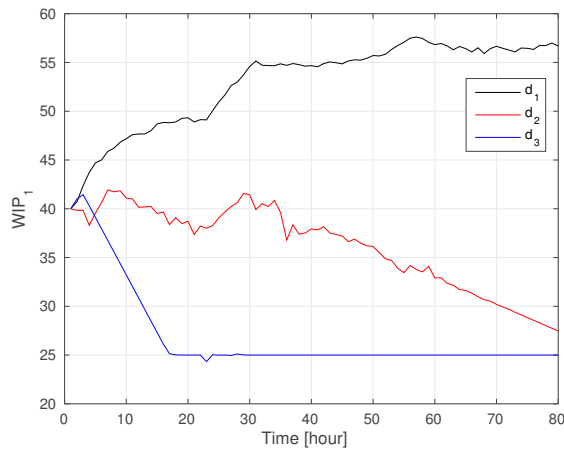
Figure 7.4.: WIP levels for different initial conditions

### 7.1.2.2. DYNAMIC FLOW CONTROL

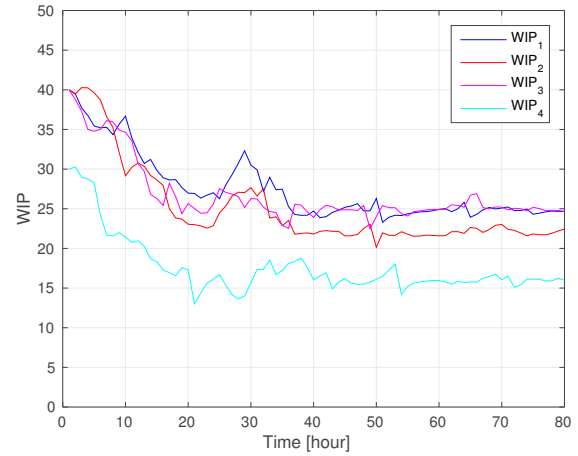
Further, we consider the case of a dynamic flow from [2] which includes the product mix handled by the workstations and which leads to the modification of off-diagonal values of  $P$  given by (7.3), e.g.,

$$\begin{aligned}
 p_{21}^2(n, O_1(u(n)), d(n)) &:= \frac{i_{02}^2(n)}{p_{12}^3 O_1(n) + i_{02}^2(n)} \\
 p_{12}^3(n, O_2(u(n)), d(n)) &:= \frac{i_{01}^\gamma(n)}{p_{21}^2 O_2(n) + i_{01}^\gamma(n)} \quad \gamma \in \{1, 3\} \\
 p_{23}^3(n, O_1(u(n)), d(n)) &:= \frac{p_{12}^3 O_1(n)}{p_{12}^3 O_1(n) + i_{02}^2(n)} \\
 p_{13}^2(n, O_2(u(n)), d(n)) &:= \frac{p_{21}^2 O_2(n)}{p_{21}^2 O_2(n) + i_{01}^\gamma(n)} \\
 p_{13}^1(n, O_2(u(n)), d(n)) &:= \frac{i_{01}^\gamma(n)}{p_{21}^2 O_2(n) + i_{01}^\gamma(n)} \quad \gamma \in \{1, 3\} \\
 p_{34}^3(n, O_2(u(n)), d(n)) &:= \frac{p_{23}^3 O_2(n)}{p_{23}^3 O_2(n) + (p_{13}^1 + p_{13}^2) O_1(n)}
 \end{aligned}$$

(7.3)



**Figure 7.5.:** Variation of  $WIP_1$  under different order release rate



**Figure 7.6.:** Variations of  $WIP$  in the dynamic flow

which are determined by the respective output rate and external input rate.

**Table 7.3.:** Process flow of each product [2]

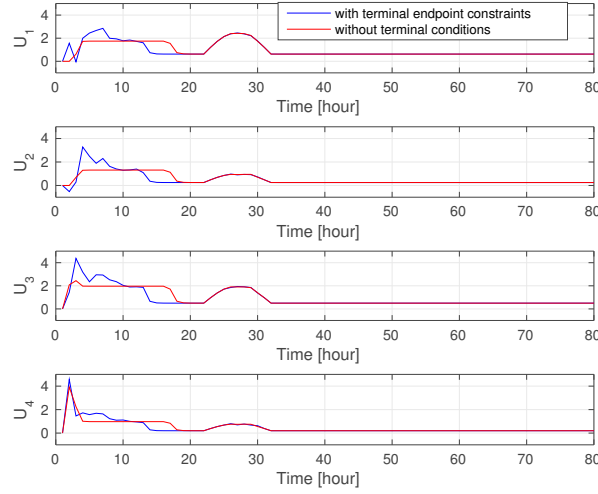
Product type	Arrival rate	Workstation sequence				Probability of Product flow
1	$i_{01}^1$	1	3	-	-	$p_{13}^1 \rightarrow p_{30}^1$
2	$i_{02}^2$	2	1	3	-	$p_{21}^2 \rightarrow p_{13}^2 \rightarrow p_{30}^2$
3	$i_{01}^3$	1	2	3	4	$p_{12}^3 \rightarrow p_{23}^3 \rightarrow p_{34}^3 \rightarrow p_{40}^3$

Respective results are illustrated in Figure 7.6. Here, we observe that – despite availability of all information regarding workstation outputs – the resulting trajectories show oscillations. Therefore, the closed-loop system is not asymptotically stabilized but showed practical asymptotic stability for the chosen prediction horizon  $N = 16$  only. For a complete stability proof in the case of time varying dynamics, also time varying Lyapunov arguments would be required, which is out of the scope of this thesis. Yet the simulation indicates that the applicability of MPC in the time varying case is not perfectly straight forward and requires further analysis.

### 7.1.2.3. PREDICTION HORIZON

Using the terminal endpoint constraints method is easy to be implemented, yet, one disadvantage is that typically a large prediction horizon is required to guarantee a large feasible set. This is fine with continuous variables which shows less computational requirement. Yet, for discrete variables, enormous numerical difficulties will arise with large  $N$  for stability and feasibility. To this end, we additionally compare terminal endpoint constraints MPC with unconstrained MPC scheme, cf., Figure 7.7 in which the prediction horizon is

chosen as  $N = 2$ . We observe that the former has a larger control effort and therefore may



**Figure 7.7.:** Comparison between with and without terminal endpoint constraints

suffer the feasibility with short horizon (e.g.  $u_2(2)$  is negative). In order to ensure feasibility using terminal endpoint constraints, we increase the basin of attraction and set  $N = 16$ , which was previously applied for comparison with PID. The control effort obtained without terminal conditions is more cautious and slow than the former.

Now, considering the case study with parameters setting presented in Table 7.1 and Figure 7.1. Assume the derive feedback law  $u(n) = \Phi_1(x(n) - x^*) + u^* \in \mathbb{U}$  (i.e., (6.13)) holds, based on the asymptotical controllability condition (6.16), we have  $\|\Phi_1\|_2 = 1.0800$ ,  $C := 1 + \lambda\|\Phi_1\|_2^2 = 1.11664 > 1$  and  $0 < \sigma := \|(\mathbf{Id} + r^{RMT} \cdot P \cdot \Phi_1)\|_2 = 0.01817104 < 1$ . Based on the computation of  $\alpha_N$  via (3.47), asymptotic stability is guaranteed and we obtain the performance index  $\alpha_2 = 1 - (\gamma_2 - 1)^2 = 0.98125004$ , where  $\gamma_2 = 1.13693051$ .

The above applies the asymptotic controllability with respect to running costs to show stability. Now, we additionally check the controllability with respect to optimal value function (3.31) with  $N = 2$ . Then, we have  $\ell(x, \kappa_2(x)) = \|x - x^*\|_2^2 + \lambda\|\Phi_1(x - x^*)\|_2^2$ ,  $V_1(x) = \|x - x^*\|_2^2$ ,  $V_1(f(x, \kappa_2(x))) = \|(\mathbf{Id} + r^{RMT} P \Phi_1)(x - x^*)\|_2^2$ , and

$$V_2(x) = \ell(x, \kappa_2(x)) + V_1(f(x, \kappa_2(x))) \leq (1 + \overbrace{\lambda\|\Phi_1\|_2^2 + \|\mathbf{Id} + r^{RMT} P \Phi_1\|_2^2}^{\gamma})\ell(x, u) \quad (7.4)$$

which shows that Assumption 3.31 holds, cf., [Remark 4.3 in [156]]. Then,  $\alpha = \frac{(\gamma+1)^{N-2} - \gamma^N}{(\gamma+1)^{N-2}} = 0.98182598$ , where  $0 < \gamma = \sigma + C - 1 = 0.13481104 < 1$ . Here,  $\alpha$  is computed slightly different compared to the former  $\alpha_2$ . But for both cases, we can guarantee  $\alpha \in (0, 1]$  and hence forth stability. Also, the estimate between the infinite horizon feedback controller is

given by

$$\frac{V_{\infty}^{\kappa_2}(x) - V_{\infty}(x)}{V_{\infty}(x)} \leq \frac{\gamma^N}{(\gamma + 1)^{N-2} - \gamma^N} = 0.01851043 \quad (7.5)$$

Yet, constraints violation occur, e.g. at initial high WIP level

$$\begin{pmatrix} \overbrace{\begin{matrix} 0.5671 & 0.2569 & -0.0231 & -0.0078 \\ 0.1994 & 0.5686 & -0.0216 & -0.0073 \\ 0.4276 & 0.4350 & 0.4662 & -0.0112 \\ 0.1669 & 0.1663 & 0.1803 & 0.4828 \end{matrix}}^{\Phi_1} \\ \times \\ \begin{pmatrix} \overbrace{\begin{matrix} 40 - 25 \\ 40 - 22 \\ 40 - 25 \\ 30 - 16 \end{matrix}}^{x(0)-x^*} \\ + \\ \begin{pmatrix} \overbrace{0.625}^{u^*} \\ 0.25 \\ 0.5 \\ 0.2 \end{pmatrix} \\ = \\ \begin{pmatrix} \overbrace{13.3000}^{u \notin \mathbb{U}} \\ 13.0496 \\ 21.5802 \\ 15.1606 \end{pmatrix} \end{pmatrix}$$

Nevertheless, we have shown that stability still can be guaranteed via (6.17) with  $N = 2$ . Now, we set  $\Phi_2 := \Phi = -0.05 \cdot a_1^{-1} \cdot \mathbf{Id}$ ,  $\|\Phi_2\|_2 = 0.0624$ . Then, this control law is feasible whiling considering the worst case at initial high WIP level and we have

$$\begin{pmatrix} \overbrace{\begin{matrix} 0.0312 & 0.0156 & 0 & 0 \\ 0.0125 & 0.0312 & 0 & 0 \\ 0.0250 & 0.0250 & 0.0250 & 0 \\ 0.0100 & 0.0100 & 0.0100 & 0.0250 \end{matrix}}^{\Phi_2} \\ \times \\ \begin{pmatrix} \overbrace{\begin{matrix} 40 - 25 \\ 40 - 22 \\ 40 - 25 \\ 30 - 16 \end{matrix}}^{x(0)-x^*} \\ + \\ \begin{pmatrix} \overbrace{0.625}^{u^*} \\ 0.25 \\ 0.5 \\ 0.2 \end{pmatrix} \\ = \\ \begin{pmatrix} \overbrace{1.3750}^{u \in \mathbb{U}} \\ 1.0000 \\ 1.7000 \\ 1.0300 \end{pmatrix} \end{pmatrix}$$

Then we get  $0 < \sigma := \|\mathbf{Id} + a_1 \cdot \Phi_2\|_2^2 = 0.9025 < 1$ , choose  $\lambda = 0.1$ , then  $C = 1 + \lambda \|\Phi_2\|_2^2 = 1.00039$ ,  $\gamma_2 = C(1 + \sigma) = 1.9032$  and  $\alpha_2 = 1 - (\gamma_2 - 1)^2 = 0.1842$ , which is much smaller than the former with  $\alpha_2 = 0.98182598$ . This is also reflected in Table 7.5, which shows  $\alpha_2$  at the initial high WIP stage with strong constraint limitations is smaller than the latter ones (e.g. around the practical stability region). The reason is attributed to  $\Phi_1$  such that  $u(\Phi_1(x)) \in \mathbb{U}$  holds. Then,  $\alpha_2$  is directly derived from  $\Phi_1$  with a larger estimate. From another point of view, the estimate for all  $x \in \mathbb{X}$  is too conservative and (7.5) cannot be accurately estimated. Later, we will quantitatively estimate degree of suboptimality online, which is particularly conducive to the design of adaptive horizon NMPC.

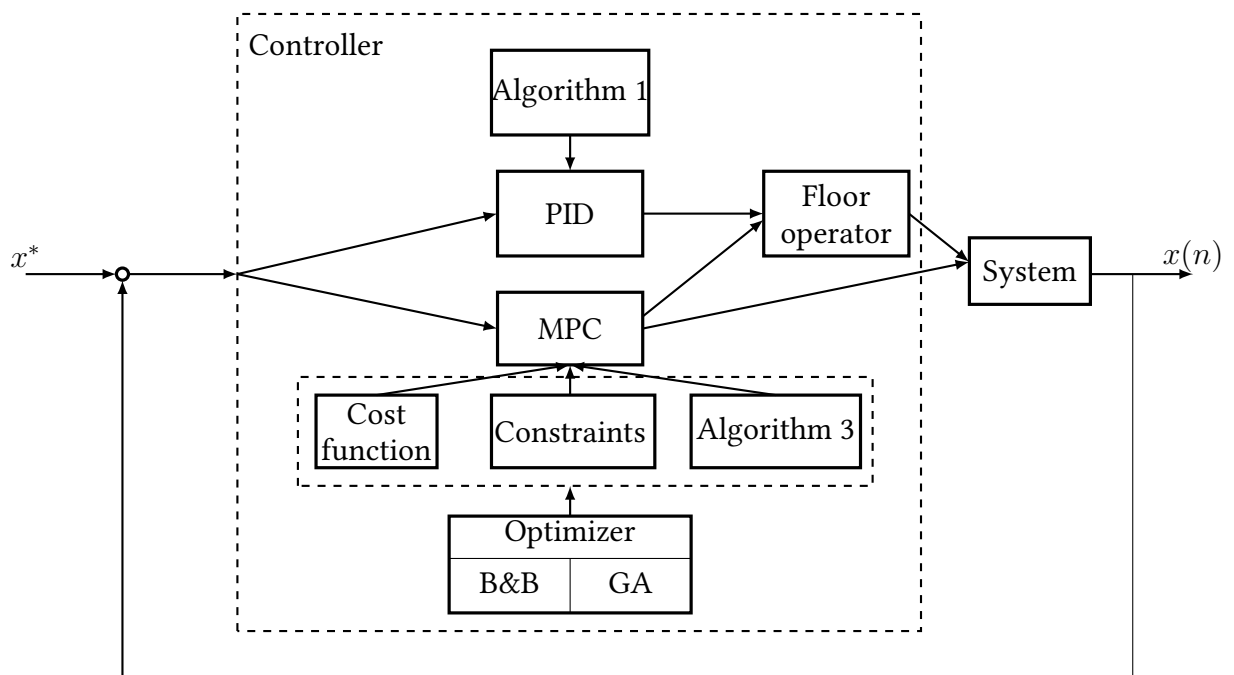
### Remark 7.2

*While the solution derived from optimization typically outperforms the decision based on worker experience, the computational cost grows with the dimension of the system and may be intractable, i.e. the best solution may not found in a reasonable time. Within a job shop*

system, this may be the case if the initial state of the job shop system is far from the desired equilibrium. In this case, decentralized or distributed control could be as alternatives applied for the high order system [223, 224, 225], which are out of the scope of this thesis.

## 7.2. PRACTICAL STABILITY WITH INTEGER CONSTRAINTS AND RECONFIGURATION DELAYS

Based on Section 7.1 , we extend the research and consider a NP-hard optimization problem for the assignment of RMTs while considering reconfiguration delay to maintain WIP for each workstation. The closed-loop predictive control system including integer operators is built, cf. Figure 7.8 for a respective block diagram. The description concerning elements (e.g. GA, Algorithm 1, etc) were introduced in Chapter 5. The parameter setting concerning stage cost and system dynamics are identical to the former defined in Section 7.1 but we shift the demand fluctuation period to [35, 45] for a better representation that such the fluctuation is occurring at the steady state stage.



**Figure 7.8.:** Different control strategies for controlling *WIP* by integer assignment of RMTs

### 7.2.1. COMPARISON OF DIFFERENT INTEGER CONTROL STRATEGIES

According to the analysis concerning stage cost and stability presented in Remark 6.5, we change the stage cost as

$$\ell(x(k), u(k)) = \|x(k) - x^*\|_2^2 + 0.1 \cdot \|u(k)\|_2^2 \quad (7.6)$$

and keep other system settings (e.g.,  $P$ ) unchanged. We consider the shortest possible prediction horizon  $N = 2$ , simulate the process using MATLAB and obtain the results shown in Table 7.4. Here,  $\text{MPC}_1$ ,  $\text{MPC}_2$  and  $\text{MPC}_3$  represent MPC with floor operator, branch and bound and genetic algorithm, respectively. To assess the performance of the controllers, we employ the mean value of WIP  $\mu(WIP)$  and its standard deviation  $\sigma(WIP)$  as well as the mean value of utilized RMTs in the system  $\mu(\text{RMT})$ . As benchmark, we additionally implement a PID controller together with the floor operator, cf. [226, 227] for details. As  $N = 2$  according to the literature [107] and also in our simulations reveals

**Table 7.4.:** Comparison between standard PID and different integer control strategies based on MPC

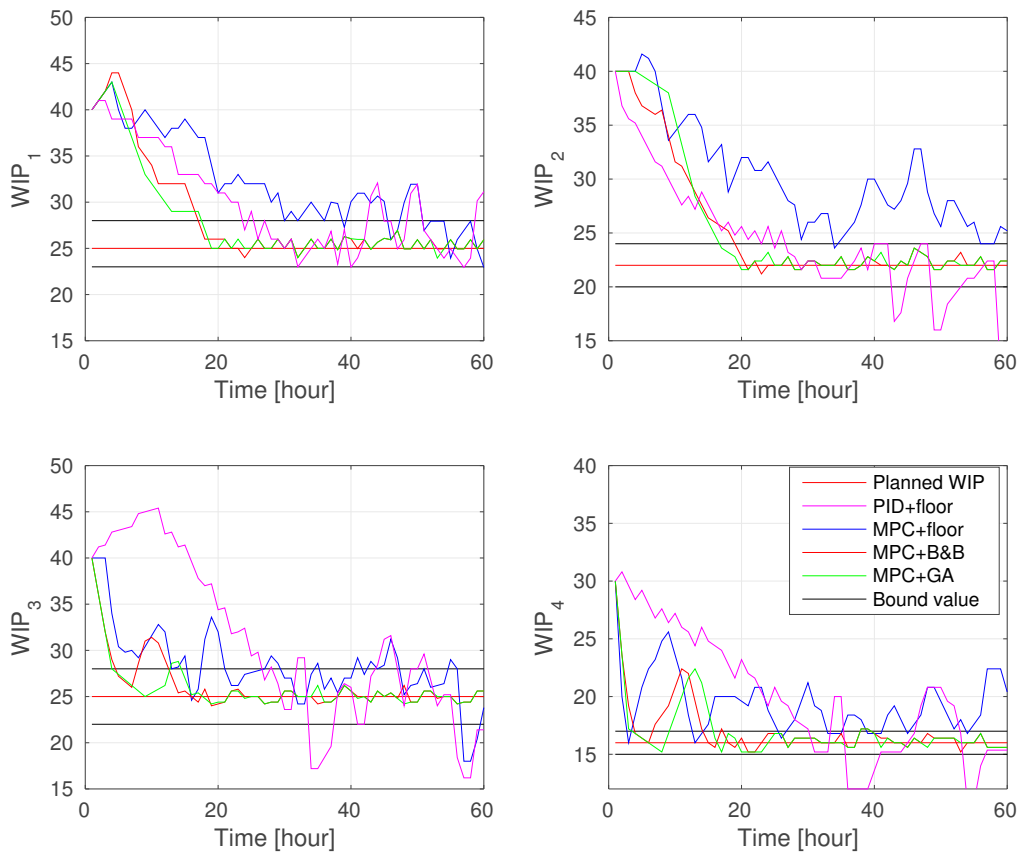
$N = 2$		PID	$\text{MPC}_1$	$\text{MPC}_2$	$\text{MPC}_3$	$\text{MPC}_{u^*(0)}^2$
$\mu(WIP)$	WIP <sub>1</sub>	29.8871	32.2314	28.4981	28.0647	28.5314
	WIP <sub>2</sub>	24.2600	30.3400	25.3933	25.6733	25.3800
	WIP <sub>3</sub>	30.8567	28.2767	26.1567	25.8767	26.1367
	WIP <sub>4</sub>	19.9573	19.3600	17.0467	16.8867	16.8800
$\sigma(WIP)$	WIP <sub>1</sub>	5.3645	5.0129	5.6422	5.0149	5.6781
	WIP <sub>2</sub>	5.5420	4.9867	5.6820	6.3205	5.6762
	WIP <sub>3</sub>	8.7065	4.0782	2.9088	2.6326	2.7466
	WIP <sub>4</sub>	5.4342	2.6071	2.4226	2.4222	2.3599
$\mu(\text{RMT})$	WS <sub>1</sub>	1.4500	1.0492	1.0492	1.0667	1.0667
	WS <sub>2</sub>	0.9833	0.5738	0.5574	0.5667	0.5667
	WS <sub>3</sub>	1.4333	1.0328	1.0328	1.0500	1.0500
	WS <sub>4</sub>	0.7333	0.5246	0.5246	0.5333	0.5333
CPU time (s)		1.274	24.857	1031.312	503.020	328.343

the worst performance for MPC, we limited ourselves to this case. Note that choosing a larger prediction horizon  $N$  typically improves the performance of the controller, yet from Table 7.4 we observe that MPC with even only  $N = 2$  for any choice of integer optimization method outperforms PID.

From Table 7.4 we additionally observe that the results concerning the key performance indicators  $\mu(WIP)$ ,  $\sigma(WIP)$  and  $\mu(\text{RMT})$  are almost identical if branch and bound or a genetic algorithm is applied. Yet, we observe a significant improvement of both approaches as compared to the floor operator concerning the mean WIP level  $\mu(WIP)$  and its first standard deviation  $\sigma(WIP)$ , whereas the mean number of utilized RMTs  $\mu(\text{RMT})$  almost

remain at the same level as compared to MPC with floor operator. However, using PID controller requires a higher control effort to maintain the WIP level.

On the downside, however, the computation time employing B&B or GA is drastically increased. Moreover, the computational complexity especially of B&B would grow exponentially if the prediction horizon is prolonged. We like to note that the computational time can be significantly reduced if only  $u^*(0)$  is considered as an integer optimization variable, cf., Section 5.1.3. To this end, we additionally simulate this case and present the results at Table 7.4 called  $\text{MPC}_2^2$ . We observe that the number of utilized RMTs is a little higher than  $\text{MPC}_2$ , the performance is also acceptable. In Figure 7.9, we include the closed loop trajec



**Figure 7.9.:** Comparison of *WIP* level in each workstation by different methods with  $N = 2$

Here, we observe that the closed loop trajectories for MPC with B&B and with GA coincide almost perfectly and converge towards a practical stability region near the planned WIP level restricted in the scope of one unit of  $r^{RMT}$ , cf. Figure 7.10. Also, we observe that the tracking error of workstation 3 and workstation 4 are locally increasing. The convergence of  $x_{\{3,4\}}(n) \rightarrow x_{\{3,4\}}^*$  may be non-monotone. Yet,  $V_N$  is ensured to be strictly monotonically decreasing at those time instants, cf. Figure 7.11 with red markers. Note that  $V_N$  may be also locally increasing under reconfiguration delay with different system dynamics, see Section 7.2.4.3 for detailed discussion. We like to note that due to the iterative nature of the MPC algorithm, the violation of integer control constraints is healed implic-



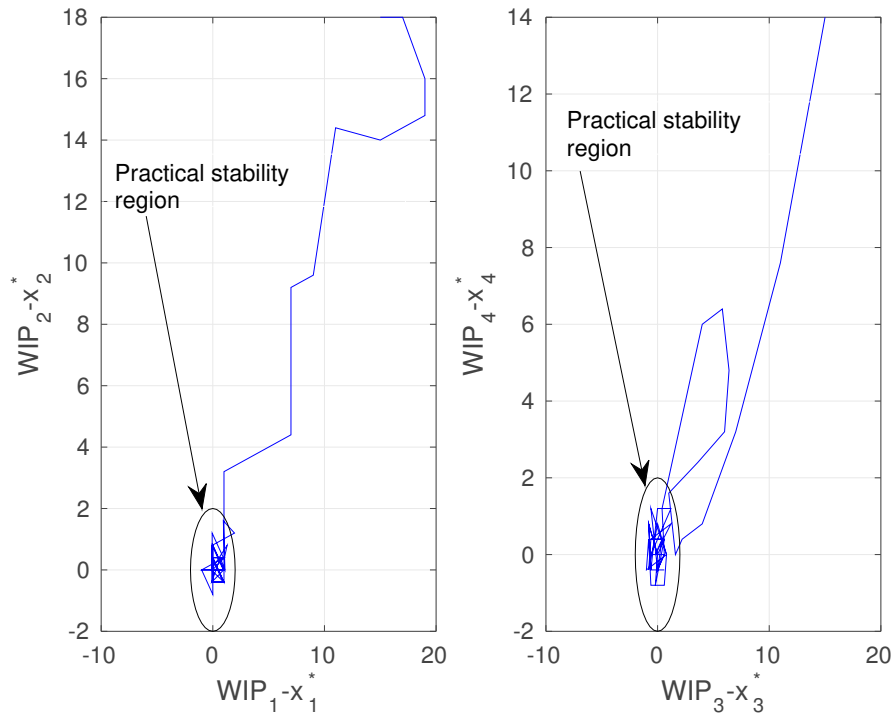


Figure 7.10.: Difference between current  $WIP$  value with planned value for each workstation

ity. Yet, standard techniques for showing stability [107] may not apply as each solution may be infeasible. Also, asymptotic stability may not be reached, cf. [107] for details on the stability properties.

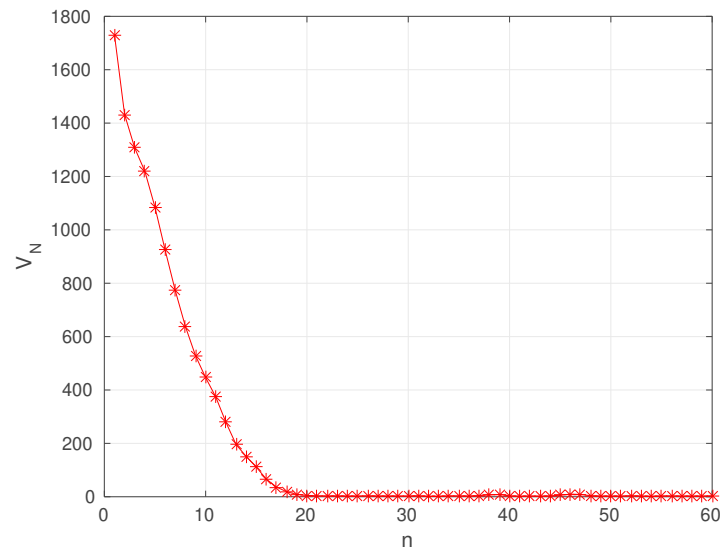
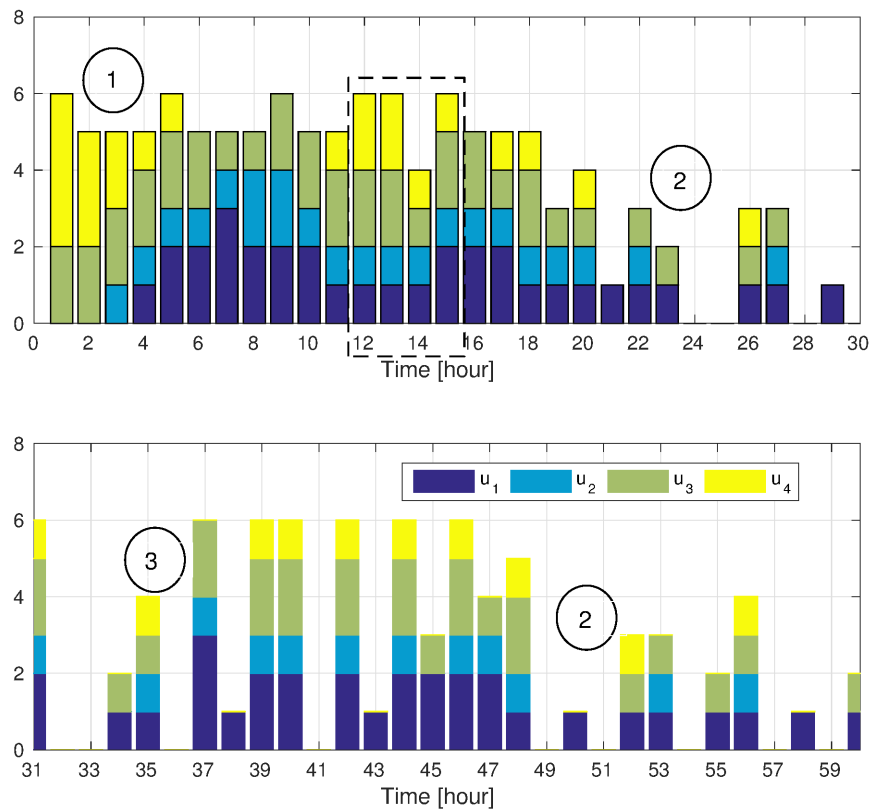


Figure 7.11.: Optimal value function for  $N = 2$

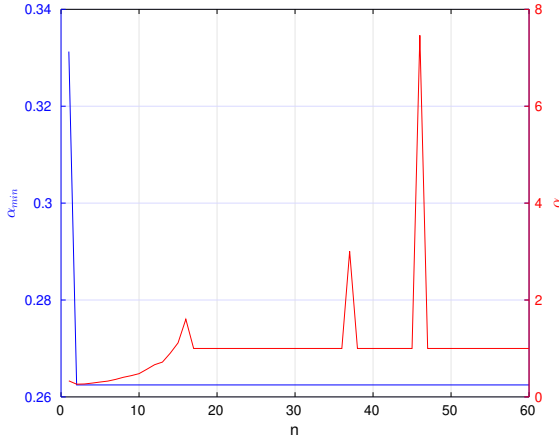
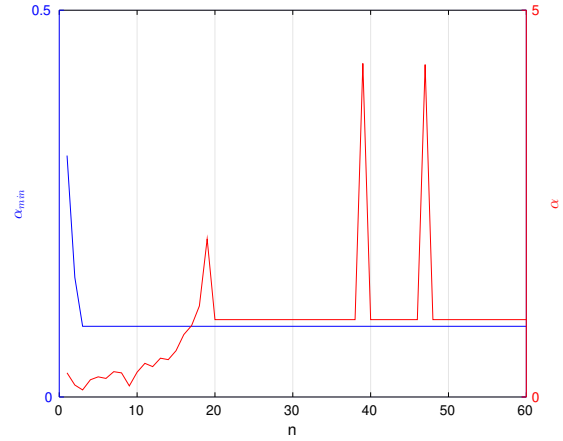
In Figure 7.9, we additionally observe that the settling time for PID or MPC in combination with the floor operator is significantly larger compared to optimization based integer operators. Based on this observation, we conjecture that the usage of the floor operator is not applicable to frequent reconfiguration requirements and, if the frequency becomes too high, it may actually destabilize the system.

To support the latter, Figure 7.12 displays the usage of RMTs in the workstations by applying MPC together with B&B. Here, the time period is divided into three parts, settling time period ①, stable period ② and fluctuation period ③. We observe that after the settling time only a few RMTs are required within the workstations, which are alternately assigned and reassigned. Also, we can observe that the number of utilized RMTs increases due to demand fluctuation at time period  $n \in [35, 45]$  ③. After a short time, the system is practically stable again. The variations of employed RMTs correspond with the changes in WIP controlled by MPC with B&B (i.e., MPC-B&B). We like to note that if the RMTs are fully exploited at both of the adjacent time instants, the RMTs of operating conditions must be identical. Also, if the number of previously utilized RMTs is equal to the maximum  $m$ , then there are no idle RMTs left to be assigned. From Figure 7.12, we observe that one RMT previously employed in the workstation 4 is assigned to workstation 1 with reconfiguration delay. This behavior reflects and satisfies the real constraints, which are addressed in the Algorithm 3.



**Figure 7.12.:** Number of RMTs by B&B associated with MPC for each workstation

Combined, our case study indicates that applying MPC with any type of integer control strategy is superior to PID. Amongst the considered alternatives, B&B as well as GA show better and almost identical performance regarding the considered indexes. Hence, due to computational requirements, the combination of genetic algorithm and MPC appear to be a promising control alternative. Note that due to the non-optimal solution, the practical stability region derived by MPC with GA is typically larger than MPC-B&B.

Figure 7.13.: Variation of  $\alpha$  without delayFigure 7.14.: Variation of  $\alpha$  with delay

## 7.2.2. PRACTICAL STABILITY VIA AN A POSTERIOR SUBOPTIMALITY

### ESTIMATE

To show practical stability for the closed-loop trajectory controlled MPC-B&B with  $N = 2$ , cf., Figure 7.9, we follow the method proposed in [130] to compute an online estimate the degree of suboptimality for implicitly guaranteeing practical stability. Considering the initial conditions and a possibly shorter prediction horizon, we use an a posterior estimate instead of an a priori estimate, see detailed discussion in Section 6.2.2 together with Figure 7.19. In order to analyze the influence caused by reconfiguration delays in terms of transition and steady-state performance, we first present the results in case of integer constraints but without delays. Here, we select two segments of time intervals – transition phase  $n \in [1, 20]$  and steady-state stage  $n \in [31, 50]$  which includes the demand fluctuation period  $[35, 45]$ , respectively. According to Algorithm 5 with the selected  $\varepsilon = 2.8$ , the results are given in Table 7.5. The marked values represent reaching into the practical stability region for the first time. Then it will stay there due to condition (6.31). The variations of local estimate  $\alpha$  and closed-loop estimate  $\alpha_{min}$  are illustrated in Figure 7.13. We observe that  $\alpha \in (0, 1]^5$  holds for all time instants, which implicitly ensures practical stability.

Similar to the estimates in Table 7.5, we further present the results with consideration of integer constraints and reconfiguration delay in Table 7.6. The marked values also stand for firstly reaching into the practical region, which becomes larger due to reconfiguration delay via observation on  $\ell(\cdot, \kappa_2(\cdot))$  and the truncate constant  $\varepsilon$  is determined as  $\varepsilon = 5.0$ . The dynamic variation of estimates is illustrated in Figure 7.14, which you can observe that  $\alpha \in (0, 1]$  is satisfied for the entire simulation period.

<sup>5</sup>Actually, due to the conservative estimate in case of practical stability,  $\alpha$  in the area of  $\ell(\cdot, \kappa_N(\cdot)) > \varepsilon$  may be larger than 1. In this case, we could simply set  $\alpha = 1$ . But here we prefer to keep the original value to distinguish those “real” 1 that entering into the practical stability region.

**Table 7.5.:** A posterior estimate of degree of suboptimality without reconfiguration delay in case of practical asymptotic stability

$n$	$\alpha$	$V_2(\cdot)$	$\ell(\cdot, \kappa_2(\cdot))$	$n$	$\alpha$	$V_2(\cdot)$	$\ell(\cdot, \kappa_2(\cdot))$
1	0.3313	1730.7600	972.0000	31	1.0000	3.6800	2.6800
2	0.2625	1409.6800	759.7600	32	1.0000	1.0000	1.0000
3	0.2668	1211.0000	651.1200	33	1.0000	1.0000	$1.8054 \times 10^{-15}$
4	0.2853	1039.0000	560.8800	34	1.0000	3.1200	1.1200
5	0.3068	878.8000	478.1200	35	1.0000	3.0000	2.3200
6	0.3264	732.9600	401.6800	36	1.0000	2.3600	0.6800
7	0.3614	602.7600	322.4800	37	3.0053	4.5416	3.0800
8	0.4052	483.6000	271.2800	38	1.0000	3.7001	2.4616
9	0.4385	374.8000	213.3200	39	1.0000	3.3804	1.9385
10	0.4800	282.4800	162.4800	40	1.0000	1.8028	1.6419
11	0.5706	205.8400	121.0000	41	1.0000	1.2630	0.1609
12	0.6647	138.4000	85.8400	42	1.0000	2.6326	2.1021
13	0.7187	83.2000	53.5600	43	1.0000	1.5874	0.6305
14	0.9023	46.7200	30.6400	44	1.0000	3.3009	1.9570
15	1.1148	21.6000	17.0800	45	1.0000	4.7074	2.7439
16	1.6029	5.6800	5.5200	46	7.4640	4.4894	2.9635
17	<b>1.0000</b>	<b>1.3200</b>	<b>0.1600</b>	47	1.0000	3.2694	1.8260
18	1.0000	2.8000	1.3600	48	1.0000	2.4269	1.7435
19	1.0000	3.0000	1.8400	49	1.0000	2.2494	0.6835
20	1.0000	3.3200	1.1600	50	1.0000	2.3494	1.6660

### 7.2.3. ANALYSIS OF RECONFIGURATION DELAYS ON CLOSED-LOOP PERFORMANCE

Now, the interesting question regarding the impact caused by reconfiguration delay on closed-loop performance including transient and steady-state arises. To this end, we compare the results concerning the usage of RMTs and WIP tracking for both cases, cf, Figures 7.15 and 7.16.

1 . In the transient phase  $n \in [0, 20]$ , we observe that the usage of RMTs with reconfiguration delays are lagged, cf., Figure 7.15. In contrast, without reconfiguration delay a faster convergence than the other is observed because the triggered RMTs can be utilized instantaneously without considering the reconfiguration logical limitation, cf., Figure 7.16. This is also reflected through out the results of suboptimality estimate in the Table 7.5 and 7.6, which shows that the estimates derived without delay are larger than the other for each time instant. Also, the former enters into the practical region (at step 17) faster than the latter (at step 20), cf., Remark 3.22 concerning the finite time to the practical stability region.

**Table 7.6.:** A posterior estimate of degree of suboptimality with reconfiguration delay in case of practical asymptotic stability

$n$	$\alpha$	$V_2(\cdot)$	$\ell(\cdot, \kappa_2(\cdot))$	$n$	$\alpha$	$V_2(\cdot)$	$\ell(\cdot, \kappa_2(\cdot))$
1	0.3121	1740.9600	971.8000	31	1.0000	3.6800	2.6800
2	0.1546	1439.2000	770.9600	32	1.0000	1.0000	1.0000
3	0.0914	1320.8000	669.2400	33	1.0000	1.0000	$2.8399 \times 10^{-27}$
4	0.2226	1260.1000	651.9600	34	1.0000	3.1200	1.2000
5	0.2604	1160.6000	609.0400	35	1.0000	3.0000	2.3200
6	0.2399	958.7400	508.0200	36	1.0000	2.3600	0.6800
7	0.3273	838.0800	451.3200	37	1.0000	4.5416	3.0800
8	0.3116	691.9800	388.1600	38	1.0000	6.7205	1.5616
9	0.1434	572.5800	304.1200	39	4.3126	8.4080	6.1590
10	0.3206	529.7000	268.6600	40	1.0000	3.4099	3.2491
11	0.4348	445.1600	262.2400	41	1.0000	1.2630	0.1609
12	0.3927	333.3000	183.6200	42	1.0000	2.6326	2.1021
13	0.5016	263.1600	150.3800	43	1.0000	1.5874	0.6305
14	0.4827	190.2400	113.6800	44	1.0000	3.3009	1.9570
15	0.5981	137.7800	77.2600	45	1.0000	5.6099	1.8439
16	0.8068	94.5600	61.5200	46	1.0000	10.0144	4.7660
17	0.9229	48.9600	34.0400	47	4.2973	7.7919	6.2485
18	1.1768	22.1600	15.5200	48	1.0000	2.4269	1.7435
19	2.0427	9.7800	7.3400	49	1.0000	2.2494	0.6835
20	<b>1.0000</b>	<b>5.0000</b>	<b>2.8400</b>	50	1.0000	2.3494	1.6660
21	1.0000	3.4200	2.2600	51	1.0000	2.2494	0.6835
22	1.0000	2.7400	1.4600	52	1.0000	3.9894	1.8660
23	1.0000	3.1200	1.4800	53	1.0000	2.4269	2.4235

2. In the steady-state phase (e.g.,  $n \in (20, 60]$ ), if there is no fluctuation with the required triggered reconfiguration (e.g.,  $n \in (20, 35]$ ), the usage of RMTs  $u(n)$  and the practical region would be identical. Yet, this is too optimistic and typically the expansion of the practical stability region with delay is larger than the other. The reason is attributed to the unavoidable possibly demand fluctuation on the steady-state stage (e.g.,  $n \in [35, 45]$ ). To cope with such disturbances, RMTs may need to be reassigned to keep / adapt to the original / new operational conditions (e.g., different desired WIP level), respectively. For either case, due to the weakened and lagged control efforts in the presence of reconfiguration delay, the practical region would be enlarged, cf., Figure 7.16. Also, the designed truncated constant  $\varepsilon$  has to be set larger (i.e., from  $\varepsilon = 2.8$  to  $\varepsilon = 5.0$ ) to guarantee  $\alpha \in (0, 1]$ . Otherwise, the estimate  $\alpha$  at some time instants may become negative with an inappropriate  $\xi$ , e.g.  $n = 37$ ,  $\alpha(37) = -7.7820$  in Figure 7.17. As a result, Algorithm 6.5 does not hold any longer. After

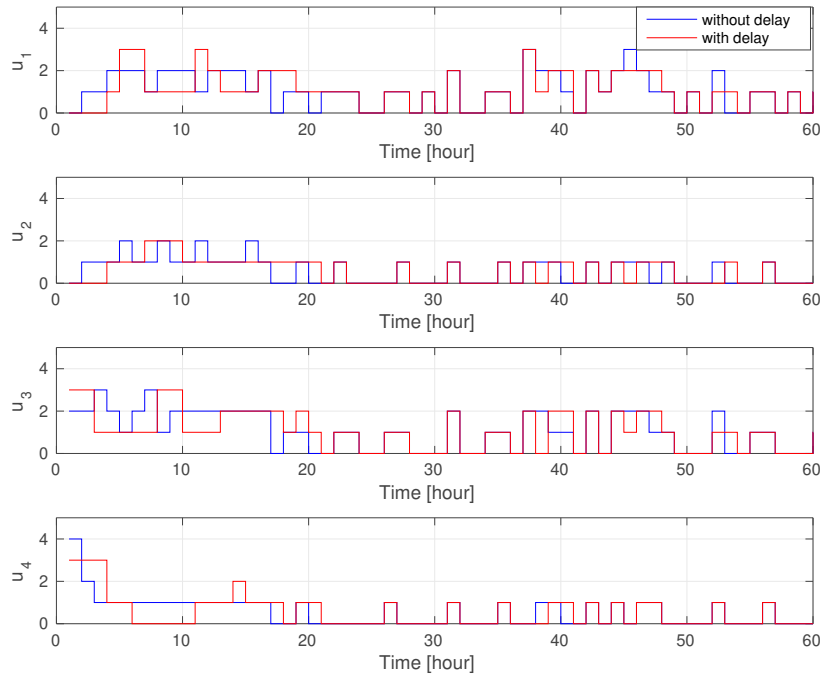


Figure 7.15.: Comparison of usage of RMTs with and without reconfiguration delay

the fluctuation, all WIP trajectories are maintained to the desired levels and the usage of RMTs would be same because at that time there are no more reconfigurations needed and RMTs are operating at their workstations without reassignment. We like to point out that at  $n = 32$  and  $n = 33$ , the usage of RMTs are equal to 0 for both cases, cf., Figure 7.15. In addition, the WIP level at  $n = 33$  is almost identical to the target value, cf., Figure 7.16. As a result, according to (7.6), the derived value of stage cost at  $n = 33$  is almost equal to 0, cf., Table 7.5 and 7.6.

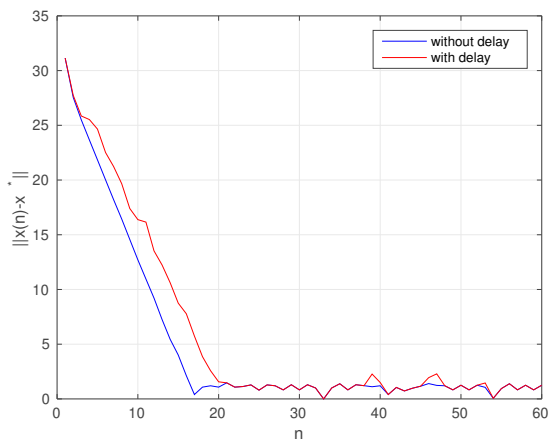


Figure 7.16.: Comparison of WIP tracking with and without delay

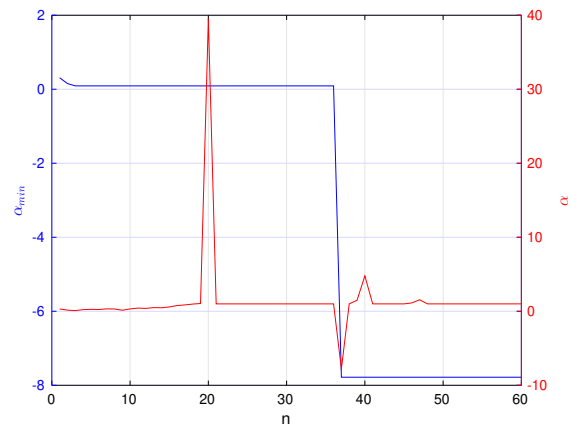


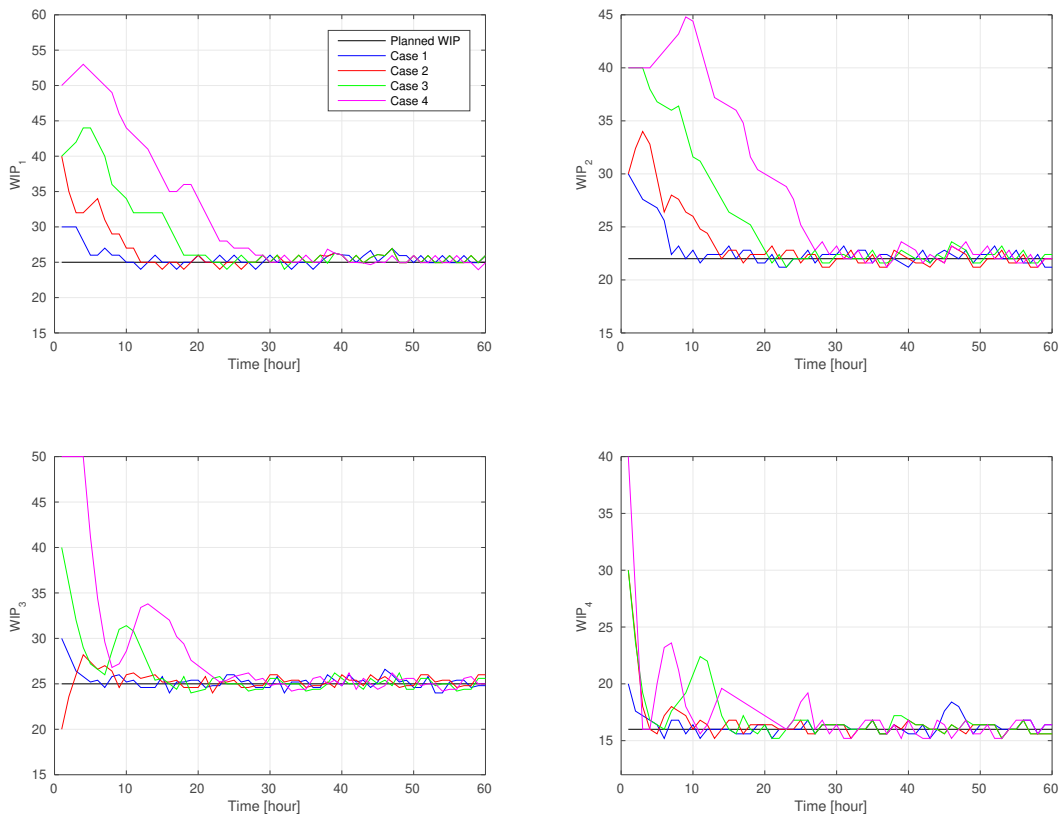
Figure 7.17.: Estimate under reconfiguration delays but with  $\varepsilon = 2.8$

### 7.2.4. SENSITIVITY ANALYSIS

Similar to the sensitivity analysis conducted in Section 7.1.2 in case of continuous variables, we consider it in the presence of integer constraints and reconfiguration delay.

#### 7.2.4.1. INITIAL CONDITIONS

Likewise, we choose the same initial conditions setting that defined in Section 7.1.2.1, i.e., 1)  $x(0) = [30 \ 30 \ 30 \ 20]^\top$ , 2)  $x(0) = [40 \ 30 \ 20 \ 30]^\top$ , 3)  $x(0) = [40 \ 40 \ 40 \ 30]^\top$ , and 4)  $x(0) = [50 \ 40 \ 50 \ 40]^\top$ , respectively. Unlike Figure 7.4, which showed that strict asymptotic stability can be achieved, Figure 7.18 shows the WIP trajectories can only be practically asymptotically stabilized with a slight oscillation around the target values. Actually, this derived results are expected since the perfect asymptotic stability is impossible under the



**Figure 7.18.:** *WIP levels for different initial conditions in practical case*

Through Table 7.6, we show that practical stability can be ensured via the less conservative a posteriori estimate under the initial conditions–case 3). Similarly, we can use the a posteriori estimate to study the property of practical stability for all other cases. Yet, here we do not go through the same procedure again for all initial conditions. Instead, based on the investigated case 3), we additionally consider case 1) in which the initial condition is not so far away to the planned level which may allow us to guarantee the stability with a short  $N$  in an a priori estimate way, cf., Remark 7.2. To this end, we choose  $N = 3$  to

compare the a posteriori estimate (blue line) and the a priori estimate (red line) for these two different initial conditions and show how conservative the latter is. In Figure 7.19, for case 1), we observe that for both estimates,  $\alpha \in (0, 1]$  can be satisfied. For case 3), however,  $\alpha$  is less than the former for each case and  $\alpha$  derived from using an a priori estimate even be negative at the high load period  $[0, 13]$  which requires a long  $N$  to guarantee the stability. Nevertheless, along with the variation of system dynamics (i.e., the controlled trajectory is not far away to the desired level), then using an adaptive NMPC technique naturally allows to switch to a short  $N$  based on a stability suboptimality estimate for saving computational cost, cf., [220].

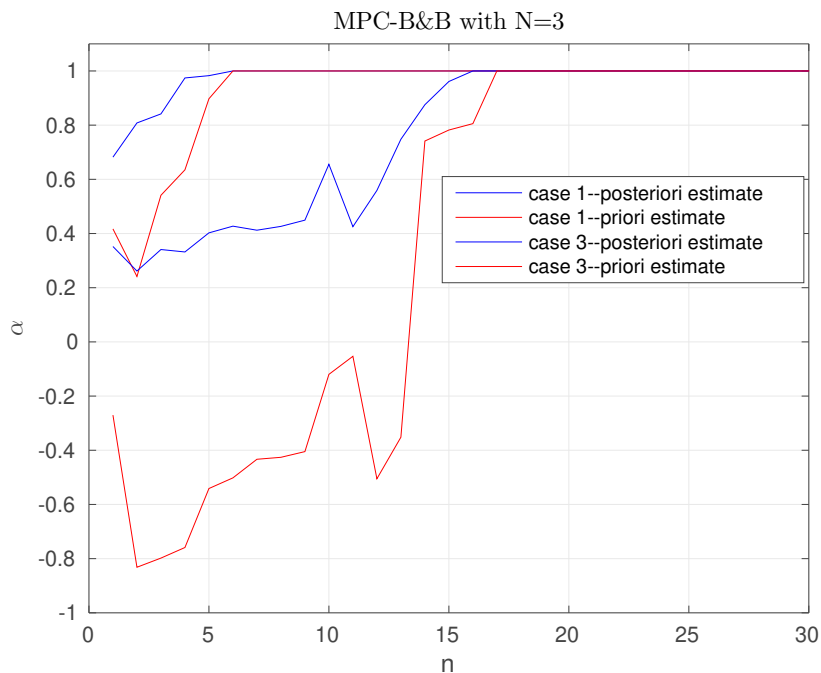
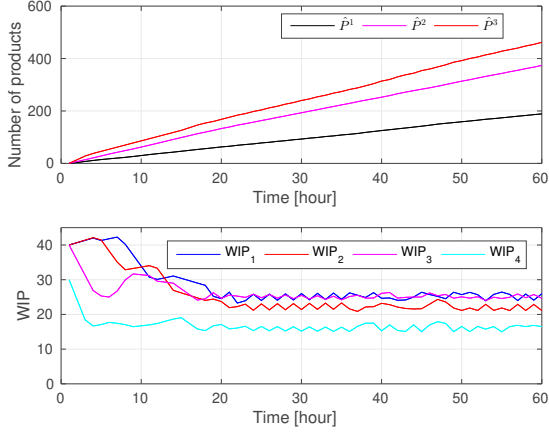


Figure 7.19.: Variation of  $\alpha$  for different situations

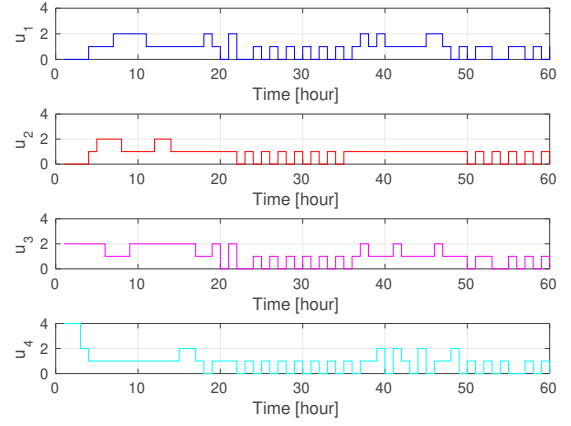
#### 7.2.4.2. DYNAMIC FLOW CONTROL

In Section 7.1.2.2, we studied the dynamic flow control with continuous variables, and the results indicated that the closed-loop trajectory shows a slight oscillation under the time-varying dynamics. Based on it, we extend this case with the consideration of integer constraints and reconfiguration delay. Such dynamic flow control allows us to monitor the number of produced products in real time while controlling WIP. As expected, WIP can be practically stabilized by usage of RMTs, cf., Figures 7.20 and 7.21. We would like to point out that in the time period between  $[35, 45]$ , there exists a demand increase concerning the quantity of product 1 and product 3. The RMTs are re-assigned expect workstation 2. Because the increasing orders are released to the system through workstation 1 which puts much burden on workstation 1, 3, 4 due to the product flow sequence (cf., Table 7.3) and the parameters setting (cf., Table 7.1). As there are only two DMTs within workstation 4,





**Figure 7.20.:** Number of produced products and  $WIP$  in the dynamic flow



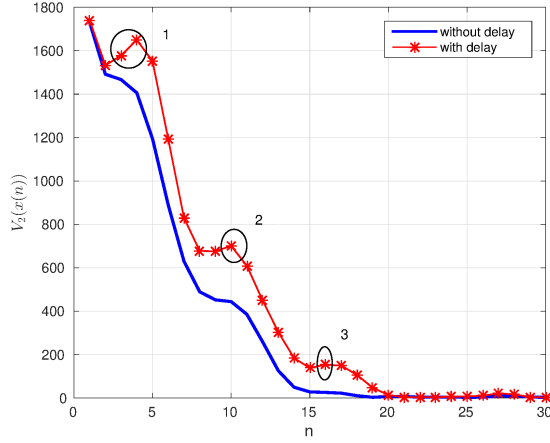
**Figure 7.21.:** Number of used RMTs in the dynamic flow

an increased number of RMTs is required to keep productivity. Also, the total produced numbers of products  $\hat{P}^1$ ,  $\hat{P}^2$  and  $\hat{P}^3$  are given in Figure 7.20, see more details in [2].

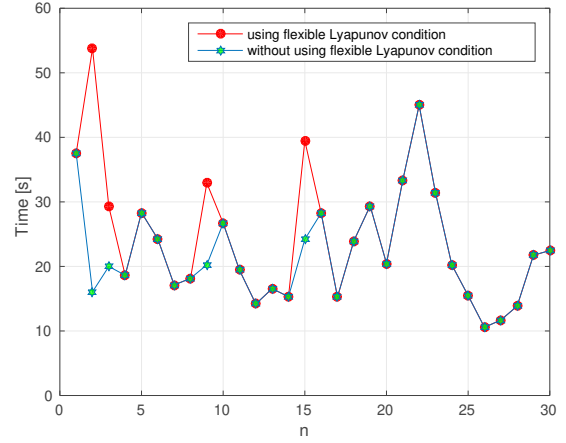
#### 7.2.4.3. PREDICTION HORIZON

As we know, the prediction horizon MPC plays an important role in terms of stability and closed-loop performance. But it is always discussed in case of continuous variables. With integer constraints, the computation burden would exponentially increase especially with an a posteriori suboptimality estimate for implicitly guaranteeing stability. Particularly, the inherent reconfiguration delay may render the system to be locally unstable, which we concluded in Section 7.2.3. To cope with it, we enlarge  $\varepsilon$  by extensive simulation such that the practical Lyapunov inequality condition (6.31) still holds. Yet, this local instability occurs at the steady-state stage with a small amplitude fluctuation. In this case, appropriately enlarging  $\varepsilon$  is a simple but applicable way to guarantee the practical stability. Yet, such an estimate of  $\varepsilon$  is not trivial especially with unknown external disturbance, which may lead to an useless estimate with inappropriate  $\varepsilon$ , see the detailed discussion in Section 6.2.3. To this end, we consider the flexible Lyapunov function combined with an adaptive MPC method, cf., Definition 6.13. The advantage of this method is that it allows for such local instabilities to exist but with a required decrease in future steps to converge to the practical stability region.

As we discussed before, the prerequisites to use flexible Lyapunov function is to be able to show  $\|x(n) - x^*\| \rightarrow \delta$  as  $n \rightarrow \infty$ . Based on the derived theoretical results from Section 6.2.5 and the given system parameters setting (e.g.,  $\ell(\cdot, \cdot)$ ,  $P$ ) which is identical to the previous setting in Section 7.2.1 but the external input rate is now changed to  $i_{01}(n) = 10 + 3\sin(0.3\pi n)$  and  $i_{02}(n) = 6$ ,  $n = 0, 1, \dots$  for representing the dynamic processes. More specifically, we study the case where a local instability occurs at transient phase and the current initial value is far way from the planned level. As a result, we cannot simply



**Figure 7.22.:** Comparison of optimal value function with and without delay



**Figure 7.23.:** Computing time with  $N = 2$  with flexible Lyapunov function

enlarge  $\varepsilon$ , and based on the system setting, we have

$$\left( \begin{array}{cccc} \overbrace{-2.0000 & 1.0000 & 0 & 0}^{a_1 := r^{RMT} \cdot P} \\ 0.8000 & -2.0000 & 0 & 0 \\ 1.2000 & 1.0000 & -2.0000 & 0 \\ 0 & 0 & 0.8000 & -2.0000 \end{array} \right) \left( \begin{array}{cccc} \overbrace{0.0651 & 0.0549 & 0.0247 & 0.0083}^{\mathbf{Id} + a_1 \Phi} \\ 0.0549 & 0.0682 & 0.0246 & 0.0083 \\ 0.0247 & 0.0246 & 0.0267 & 0.0090 \\ 0.0083 & 0.0083 & 0.0090 & 0.0241 \end{array} \right)$$

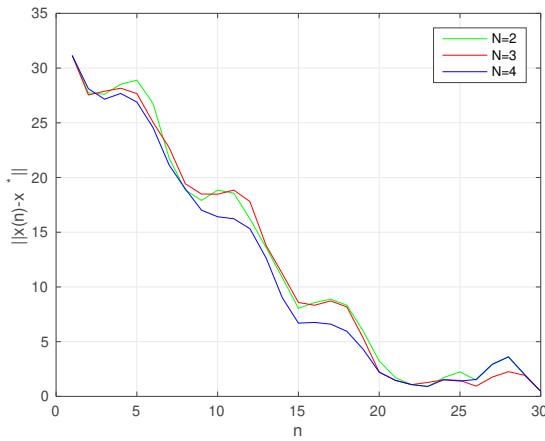
Using (6.42), we obtain

$$\lim_{n \rightarrow \infty} \|x(n) - x^*\|_2 \leq \frac{\xi_m}{1 - \|(\mathbf{Id} + a_1 \Phi)\|_2} = \overbrace{1.1558 \cdot \xi_m}^{\delta} \quad (7.7)$$

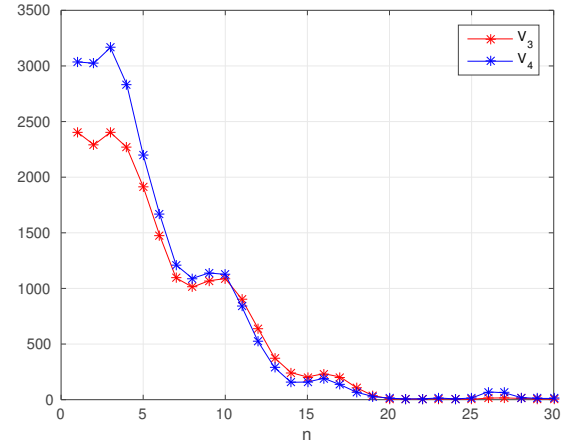
where  $\|(\mathbf{Id} + a_1 \Phi)\|_2 = 0.1348$ ,  $\xi_m = \max\{\|w(k)\|_2\}$ ,  $k = 0, 1, \dots, n-1$ . From (6.37), we know  $w(k) := a_1 \cdot \xi_{i+r}(k)$  that can be computed online through calculation of  $\xi_{i+r}(k)$  – the bounded “disturbance” caused by integer constraints and reconfiguration delays compared to the solution derived from continuous variables.

After it, the parameters in the flexible Lyapunov function condition (6.50) need to be designed before implementing. For simplicity but without loss of generality, based on Assumption 6.12 and the chosen step  $k = 5$ , we first employ MPC-B&B method with the initial prediction horizon  $N = 2$  to study the dynamic process in the framework of a flexible Lyapunov function combined with adaptive MPC.

In order to show such local instability is caused by the reconfiguration delay, we also present the results with integer constraints but without delay for comparison. The simulation result is illustrated in Figure 7.22 which shows the delay renders the  $V_2$  to non-monotonically decrease even before reaching the practical stability region (e.g.,  $(V_2(x(2))) <$



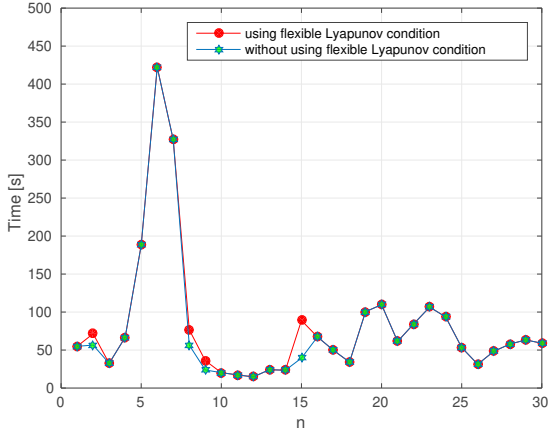
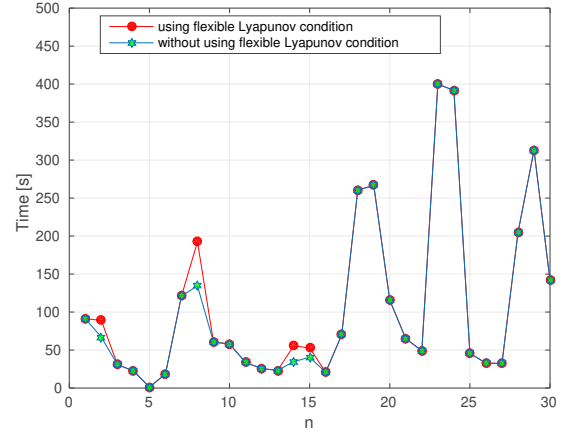
**Figure 7.24.:** Comparison of *WIP* tracking With different  $N$



**Figure 7.25.:** Variation of optimal value function  $V_N$  with  $N = 3$  and  $N = 4$

$V_2(x(3))$ ) with resulting  $\alpha_2(2) < 0$ . Yet, at  $n = 2$ , given the step  $k = 5$ , we iteratively check the flexible Lyapunov function condition by means of computing additional optimal control problems within the step  $k$  and find  $V_2(x(2 + 3)) < V_2(x(2 + 0))$ , cf., Figure 7.22, which shows that the flexible Lyapunov function condition holds. Then, we do not need to compute the left optimal control problems within  $k$  (i.e.,  $V_2(2 + 4)$  and  $V_2(2 + 5)$ ), and the prediction horizon  $N_n$  is kept unchanged and is used as the initial guess concerning the prediction horizon for the next iteration. Meanwhile, in Figure 7.23, we present the computing time at each time instant which shows that the computing time via flexible Lyapunov function is higher. This result is exactly corresponding to the Figure 7.22 in which the flexible Lyapunov function is only triggered when local instability occurs with the resulting increasing computing time.

Now the question arises whether the closed-loop performance improves and stability can be guaranteed by increasing  $N$  at the expense of computational cost in the presence of integer constraints and reconfiguration delay. To this end, we additionally employ MPC-B&B method with  $N = 3$  and  $N = 4$  for comparison in terms of *WIP* tracking performance, computing time and closed-loop stability. In Figure 7.24, we observe that closed-loop tracking performance with  $N = 2$  and  $N = 3$  are almost identical. In the contrast to that, the case with  $N = 4$  shows a faster convergence to the practical stability region. Yet, for each case, the monotonicity of  $V_N$  cannot be guaranteed, cf., Figure 7.25, and the computational cost increases along with the increase of  $N$  together with iteratively evaluating the flexible Lyapunov function, cf. Figures 7.26 and 7.27. That is to say, in some cases, increasing  $N$  may not improve the performance and cannot get rid of local instability caused by the inherent reconfiguration delay. To simplify the structure of controller and to not change  $N_n$  often, we prefer to use a short  $N$  but with a relative long step  $k$  in the framework of flexible Lyapunov function combined with adaptive MPC for achieving acceptable performance. There are several points to support it. The first is that since the computational cost

Figure 7.26.: Computing time with  $N = 3$ Figure 7.27.: Computing time with  $N = 4$ 

is exponentially increasing with the system dimension and length of prediction horizon, the time consumed by solving an optimal control problem with a large  $N$  is typically more than the time from solving multi small scale optimization problems. Second, as mentioned before, although with a relative long  $k$ , we do not need to compute all these additional optimal control problems at once but can check them iteratively. Third, if no  $k$  exists, then the  $N_n$  is required to be sufficiently large such that the flexible Lyapunov function (6.50) holds, which drastically aggravates the computational load and may render this method infeasible.

As we know, there are two critical components which are highly connected to the MPC closed-loop performance and stability. One is the importance of prediction horizon  $N$ . For instance, given a stage cost and system dynamics, a sufficiently large  $N$  is required for stability (3.42). The other is the stage cost. In this section, the stage cost is chosen as<sup>6</sup>

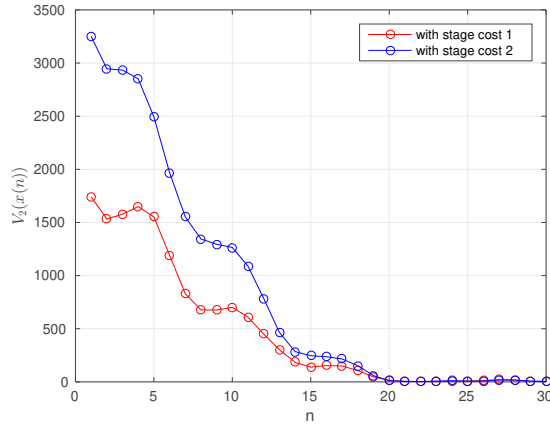
$$\ell(x(k), u(k)) = \|x(k) - x^*\|_2^2 + 0.1 \cdot \|u(k)\|_2^2. \quad (7.8)$$

Under the time-varying external input rate  $i_{01} = 10 + \sin(0.3\pi n)$ ,  $n = 0, 1, \dots$ , integer constraints and reconfiguration delays, monotonicity of  $V_N$  may not be guaranteed even for a larger  $N$ . Considering the system setting (e.g., product flow matrix  $P$ ) along with the respective control burden, through a repeated trial and error procedure we replace it by the following:

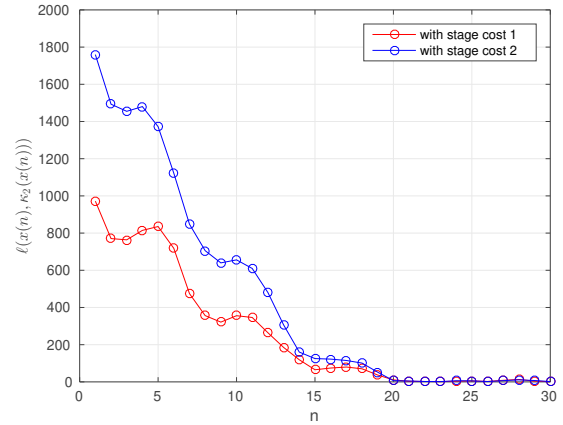
$$\begin{aligned} \ell(x(k), u(k)) = & 3 \cdot (x_1(k) - x_1^*)^2 + 2 \cdot (x_2(k) - x_2^*)^2 + 1.5 \cdot (x_3(k) - x_3^*)^2 \\ & + 0.5 \cdot (x_4(k) - x_4^*)^2 + 0.1 \cdot \|u(k)\|_2^2. \end{aligned} \quad (7.9)$$

with different penalty weighting coefficient for each workstation. Then,  $V_2$  is strictly decreasing before reaching the practical stability region, cf., Figure 7.28. The reason may be due to its associate stage cost  $\ell(x(k), u(k))$  with a small overshoot, cf., Figure 7.29 [107].

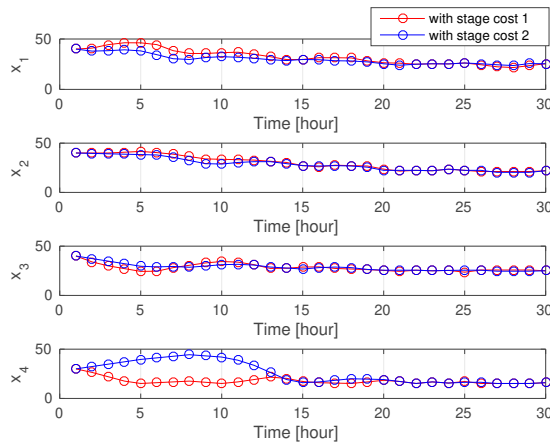
<sup>6</sup>(7.8) is called “stage cost 1” and (7.9) called “stage cost 2” in the following Figures.



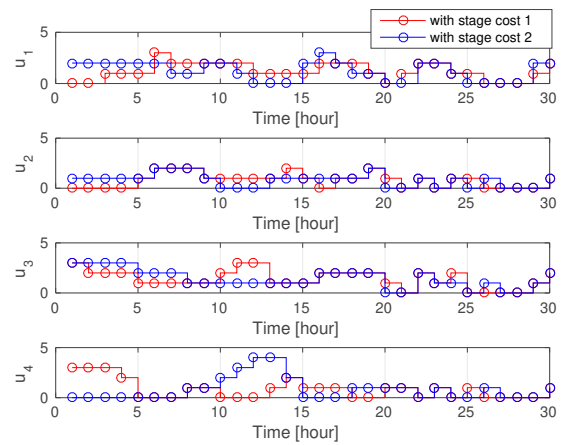
**Figure 7.28.:** Comparison of  $V_2(x(n))$  under different stage costs



**Figure 7.29.:** Comparison of  $\ell(\cdot, \kappa_2(x(n)))$  under different stage costs



**Figure 7.30.:** Variation of  $WIP$  under different stage costs



**Figure 7.31.:** Usage of RMTs under different stage costs

Note that the system designed for stability does not guarantee perfect closed-loop performance, cf., Figures 7.30 and 7.31 which represent the variation of  $x - (WIP)$  tracking and  $u$  – usage of RMTs for each workstation with different stage cost, respectively. For example, at the beginning, due to much penalty imposed on workstation 1 in case of with the stage cost (7.9), more RMTs are assigned to workstation 1 with the resulting decreased  $WIP_1$ . In contrast to that, there is no RMTs assigned to workstation 4 due to the less penalty on workstation 4. As a consequence,  $WIP_4$  increases. Concerning the relationship between closed-loop performance and stability, see Remark 6.5.

### 7.3. SUMMARY

In this chapter, according to the derived theoretical results from Chapter 6, we evaluated our proposed method through numerical case studies of a four workstation job shop system. In Section 7.1, we qualitatively analyzed the asymptotic stability with continuous assignment of RMTs in the cases with and without terminal endpoint constraints. In order to ensure the

feasibility of the former, we increased the basin of attraction and choose  $N = 16$  while the latter only needs  $N = 2$ . Then, we compared the applicability of MPC with standard PID in terms of WIP tracking and usage of RMTs, and observed that applying MPC directly showed better results as compared to PID, which was tuned manually. This showed the plug-and-play property of MPC. In Section 7.2, with integer constraints and reconfiguration delays, we compared different integer control strategies—PID with floor operator, MPC with floor operator, MPC with GA and MPC with B&B. The results indicated the latter two are almost identical and outperformed the former two.

In Section 7.2.2, for the trajectory controlled by MPC-B&B method, we analyzed the closed-loop stability and adopted the trajectory-dependent approach that estimates estimate the degree of suboptimality online. Further, we quantitatively analyzed the influence caused by reconfiguration delay on both transient and steady-state performance. The results showed that the existence of a reconfiguration delay leads to a slower convergence to the steady state and enlarges the size of practical stability region when fluctuation occurs on the steady state period. In order to guarantee practical stability, we enlarged  $\varepsilon$  to ensure  $\alpha \in (0, 1]$ . Further, through comparing the initial conditions via Figure 7.19 in Section 7.2.4.1, we demonstrated that the a posteriori estimate is less conservative but at the expense of computational cost, while the a priori estimate has less computation burden presents a more conservative estimate and is more dependent with the operating range of the job shop system, which in practice is defined by responsible managers in conjunction with choosing the prediction horizon.

In Section 7.2.4.3, we analyzed the role of the prediction horizon on closed-loop stability in the presence of integer constraints and reconfiguration delay. The results showed that increasing  $N$  may not improve performance and may not get rid of local instability caused by reconfiguration delay. After demonstrating that the closed-loop trajectory will be convergent to the practical stability region, in Section 7.2.4.3, we combined the flexible Lyapunov function with the adaptive MPC method, which allows  $V_N$  to increase locally but with a ensured decrease in future steps. At the end of this chapter, we discussed the relationship between the “good” stage cost and stability, and designed a tailored stage cost by trial and error method to ensure the monotonicity of  $V_N$  before reaching the practical stability region in the presence of integer constraints and reconfiguration delay.

Through numerical results, we showed that RMTs allow to adjust capacity and functionality of a job shop system effectively in the presence of demand fluctuations to maintain the WIP levels for each workstation, which we controlled by optimally reallocating RMTs using MPC. Thus, MPC represents a readily available and plug-and-play applicable tool to include RMTs into job shop systems. Under the complex time-varying dynamics with integer constraint and reconfiguration delay, an accurate prediction horizon and a good stage cost are very difficult to be constructed to guarantee the monotonicity of  $V_N$  before reaching the practical stability region. Here, combining a flexible Lyapunov function with the adaptive

MPC method simplifies the structure of controller without changing the prediction horizon too often and possibly reduce the computational burden with a short prediction horizon, which shows flexibility and applicability in practice.

# 8

## CONCLUSION AND OUTLOOK

### 8.1. CONCLUSION

According to the evolution of the market demand, there is a big trend to personalize products with shorter lifecycle, high variety and decreased production volumes for single variants. Relying on their high flexibility, job-shop production systems still retain their importance. Yet, this kind of system suffers from high work in process (WIP) with resulting bottlenecks which leads to long lead times and low reliability of due dates. In order to achieve a high quality of shop floor control performance in the turbulent industrial manufacturing environment, capacity adjustment as one effective measure is generally achieved by labor-oriented methods (e.g., overtime), which are already established in practice. Yet, the respective cost is relatively high and therefore not a sustainable solution in a long-term consideration. As an additional degree of freedom of capacity control, reconfigurability as a key enabler for handling exceptions and performance deteriorations in manufacturing operations have been developed and accomplished via reconfigurable machine tools (RMTs). This type of equipment is modularly designed for a part family of customized products in an open architecture environment. An increased production flexibility can be achieved by usage of these real time reconfigurable machines which allow a convertible, scalable and profitable personalized manufacturing process with small lot sizes. Moreover, the developed Internet of Things (IoT) with Radio Frequency Identification (RFID) techniques provide accurate information on the shop floor which renders closed-loop control to be practically applicable.

In this thesis, relying on the capability of flexibly changing capacity and functionality, we focused on machine level and considered a machinery-based approach via RMTs for capacity adjustment to compensate the unpredictable events in a rapid responsiveness and cost-effective way. As WIP is an essential variable that influences other performance indexes, we built up a discrete space state model of a job shop system with RMTs to maintain WIP to the planned level for each workstation. Compared to workload control in which



WIP could be adjusted by the rate of order release to the shop floor, we focused on the operational layer and therefore cannot determine the order release rate but internally implement capacity adjustment via RMTs to eliminate or shift bottlenecks to maximize the production process efficiency. In order to take full advantage of RMTs, we systematically formulated a set of reconfiguration rules for the determination of the triggered RMTs based on different objective function from the view of customers and manufacturers, respectively. For the former, the reconfiguration rule was obtained based on the goal of controlling WIP to the desired values for the improvement of reliability of due dates. In this case, due to the limited available RMTs, a priority policy was derived according to the tracking conditions, which may result in possible frequent reconfiguration, cf., Algorithm 3 (in the Page 54). For the latter, if the due date is not compulsively required, manufactures may prefer to consider production cost including inventory cost and reconfiguration cost as the primary goal to optimize and plan the operation of manufacturing process. After the analysis of the modelling layer, we considered the optimization and planning layer and proposed three strategies – floor operator, branch and bound (B&B) and genetic algorithm (GA) for resolving the integer assignment of RMTs. Last, on the control layer, we employed model predictive control (MPC), which allowed us to incorporate the system dynamics (modelling layer), performance indicators and constraints (planning layer) in a unified manner through solving an open-loop optimization control problem (3.25) (in the Page 33). After development and implementation of the proposed method, we compared them with the state of the art method PID as the benchmark. From the results, we concluded that under integer constraints and reconfiguration delay, the WIP cannot effectively be controlled by traditional PID with floor operator. MPC with B&B and MPC with GA showed competitive results and were capable of tracking WIP with a good performance in both of transient and steady-state. With the increase of system dimension and/or prediction horizon, the optimal solutions may not be obtainable in a reasonable time. Here, MPC with GA showed competitive results due to its fast search ability and found acceptable solution rendering it to be of preference. Since the solutions are derived through solving a NP-hard optimization problem incorporating complicated reconfiguration policies, it typically outperforms experienced workers.

In order to maintain WIP to the planned level and achieve a high reliability of due date to earn the trust of customers, the manufacturers hope to realize and keep their production process as stable as possible, especially in the presence of demand fluctuation along with the possible frequent but necessary reconfiguration via RMTs for capacity adjustment. In doing so, it may destabilize the system due to lost productivity in the reconfiguration phases. Relying on the property of prediction functionality, MPC is naturally capable of dealing with the relationship between prediction horizon and reconfiguration delay for a better performance. Yet, since MPC truncates the infinite optimization problem with a fixed finite receding time window and iteratively solves an open-loop optimization problem with given initial conditions, optimality and stability from the original infinite optimization problem

may be lost. To this end, considering the applicability in practice, we first adopt MPC with terminal endpoint constraints and unconstrained MPC scheme in case of continuous assignment of RMTs without integer constraints and reconfiguration delay. The asymptotic stability can be qualitatively guaranteed. Further, since there is a lack of the explicit characterization of the stability region in case of a practical implementation of MPC, based on the relaxed dynamic programming (6.31) (in the Page 74), we further quantitatively analyzed the practical stability via online estimating the degree of suboptimality to show the practical stability in the presence of integer constraints and reconfiguration delay, cf., Algorithm 5 (in the Page 75) or Algorithm 6 (in the Page 77). Since the system with practical stability has the advantage of executing an MPC scheme which is robust against a certain extend of perturbations, we used the inherent robustness due to the iteratively receding strategy. Moreover, we analyzed the impact caused by reconfiguration delay on the closed-loop in terms of transient and steady-state performance, and pointed out that such delay may render the Lyapunov function to be non-monotonous even before reaching the practical stability region. To cope with it, we either repeatedly design the stage cost with an appropriate prediction horizon to guarantee the Lyapunov function is monotonically decreasing outside of the practical stability region or we make use of the principle of a flexible Lyapunov function that allows the cost functional to increase locally but with an average decrease in future steps. For the latter, the tight classical one step Lyapunov stability inequality is extended to  $k$  step. Integrating adaptive MPC with the principle of a flexible Lyapunov function allow us to evaluate the closed-loop performance and stability via online estimating the degree of suboptimality and simplify the structure of the controller without changing the prediction horizon too often, possibly reducing the computational burden with a short prediction horizon, which shows flexibility and applicability in practice.

In the framework of Industrial 4.0, as a new promising equipment and a powerful enabler, RMTs will exhibit competitiveness for producing value-added and highly customised products. With the aid of enabling IoT, RMTs can be utilized effectively to adhere to quick and accurate information exchanges. MPC as control method based optimization can predict the future developments while explicitly considering constraints – input, output, state and even the reconfiguration delays. Given the goal of controlling WIP, combining MPC and RMTs for capacity adjustment will economically allow for a better compliance with logistics objectives and a sustainable demand oriented capacity allocation. Also, based on the plug-and-play property of MPC, the operators are able to simply change different operating points online, i.e., there is no need to repeatably tune parameters as in the PID case to achieve a good performance, which shows practical applicability of the proposed method.

## 8.2. OUTLOOK

There are some directions that could be starting points for the future research.

- **Modeling**

MPC has a high requirement on model which significantly influence the controller's performance. Relying on the advanced sensor techniques for real time monitoring the manufacturing process with uncertainty, there is a possibility to build dynamic models based on captured data which can be used by big data techniques from the domain of machine learning, which is similar to system identification in the framework of control theory [228, 229]. This is significant for optimizing shop floor operations and improving effectiveness of production schedules.

- **Planning and optimization**

Allocation of RMTs via MPC to balance capacity and load is actually a NP-hard optimization problem which is closely connected to system dimension and the length of prediction horizon for stability and/or closed-loop performance. In practice, the best solution may not be obtained in a reasonable time via deterministic methods especially for the non-convex mixed-integer optimization problems. Since they derive acceptable solutions Meta-heuristic methods are widely applied in applications which is of interest for this direction. Particularly, due to that fast solving speed, heuristics can be combined with a flexible Lyapunov function argument when the derived sub-optimal solution leads to local instabilities.

- **Control and stability**

To show stability, we used a trajectory-dependent approach to online estimate the degree of suboptimality in an a posteriori way. From Figure 7.19 (in the Page 109), we conclude that the a posteriori estimate is a less conservative estimate but at the expense of computational cost. In contrast to that, the a priori estimate shows less computational burden but with a more conservative result and is more dependent with initial conditions, typically require a large  $N$  for stability. In turn, it also aggravates the computational burden. Taking both advantages may be applicable when the designed stage cost is independent of  $u$  [130]. In this way, no additional computing burden arises from using the a priori estimate, and we can compute the suboptimality estimate only based on the computed closed-loop solutions. If this evaluated estimate  $\alpha$  is negative (e.g., initial conditions far away for the planned level), then we switch to using the a posteriori estimate and get a less conservative value. If this a posteriori estimate is still negative, then we use the flexible Lyapunov function, cf., Algorithm 8 (in the Page 86) as an example that combines the a priori estimate and the a posterior estimate through utilizing the upper bound  $V_N(n+2)$ . Also, the additional parameter  $N_0$  can be used for more accurate a priori estimate and will be considered in the framework of adaptive MPC with flexible Lyapunov function.

## REFERENCES

---

- [1] Yoram Koren and Moshe Shpitalni. Design of reconfigurable manufacturing systems. *Journal of manufacturing systems*, 29(4):130–141, 2010.
- [2] Qiang Zhang, Ping Liu, and Jürgen Pannek. Modeling and predictive capacity adjustment for job shop systems with RMTs. In *Control and Automation (MED), 2017 25th Mediterranean Conference on*, pages 310–315. IEEE, 2017.
- [3] Hans-Peter Wiendahl et al. *Load-oriented manufacturing control*. Springer, 1995.
- [4] Hermann Lödding. *Handbook of manufacturing control: Fundamentals, description, configuration*. Springer Science & Business Media, 2012.
- [5] Guoxin Wang, Xiwen Shang, Yan Yan, Janet K Allen, and Farrokh Mistree. A tree-based decision method for the configuration design of reconfigurable machine tools. *Journal of manufacturing systems*, 49:143–162, 2018.
- [6] Qiang Zhang, Ping Liu, and Jürgen Pannek. Combining mpc and integer operators for capacity adjustment in job-shop systems with RMTs. *International Journal of Production Research*, DOI:10.1080/00207543.2018.1521022, 2018.
- [7] Hans-Peter Wiendahl, Hoda A ElMaraghy, Peter Nyhuis, Michael F Zäh, H-H Wiendahl, Neil A Duffie, and Michael Brieke. Changeable manufacturing-classification, design and operation. *CIRP annals*, 56(2):783–809, 2007.
- [8] Bernd Scholz-Reiter, Michael Freitag, and Alf Schmieder. Modelling and control of production systems based on nonlinear dynamics theory. *CIRP Annals-Manufacturing Technology*, 51(1):375–378, 2002.
- [9] Hans-Peter Wiendahl and Jan-Wilhelm Breithaupt. Modelling and controlling the dynamics of production systems. *Production planning & control*, 10(4):389–401, 1999.
- [10] Neil A Duffie, Debjit Roy, and Leyuan Shi. Dynamic modeling of production networks of autonomous work systems with local capacity control. *CIRP Annals-Manufacturing Technology*, 57(1):463–466, 2008.

- [11] Bernd Scholz-Reiter, Dennis Lappe, and Sebastian Grundstein. Capacity adjustment based on reconfigurable machine tools—harmonising throughput time in job-shop manufacturing. *CIRP Annals-Manufacturing Technology*, 64(1):403–406, 2015.
- [12] Martin J Land, Mark Stevenson, Matthias Thüerer, and Gerard JC Gaalman. Job shop control: in search of the key to delivery improvements. *International Journal of Production Economics*, 168:257–266, 2015.
- [13] Neil A Duffie, John Fenske, and Madhu Vadali. Coordination of capacity adjustment modes in work systems with autonomous WIP regulation. In *Robust Manufacturing Control*, pages 135–145. Springer, 2013.
- [14] Marco Garetti and Marco Taisch. Sustainable manufacturing: trends and research challenges. *Production planning & control*, 23(2-3):83–104, 2012.
- [15] Faisal Hasan, PK Jain, and Upendra Kumar. Performance modeling of reconfigurable manufacturing system for different dispatching strategies under exception. In *2012 IEEE International Conference on Industrial Engineering and Engineering Management*, pages 593–596. IEEE, 2012.
- [16] Jan H Richter. *Reconfigurable control of nonlinear dynamical systems: a fault-hiding approach*, volume 408. Springer, 2011.
- [17] Neil A Duffie, Chehade A Abdallah, and Athavale Apoorva. Control theoretical modeling of transient behavior of production planning and control: A review. *Procedia CIRP*, 17:20–25, 2014.
- [18] Sergey Dashkovskiy, Michael Görges, Michael Kosmykov, Andrii Mironchenko, and Lars Naujok. Modeling and stability analysis of autonomously controlled production networks. *Logistics Research*, 3(2-3):145–157, 2011.
- [19] Bernd Scholz-Reiter, Sergey Dashkovskiy, Michael Görges, and Lars Naujok. Stability analysis of autonomously controlled production networks. *International Journal of Production Research*, 49(16):4857–4877, 2011.
- [20] Felicita Chromjaková, Roman Bobák, and D Hrusecka. Production process stability—core assumption of industry 4.0 concept. In *IOP Conference Series: Materials Science and Engineering*, volume 215, page 012024. IOP Publishing, 2017.
- [21] Joseph Z Lu. Challenging control problems and emerging technologies in enterprise optimization. *Control Engineering Practice*, 11(8):847–858, 2003.
- [22] Jens Tonne. *Robust Model Predictive Control for Large-Scale Manufacturing Systems subject to Uncertainties*. kassel university press GmbH, 2018.

- [23] Martin Land and Gerard Gaalman. Workload control concepts in job shops a critical assessment. *International journal of production economics*, 46:535–548, 1996.
- [24] Nuno O Fernandes and S Carmo-Silva. Workload control under continuous order release. *International Journal of Production Economics*, 131(1):257–262, 2011.
- [25] Behzad Esmailian, Sara Behdad, and Ben Wang. The evolution and future of manufacturing: A review. *Journal of Manufacturing Systems*, 39:79–100, 2016.
- [26] Bernd Scholz-Reiter, Christian Toonen, and Dennis Lappe. Impact of market dynamics on performance and internal dynamics of job-shop systems. *International Journal of Systems Applications, Engineering and Development*, 5(4):537–544, 2011.
- [27] Shyh-Chang Lin, Erik D Goodman, and William F Punch III. A genetic algorithm approach to dynamic job shop scheduling problem. In *ICGA*, pages 481–488, 1997.
- [28] Ferdinando Pezzella, Gianluca Morganti, and Gianfranco Ciaschetti. A genetic algorithm for the flexible job-shop scheduling problem. *Computers & Operations Research*, 35(10):3202–3212, 2008.
- [29] James C Chen, Cheng-Chun Wu, Chia-Wen Chen, and Kou-Huang Chen. Flexible job shop scheduling with parallel machines using genetic algorithm and grouping genetic algorithm. *Expert Systems with Applications*, 39(11):10016–10021, 2012.
- [30] Joaquim AS Gromicho, Jelke J Van Hoorn, Francisco Saldanha-da Gama, and Gerrit T Timmer. Solving the job-shop scheduling problem optimally by dynamic programming. *Computers & Operations Research*, 39(12):2968–2977, 2012.
- [31] Djamila Ouelhadj and Sanja Petrovic. A survey of dynamic scheduling in manufacturing systems. *Journal of scheduling*, 12(4):417, 2009.
- [32] Qiang Zhang and Jürgen Pannek. Predictive control of a job shop system with RMTs using equilibrium terminal constraints. In *International Conference on Dynamics in Logistics*, pages 361–367. Springer, 2018.
- [33] Bhavani S Earni. *Control-Theoretic Dynamic Modeling and Analysis of a Production System*. PhD thesis, Oklahoma State University, 2009.
- [34] George S Fishman. *Discrete-event simulation: modeling, programming, and analysis*. Springer Science & Business Media, 2013.
- [35] Ashkan Negahban and Jeffrey S Smith. Simulation for manufacturing system design and operation: Literature review and analysis. *Journal of Manufacturing Systems*, 33(2):241–261, 2014.



- [36] Albert Sweetser. A comparison of system dynamics (SD) and discrete event simulation (DES). In *17th International Conference of the System Dynamics Society*, pages 20–23, 1999.
- [37] Deogratias Kibira, Sanjay Jain, and Charles McLean. A system dynamics modeling framework for sustainable manufacturing. In *Proceedings of the 27th annual system dynamics society conference*, volume 301, pages 1–22, 2009.
- [38] Patroklos Georgiadis and Charalampos Michaloudis. Real-time production planning and control system for job-shop manufacturing: A system dynamics analysis. *European Journal of Operational Research*, 216(1):94–104, 2012.
- [39] Yadati Narahari and Nukala Viswanadham. Transient analysis of manufacturing systems performance. *IEEE Transactions on Robotics and Automation*, 10(2):230–244, 1994.
- [40] George Q Huang, Yongbin Zhang, and Pingyu Jiang. RFID-based wireless manufacturing for real-time management of job shop WIP inventories. *The International Journal of Advanced Manufacturing Technology*, 36(7):752–764, 2008.
- [41] Ray Y Zhong, George Q Huang, Shulin Lan, Qingyun Dai, Xu Chen, and Tao Zhang. A big data approach for logistics trajectory discovery from RFID-enabled production data. *International Journal of Production Economics*, 165:260–272, 2015.
- [42] Kai Ding and Pingyu Jiang. RFID-based production data analysis in an IoT-enabled smart job-shop. *IEEE/CAA Journal of Automatica Sinica*, 5(1):128–138, 2018.
- [43] Sven Axsater. Control theory concepts in production and inventory control. *International Journal of Systems Science*, 16(2):161–169, 1985.
- [44] Maximo Ortega and L Lin. Control theory applications to the production–inventory problem: a review. *International Journal of Production Research*, 42(11):2303–2322, 2004.
- [45] Gene F Franklin, J David Powell, Abbas Emami-Naeini, and J David Powell. *Feedback control of dynamic systems*, volume 3. Addison-Wesley Reading, MA, 1994.
- [46] Dimitri P Bertsekas, Dimitri P Bertsekas, Dimitri P Bertsekas, and Dimitri P Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena scientific Belmont, MA, 2005.
- [47] Haralambos Sarimveis, Panagiotis Patrinos, Chris D Tarantilis, and Chris T Kiranoudis. Dynamic modeling and control of supply chain systems: A review. *Computers & operations research*, 35(11):3530–3561, 2008.

- [48] Karl J Astrom and PR Kumar. Control: A perspective. *Automatica*, 50(1):3–43, 2014.
- [49] Dmitry Ivanov, Suresh Sethi, Alexandre Dolgui, and Boris Sokolov. A survey on control theory applications to operational systems, supply chain management, and industry 4.0. *Annual Reviews in Control*, 2018.
- [50] Jose M Framinan, Pedro L González, and Rafael Ruiz-Usano. The CONWIP production control system: review and research issues. *Production Planning & Control*, 14(3):255–265, 2003.
- [51] Hermann Lödding, Kwork-Wai Yu, and Hans-Peter Wiendahl. Decentralized WIP-oriented manufacturing control (DEWIP). *Production Planning & Control*, 14(1):42–54, 2003.
- [52] Jan-Wilhelm Breithaupt, Martin Land, and Peter Nyhuis. The workload control concept: theory and practical extensions of load oriented order release. *Production Planning & Control*, 13(7):625–638, 2002.
- [53] Matthias Thürer, Mark Stevenson, and Martin J Land. On the integration of input and output control: Workload control order release. *International Journal of Production Economics*, 174:43–53, 2016.
- [54] Christoph Roser, Kai Lorentzen, and Jochen Deuse. Reliable shop floor bottleneck detection for flow lines through process and inventory observations: the bottleneck walk. *Logistics Research*, 8(1):7, 2015.
- [55] Peter Nyhuis and Hans-Peter Wiendahl. *Fundamentals of production logistics: theory, tools and applications*. Springer Science & Business Media, 2008.
- [56] Günther Schuh, Till Potente, Christina Thomas, and Thomas Hempel. Short-term cyber-physical production management. *Procedia CIRP*, 25:154–160, 2014.
- [57] Hans-Peter Wiendahl and Jan-Wilhelm Breithaupt. Automatic production control applying control theory. *International Journal of Production Economics*, 63(1):33–46, 2000.
- [58] Jin-Hyung Kim and Neil A Duffie. Design and analysis of closed-loop capacity control for a multi-workstation production system. *CIRP Annals-Manufacturing Technology*, 54(1):455–458, 2005.
- [59] Jin-Hyung Kim and Neil A Duffie. Performance of coupled closed-loop workstation capacity controls in a multi-workstation production system. *CIRP Annals-Manufacturing Technology*, 55(1):449–452, 2006.



- [60] Ping Liu and Jürgen Pannek. Operator-based capacity control of job shop manufacturing systems with RMTs. In *International Conference on Dynamics in Logistics*, pages 264–272. Springer, 2018.
- [61] V Toshniwal, Neil A Duffie, Jagalski Thomas, Rekersbrink Henning, and Bernd Scholz-Reiter. Assessment of fidelity of control-theoretic models of WIP regulation in networks of autonomous work systems. *CIRP Annals-Manufacturing Technology*, 60(1):485–488, 2011.
- [62] Hamid R Karimi, Neil A Duffie, and Sergey Dashkovskiy. Local capacity control for production networks of autonomous work systems with time-varying delays. *IEEE Transactions on Automation Science and Engineering*, 7(4):849–857, 2010.
- [63] Chehade A Abdallah and Neil A Duffie. Optimal dynamic behavior of adaptive WIP regulation with multiple modes of capacity adjustment. *Procedia CIRP*, 19:168–173, 2014.
- [64] Juliana K Sagawa and Marcelo S Nagano. Modeling the dynamics of a multi-product manufacturing system: A real case application. *European Journal of Operational Research*, 244(2):624–636, 2015.
- [65] Tobias Sprodowski, Juliana K Sagawa, and Jürgen Pannek. Frequency based model predictive control of a manufacturing system. *IFAC-PapersOnLine*, 51(2):801–806, 2018.
- [66] Rodrigo Reyes Levalle, Manuel Scavarda, and Shimon Y Nof. Collaborative production line control: Minimisation of throughput variability and WIP. *International Journal of Production Research*, 51(23-24):7289–7307, 2013.
- [67] Yoram Koren, Uwe Heisel, Francesco Jovane, Toshimichi Moriwaki, Gunter Pritschow, Galip Ulsoy, and Hendrik Van Brussel. Reconfigurable manufacturing systems. *CIRP Annals-Manufacturing Technology*, 48(2):527–540, 1999.
- [68] Mostafa G Mehrabi, A Galip Ulsoy, and Yoram Koren. Reconfigurable manufacturing systems: Key to future manufacturing. *Journal of Intelligent manufacturing*, 11(4):403–419, 2000.
- [69] Shokraneh K Moghaddam, Mahmoud Houshmand, and Valilai F Omid. Configuration design in scalable reconfigurable manufacturing systems (RMS); a case of single-product flow line (SPFL). *International Journal of Production Research*, 56(11):3932–3954, 2018.

- [70] Sihan Huang, Guoxin Wang, and Yan Yan. Delayed reconfigurable manufacturing system. *International Journal of Production Research*, DOI:10.1080/00207543.2018.1518605, 2018.
- [71] Farshid Maghami Asl, A Galip Ulsoy, and Yoram Koren. Dynamic modeling and stability of the reconfiguration of manufacturing systems. In *Proc. of Japan-USA Symp. on Flexible Automation*, 2000.
- [72] Reza M Abdi and Ashraf Labib. Feasibility study of the tactical design justification for reconfigurable manufacturing systems using the fuzzy analytical hierarchical process. *International Journal of Production Research*, 42(15):3055–3076, 2004.
- [73] Cesar H Ortega Jimenez, Jose AD Machuca, Pedro Garrido-Vega, and Roberto Filipini. The pursuit of responsiveness in production environments: From flexibility to reconfigurability. *International Journal of Production Economics*, 163:157–172, 2015.
- [74] Alessia Napoleone, Alessandro Pozzetti, and Marco Macchi. A framework to manage reconfigurability in manufacturing. *International Journal of Production Research*, 56(11):3815–3837, 2018.
- [75] Durga Prasad and SC Jayswal. Reconfigurability consideration and scheduling of products in a manufacturing industry. *International Journal of Production Research*, 56(19):6430–6449, 2018.
- [76] Wichai Narongwanich, Izak Duenyas, and John R Birge. Optimal portfolio of reconfigurable and dedicated capacity under uncertainty. *Preprint. University of Michigan*, 2002.
- [77] Andreas Hees and Gunther Reinhart. Approach for production planning in reconfigurable manufacturing systems. *Procedia CIRP*, 33:70–75, 2015.
- [78] Andreas Hees, Christina Bayerl, Brian Van Vuuren, Corné SL Schutte, Stefan Bra unreuther, and Gunther Reinhart. A production planning method to optimally exploit the potential of reconfigurable manufacturing systems. *Procedia CIRP*, 62:181–186, 2017.
- [79] Sihan Huang, Guoxin Wang, Xiwen Shang, and Yan Yan. Reconfiguration point decision method based on dynamic complexity for reconfigurable manufacturing system (RMS). *Journal of Intelligent Manufacturing*, 29(5):1031–1043, 2018.
- [80] Kezia A Kurniadi and Kwangyeol Ryu. Development of IoT-based reconfigurable manufacturing system to solve reconfiguration planning problem. *Procedia Manufacturing*, 11:965–972, 2017.

- [81] Marco Bortolini, Francesco Gabriele Galizia, and Cristina Mora. Reconfigurable manufacturing systems: Literature review and research trend. *Journal of Manufacturing Systems*, 49:93–106, 2018.
- [82] Ahmed M Deif and Waguih H ElMaraghy. A control approach to explore the dynamics of capacity scalability in reconfigurable manufacturing systems. *Journal of Manufacturing Systems*, 25(1):12, 2006.
- [83] Eckart Uhlmann, Klaus-Dieter Lang, Lukas Prasol, Simon Thom, Bernd Peukert, Stephan Benecke, Eduard Wagner, Fiona Sammler, Sebastian Richarz, and Nils F Nissen. Sustainable solutions for machine tools. In *Sustainable Manufacturing*, pages 47–69. Springer, 2017.
- [84] Robert G Landers, B-K Min, and Yoram Koren. Reconfigurable machine tools. *CIRP Annals-Manufacturing Technology*, 50(1):269–274, 2001.
- [85] Manfredi Bruccoleri, Paolo Renna, and Giovanni Perrone. Reconfiguration: a key to handle exceptions and performance deteriorations in manufacturing operations. *International Journal of Production Research*, 43(19):4125–4145, 2005.
- [86] Moustafa Gadalla and Deyi Xue. Recent advances in research on reconfigurable machine tools: a literature review. *International Journal of Production Research*, 55(5):1440–1454, 2017.
- [87] Yong-Mo Moon and Sridhar Kota. Generalized kinematic modeling of reconfigurable machine tools. *Journal of Mechanical Design*, 124(1):47–51, 2002.
- [88] Eberhard Abele, Arno Wörn, Patrick Martin, and Robert Klöpper. Performance evaluation methods for mechanical interfaces in reconfigurable machine tools. In *International Symposium on Flexible Automation*, 2006.
- [89] Arda N Erol, Yusuf Altintas, and Mabo Robert Ito. Open system architecture modular tool kit for motion and machining process control. *IEEE/ASME transactions on mechatronics*, 5(3):281–291, 2000.
- [90] Zhuming Bi, Sherman YT Lang, M Verner, and P Orban. Development of reconfigurable machines. *The International Journal of Advanced Manufacturing Technology*, 39(11-12):1227–1251, 2008.
- [91] Eberhard Abele, Arno Wörn, Jürgen Fleischer, Jan Wieser, Patrick Martin, and Robert Klöpper. Mechanical module interfaces for reconfigurable machine tools. *Production Engineering*, 1(4):421–428, 2007.
- [92] Jared Padayachee and Glen Bright. Modular machine tools: Design and barriers to industrial implementation. *Journal of Manufacturing Systems*, 31(2):92–102, 2012.

- [93] Patrick Spicer, Derek Yip-Hoi, and Yoram Koren. Scalable reconfigurable equipment design principles. *International journal of production research*, 43(22):4839–4852, 2005.
- [94] Yoram Koren. Reconfigurable manufacturing and beyond. In *the summary of keynote Speech of CIRP05 3rd International Conference on Reconfigurable Manufacturing, Ann Arbor, Michigan, USA*, 2005.
- [95] Thomas Lorenzer, Sascha Weikert, Sergio Bossoni, and Konrad Wegener. Modeling and evaluation tool for supporting decisions on the design of reconfigurable machine tools. *Journal of Manufacturing Systems*, 26(3-4):167–177, 2007.
- [96] Jaspreet Dhupia, Bartosz Powalka, Reuven Katz, and A Galip Ulsoy. Dynamics of the arch-type reconfigurable machine tool. *International Journal of Machine Tools and Manufacture*, 47(2):326–334, 2007.
- [97] Moustafa Gadalla and Deyi Xue. An approach to identify the optimal configurations and reconfiguration processes for design of reconfigurable machine tools. *International Journal of Production Research*, 56(11):3880–3900, 2018.
- [98] Nan Xie, Aiping Li, and Wei Xue. Cooperative optimization of reconfigurable machine tool configurations and production process plan. *Chinese Journal of Mechanical Engineering*, 25(5):982–989, 2012.
- [99] Cristina Renzi, F Leali, Marco Cavazzuti, and AO Andrisano. A review on artificial intelligence applications to the optimal design of dedicated and reconfigurable manufacturing systems. *The International Journal of Advanced Manufacturing Technology*, 72(1-4):403–418, 2014.
- [100] Kapil Kumar Goyal, PK Jain, and Madhu Jain. Optimal configuration selection for reconfigurable manufacturing system using nsga ii and topsis. *International Journal of Production Research*, 50(15):4175–4191, 2012.
- [101] Wei Liu and Ming Liang. Multi-objective design optimization of reconfigurable machine tools: a modified fuzzy-chebyshev programming approach. *International Journal of Production Research*, 46(6):1587–1618, 2008.
- [102] Klaus Werner Schmidt. Computation of supervisors for reconfigurable machine tools. *Discrete Event Dynamic Systems*, 25(1-2):125–158, 2015.
- [103] Sibusiso M Ndzabukelwayo, Khumbulani Mpofo, and Nkgatho Tlale. An intelligent fuzzy PID controller for a reconfigurable machine tool. In *Information Security and Intelligence Control (ISIC), 2012 International Conference on*, pages 99–102. IEEE, 2012.

- [104] Pedro Ponce, Arturo Molina, Hector Bastida, and Brian MacCleery. Real-time hardware ANN-QFT robust controller for reconfigurable micro-machine tool. *The International Journal of Advanced Manufacturing Technology*, 79(1-4):1–20, 2015.
- [105] Kapil Kumar Goyal, Pramod Kumar Jain, and Madhu Jain. A novel methodology to measure the responsiveness of RMTs in reconfigurable manufacturing system. *Journal of Manufacturing Systems*, 32(4):724–730, 2013.
- [106] Ping Liu, Qiang Zhang, and Jürgen Pannek. Application of reconfigurable machine tools in the capacity control of job shop systems. *International Journal of Agile Systems and Management*, 11(3):206–221, 2018.
- [107] Lars Grüne and Jürgen Pannek. *Nonlinear model predictive control: Theory and Algorithms*. Springer, 2017.
- [108] Stephen Boyd and Lieven Vandenbergh. *Convex optimization*. Cambridge university press, 2004.
- [109] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.
- [110] Pietro Belotti, Christian Kirches, Sven Leyffer, Jeff Linderoth, James Luedtke, and Ashutosh Mahajan. Mixed-integer nonlinear optimization. *Acta Numerica*, 22:1–131, 2013.
- [111] Joseph Adams, Egon Balas, and Daniel Zawack. The shifting bottleneck procedure for job shop scheduling. *Management science*, 34(3):391–401, 1988.
- [112] Parviz Fattahi and Alireza Fallahi. Dynamic scheduling in flexible job shop systems by considering simultaneously efficiency and stability. *CIRP Journal of Manufacturing Science and Technology*, 2(2):114–123, 2010.
- [113] Mircea Lazar. *Model predictive control of hybrid systems: stability and robustness*. PhD thesis, Eindhoven University of Technology, 2006.
- [114] John Bellingham, Arthur Richards, and Jonathan P How. Receding horizon control of autonomous aerial vehicles. In *American Control Conference, 2002. Proceedings of the 2002*, volume 5, pages 3741–3746. IEEE, 2002.
- [115] Arthur Richards and Jonathan How. Mixed-integer programming for control. In *American Control Conference, 2005. Proceedings of the 2005*, pages 2676–2683. IEEE, 2005.
- [116] Jacob Roll, Alberto Bemporad, and Lennart Ljung. Identification of piecewise affine systems via mixed-integer programming. *Automatica*, 40(1):37–50, 2004.

- [117] Pierre Bonami, Mustafa Kilinç, and Jeff Linderoth. Algorithms and software for convex mixed integer nonlinear programs. In *Mixed integer nonlinear programming*, pages 1–39. Springer, 2012.
- [118] Pierre Bonami, Lorenz T Biegler, Andrew R Conn, Gérard Cornuéjols, Ignacio E Grossmann, Carl D Laird, Jon Lee, Andrea Lodi, François Margot, Nicolas Sawaya, et al. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization*, 5(2):186–204, 2008.
- [119] Jens Clausen. Branch and bound algorithms-principles and examples. *Department of Computer Science, University of Copenhagen*, pages 1–30, 1999.
- [120] Arthur M Geoffrion. Generalized benders decomposition. *Journal of optimization theory and applications*, 10(4):237–260, 1972.
- [121] Christian Kirches. *Fast numerical methods for mixed-integer nonlinear model-predictive control*. Springer, 2011.
- [122] Ambros M Gleixner. Exact and fast algorithms for mixed-integer nonlinear programming. 2015.
- [123] El-Ghazali Talbi. *Metaheuristics: from design to implementation*, volume 74. John Wiley & Sons, 2009.
- [124] Hassan K Khalil. Nonlinear systems. *3rd ed. Upper Saddle River, NJ, USA:Prentice-Hall*, 2002.
- [125] Daniel Limon, Teodoro Alamo, DM Raimondo, D Muñoz De La Peña, JM Bravo, A Ferramosca, and EF Camacho. Input-to-state stability: a unifying framework for robust model predictive control. In *Nonlinear model predictive control*, pages 1–26. Springer, 2009.
- [126] Jürgen Pannek and Karl Worthmann. Stability and performance guarantees for model predictive control algorithms without terminal constraints. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 94(4):317–330, 2014.
- [127] Ricardo P Aguilera and Daniel E Quevedo. Predictive control of power converters: Designs with guaranteed performance. *IEEE Transactions on Industrial Informatics*, 11(1):53–63, 2015.
- [128] Amir Ali Ahmadi and Pablo A Parrilo. Non-monotonic lyapunov functions for stability of discrete time nonlinear and switched systems. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 614–621. IEEE, 2008.



- [129] Mircea Lazar. Flexible control lyapunov functions. In *American Control Conference, 2009. ACC'09.*, pages 102–107. IEEE, 2009.
- [130] Lars Grüne and Jürgen Pannek. Practical NMPC suboptimality estimates along trajectories. *Systems & Control Letters*, 58(3):161–168, 2009.
- [131] Claus Muller, Daniel E Quevedo, and Graham C Goodwin. How good is quantized model predictive control with horizon one? *IEEE Transactions on Automatic Control*, 56(11):2623–2638, 2011.
- [132] Ricardo P Aguilera, Gabriel Urrutia, Ramón A Delgado, Daniel Dolz, and Juan C Agüero. Quadratic model predictive control including input cardinality constraints. *IEEE Transactions on Automatic Control*, 62(6):3068–3075, 2017.
- [133] Lars Grüne and Marleen Stieler. Asymptotic stability and transient optimality of economic MPC without terminal conditions. *Journal of Process Control*, 24(8):1187–1196, 2014.
- [134] Lalo Magni, Davide M Raimondo, and Riccardo Scattolini. Regional input-to-state stability for nonlinear model predictive control. *IEEE Transactions on automatic control*, 51(9):1548–1553, 2006.
- [135] Kiam Heong Ang, Gregory Chong, and Yun Li. PID control system analysis, design, and technology. *IEEE transactions on control systems technology*, 13(4):559–576, 2005.
- [136] S Joe Qin and Thomas A Badgwell. A survey of industrial model predictive control technology. *Control engineering practice*, 11(7):733–764, 2003.
- [137] Christopher Edwards and Sarah Spurgeon. *Sliding mode control: theory and applications*. Crc Press, 1998.
- [138] Karl J Åström and Tore Hägglund. The future of PID control. *Control engineering practice*, 9(11):1163–1175, 2001.
- [139] Pritesh Shah and Sudhir Agashe. Review of fractional PID controller. *Mechatronics*, 38:29–41, 2016.
- [140] Bart Saerens, Moritz Diehl, and Eric Van den Bulck. Optimal control using pontryagin’s maximum principle and dynamic programming. In *Automotive Model Predictive Control*, pages 119–138. Springer, 2010.
- [141] Hong Chen and Frank Allgöwer. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10):1205–1217, 1998.

- [142] Michael G Forbes, Rohit S Patwardhan, Hamza Hamadah, and R Bhushan Gopaluni. Model predictive control in industry: Challenges and opportunities. *IFAC-PapersOnLine*, 48(8):531–538, 2015.
- [143] Edgar Perea-Lopez, Erik B Ydstie, and Ignacio E Grossmann. A model predictive control strategy for supply chain optimization. *Computers & Chemical Engineering*, 27(8-9):1201–1218, 2003.
- [144] Kaushik Subramanian, James B Rawlings, Christos T Maravelias, Jesus Flores-Cerrillo, and Lawrence Megan. Integration of control theory and scheduling methods for supply chain management. *Computers & Chemical Engineering*, 51:4–20, 2013.
- [145] Donald E Kirk. *Optimal control theory: an introduction*. Courier Corporation, 2012.
- [146] Vryan Gil Palma. *Robust updated MPC schemes*. PhD thesis, Universität Bayreuth, 2015.
- [147] Tobias Raff, Steffen Huber, Zoltan K Nagy, and Frank Allgöwer. Nonlinear model predictive control of a four tank system: An experimental stability study. In *Control Applications, 2006. CCA'06. IEEE International Conference on*, pages 237–242. IEEE, 2006.
- [148] Sathiya S Keerthi and Elmer G Gilbert. Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving-horizon approximations. *Journal of optimization theory and applications*, 57(2):265–293, 1988.
- [149] David Q Mayne, James B Rawlings, Christopher V Rao, and Pierre OM Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.
- [150] David Q Mayne. An apologia for stabilising terminal conditions in model predictive control. *International Journal of Control*, 86(11):2090–2095, 2013.
- [151] Lars Grüne. NMPC without terminal constraints. In *proceedings of the IFAC conference on nonlinear model predictive control*, volume 2012, pages 1–13, 2012.
- [152] Andrea Boccia, Lars Grüne, and Karl Worthmann. Stability and feasibility of state constrained MPC without stabilizing terminal constraints. *Systems & control letters*, 72:14–21, 2014.
- [153] Hanna Michalska and David Q Mayne. Robust receding horizon control of constrained nonlinear systems. *IEEE Transactions on automatic control*, 38(11):1623–1633, 1993.



- [154] Devin W Griffith, Sachin C Patwardhan, and Lorenz T Biegler. Quasi-infinite adaptive horizon nonlinear model predictive control. *IFAC-PapersOnLine*, 51(18):506–511, 2018.
- [155] Stefano Di Cairano. An industry perspective on MPC in large volumes applications: Potential benefits and open challenges. *IFAC Proceedings Volumes*, 45(17):52–59, 2012.
- [156] Lars Grüne and Anders Rantzer. On the infinite horizon performance of receding horizon controllers. *IEEE Transactions on Automatic Control*, 53(9):2100–2111, 2008.
- [157] Bo Lincoln and Anders Rantzer. Relaxing dynamic programming. *IEEE Transactions on Automatic Control*, 51(8):1249–1260, 2006.
- [158] Karl Worthmann. *Stability analysis of unconstrained receding horizon control schemes*. PhD thesis, Universität Bayreuth, 2011.
- [159] Konstantinos Efthymiou, Aris Pagoropoulos, Nikolaos Papakostas, Dimitris Mourtzis, and George Chryssolouris. Manufacturing systems complexity review: challenges and outlook. *Procedia CIRP*, 3:644–649, 2012.
- [160] Nicole Stricker and Gisela Lanza. The concept of robustness in production systems and its correlation to disturbances. *Procedia CIRP*, 19:87–92, 2014.
- [161] Stephen M Disney, Denis R Towill, and Roger DH Warburton. On the equivalence of control theoretic, differential, and difference equation approaches to modeling supply chains. *International Journal of Production Economics*, 101(1):194–208, 2006.
- [162] Qiang Zhang, Michael Freitag, and Jürgen Pannek. Stability of predictive control in job shop system with reconfigurable machine tools for capacity adjustment. *Logistics Research*, accepted, 2019.
- [163] Dávid Gyulai, Botond Kádár, András Kovács, and László Monostori. Capacity management for assembly systems with dedicated and reconfigurable resources. *CIRP Annals-Manufacturing Technology*, 63(1):457–460, 2014.
- [164] Mirja Meyer, Marius-Vasile Apostu, and Katja Windt. Analyzing the influence of capacity adjustments on performance robustness in dynamic job-shop environments. *Procedia CIRP*, 7:449–454, 2013.
- [165] Neil A Duffie, Rekersbrink Henning, Leyuan Shi, D Halder, and J Blazei. Analysis of lead-time regulation in an autonomous work system. *Procedia CIRP*, pages 53–60, 2010.
- [166] Thomas Steffen. *Control reconfiguration of dynamical systems: linear approaches and structural tests*, volume 320. Springer Science & Business Media, 2005.

- [167] Eric L Haseltine and James B Rawlings. Critical evaluation of extended kalman filtering and moving-horizon estimation. *Industrial & engineering chemistry research*, 44(8):2451–2460, 2005.
- [168] Samuel Burer and Dieter Vandenbussche. A finite branch-and-bound algorithm for nonconvex quadratic programming via semidefinite relaxations. *Mathematical Programming*, 113(2):259–282, 2008.
- [169] Sven Leyffer. Integrating sqp and branch-and-bound for mixed integer nonlinear programming. *Computational optimization and applications*, 18(3):295–309, 2001.
- [170] Hiroaki Fukushima, Kazuyuki Kon, and Fumitoshi Matsuno. Model predictive formation control using branch-and-bound compatible with collision avoidance problems. *IEEE Transactions on Robotics*, 29(5):1308–1317, 2013.
- [171] Yonglei Zhang, XiaoJie Wu, and Xibo Yuan. A simplified branch and bound approach for model predictive control of multilevel cascaded h-bridge statcom. *IEEE Transactions on Industrial Electronics*, 64(10):7634–7644, 2017.
- [172] Barbara Mayer, Michaela Killian, and Martin Kozek. A branch and bound approach for building cooling supply control with hybrid model predictive control. *Energy and Buildings*, 128:553–566, 2016.
- [173] Laetitia Jourdan, Matthieu Basseur, and E-G Talbi. Hybridizing exact methods and metaheuristics: A taxonomy. *European Journal of Operational Research*, 199(3):620–629, 2009.
- [174] Elena C Laskari, Konstantinos E Parsopoulos, and Michael N Vrahatis. Particle swarm optimization for integer programming. In *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, volume 2, pages 1582–1587. IEEE, 2002.
- [175] Fang Xu, Hong Chen, Xun Gong, and Qin Mei. Fast nonlinear model predictive control on fpga using particle swarm optimization. *IEEE Transactions on Industrial Electronics*, 63(1):310–321, 2016.
- [176] Kalyanmoy Deb. An efficient constraint handling method for genetic algorithms. *Computer methods in applied mechanics and engineering*, 186(2-4):311–338, 2000.
- [177] Kusum Deep, Krishna Pratap Singh, Mitthan Lal Kansal, and Chander Mohan. A real coded genetic algorithm for solving integer and mixed integer optimization problems. *Applied Mathematics and Computation*, 212(2):505–518, 2009.
- [178] Gabriele Pannocchia, James B Rawlings, and Stephen J Wright. Conditions under which suboptimal nonlinear MPC is inherently robust. *Systems & Control Letters*, 60(9):747–755, 2011.

- [179] Lars Grüne and Jürgen Pannek. Analysis of unconstrained NMPC schemes with incomplete optimization. In *Proceedings of the 8th IFAC Symposium on Nonlinear Control Systems–NOLCOS*, pages 238–243, 2010.
- [180] Frank Allgöwer, Rolf Findeisen, Zoltan K Nagy, et al. Nonlinear model predictive control: From theory to application. *Journal-Chinese Institute Of Chemical Engineers*, 35(3):299–316, 2004.
- [181] Alberto Bemporad, Manfred Morari, Vivek Dua, and Efstratios N Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.
- [182] Melanie N Zeilinger, Colin N Jones, and Manfred Morari. Real-time suboptimal model predictive control using a combination of explicit mpc and online optimization. *IEEE Transactions on Automatic Control*, 56(7):1524–1534, 2011.
- [183] Sergio Lucia, Alexandru Tătulea-Codrean, Christian Schoppmeyer, and Sebastian Engell. Rapid development of modular and sustainable nonlinear model predictive control solutions. *Control Engineering Practice*, 60:51–62, 2017.
- [184] Petros Karamanakos, Tobias Geyer, and Ralph Kennel. Reformulation of the long-horizon direct model predictive control problem to reduce the computational effort. In *Energy Conversion Congress and Exposition (ECCE), 2014 IEEE*, pages 3512–3519. IEEE, 2014.
- [185] Tobias Geyer and Daniel E Quevedo. Multistep finite control set model predictive control for power electronics. *IEEE Transactions on power electronics*, 29(12):6836–6846, 2014.
- [186] Petros Karamanakos, Tobias Geyer, and Ralph Kennel. A computationally efficient model predictive control strategy for linear systems with integer inputs. *IEEE Transactions on Control Systems Technology*, 24(4):1463–1471, 2016.
- [187] Tobias Geyer. *Model predictive control of high power converters and industrial drives*. John Wiley & Sons, 2016.
- [188] Marleen Stieler. *Performance Estimates for Scalar and Multiobjective Model Predictive Control Schemes*. PhD thesis, Universität Bayreuth, 2018.
- [189] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.
- [190] Federico Di Palma and Lalo Magni. On optimality of nonlinear model predictive control. *Systems & control letters*, 56(1):58–61, 2007.

- [191] Vincent Bachtiar, Eric C Kerrigan, William H Moase, and Chris Manzie. Continuity and monotonicity of the MPC value function with respect to sampling time and prediction horizon. *Automatica*, 63:330–337, 2016.
- [192] Yongchang Wei, Hongwei Wang, and Chao Qi. On the stability and bullwhip effect of a production and inventory control system. *International journal of production research*, 51(1):154–171, 2013.
- [193] Nasr Al-Hinai and Tarek Y ElMekkawy. Robust and stable flexible job shop scheduling with random machine breakdowns using a hybrid genetic algorithm. *International Journal of Production Economics*, 132(2):279–291, 2011.
- [194] Gabriel R Frumusanu, Ionut C Constantin, Vasile Marinescu, and Alexandru Epureanu. Development of a stability intelligent control system for turning. *The International Journal of Advanced Manufacturing Technology*, 64(5-8):643–657, 2013.
- [195] Sergey Dashkovskiy, Björn S Rüffer, and Fabian R Wirth. An iss small gain theorem for general networks. *Mathematics of Control, Signals, and Systems*, 19(2):93–122, 2007.
- [196] Ping Liu, Undram Chinges, Qiang Zhang, and Jürgen Pannek. Capacity control in disturbed and time-delayed job shop manufacturing systems with RMTs. *IFAC-PapersOnLine*, 51(2):807–812, 2018.
- [197] Tobias Raff, Carsten Angrick, Rolf Findeisen, Jung-Su Kim, and Frank Allgöwer. Model predictive control for nonlinear time-delay systems. *IFAC Proceedings Volumes*, 40(12):60–65, 2007.
- [198] LU Jingyi, Cao Zhixing, and Gao Furong. A stable two-time dimensional (2d) model predictive control with zero terminal state constraints for constrained batch processes. *IFAC-PapersOnLine*, 48(8):514–519, 2015.
- [199] Philipp N Köhler, Matthias A Müller, Jürgen Pannek, and Frank Allgöwer. On exploitation of supply chain properties by sequential distributed mpc. *IFAC-PapersOnLine*, 50(1):7947–7952, 2017.
- [200] David Laurí, Barry Lennox, and Jose Camacho. Model predictive control for batch processes: Ensuring validity of predictions. *Journal of Process Control*, 24(1):239–249, 2014.
- [201] Eleni Aggelogiannaki, Philip Doganis, and Haralambos Sarimveis. An adaptive model predictive control configuration for production–inventory systems. *International Journal of Production Economics*, 114(1):165–178, 2008.

- [202] Michael A Stephens, Chris Manzie, and Malcolm C Good. Model predictive control for reference tracking on an industrial machine tool servo drive. *IEEE Transactions on Industrial Informatics*, 9(2):808–816, 2013.
- [203] Karl Worthmann, Mohamed W Mehrez, Mario Zanon, George KI Mann, Raymond G Gosine, and Moritz Diehl. Model predictive control of nonholonomic mobile robots without stabilizing constraints and costs. *IEEE Transactions on Control Systems Technology*, 24(4):1394–1406, 2016.
- [204] Matthias A Mueller and Karl Worthmann. Quadratic costs do not always work in MPC. *Automatica*, 82:269–277, 2017.
- [205] Franco Blanchini. Set invariance in control. *Automatica*, 35(11):1747–1767, 1999.
- [206] Daniel Limón, Teodoro Alamo, Francisco Salas, and Eduardo F Camacho. Input to state stability of min–max MPC controllers for nonlinear systems with bounded uncertainties. *Automatica*, 42(5):797–803, 2006.
- [207] Davide M Raimondo, Daniel Limon, Mircea Lazar, Lalo Magni, and Eduardo Fernández Camacho. Min-max model predictive control of nonlinear systems: A unifying overview on stability. *European Journal of Control*, 15(1):5–21, 2009.
- [208] David Q Mayne, María M Seron, and Sasa V Raković. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2):219–224, 2005.
- [209] David Q Mayne, Erric C Kerrigan, EJ Van Wyk, and Paola Falugi. Tube-based robust nonlinear model predictive control. *International Journal of Robust and Nonlinear Control*, 21(11):1341–1353, 2011.
- [210] Ali Mesbah. Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems*, 36(6):30–44, 2016.
- [211] David Q Mayne. Robust and stochastic model predictive control: Are we going in the right direction? *Annual Reviews in Control*, 41:184–192, 2016.
- [212] Ricardo P Aguilera and Daniel E Quevedo. Stability analysis of quadratic MPC with a discrete input alphabet. *IEEE Transactions on Automatic Control*, 58(12):3190–3196, 2013.
- [213] Bruno Picasso, Stefania Pancanti, Alberto Bemporad, and Antonio Bicchi. Receding-horizon control of LTI systems with quantized inputs. In *Analysis and Design of Hybrid Systems 2003 (ADHS 03): A Proceedings Volume from the IFAC Conference, St. Malo, Brittany, France, 16-18 June 2003*, volume 259. Elsevier Science, 2003.

- [214] Gene Grimm, Michael J Messina, Sezai E Tuna, and Andrew R Teel. Examples when nonlinear model predictive control is nonrobust. *Automatica*, 40(10):1729–1738, 2004.
- [215] Douglas A Allan, Cuyler N Bates, Michael J Risbeck, and James B Rawlings. On the inherent robustness of optimal and suboptimal nonlinear MPC. *Systems & Control Letters*, 106:68–78, 2017.
- [216] Alberto Bemporad and Manfred Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, 1999.
- [217] Daniel E Quevedo, Graham C Goodwin, and José A De Doná. Finite constraint set receding horizon quadratic control. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 14(4):355–377, 2004.
- [218] Jürgen Pannek. *Receding Horizon Control: A Suboptimality-based Approach*. PhD thesis, Universität Bayreuth, 2009.
- [219] James B Rawlings and Michael J Risbeck. Model predictive control with discrete actuators: Theory and application. *Automatica*, 78:258–265, 2017.
- [220] Thomas Jahn and Jürgen Pannek. Stability of constrained adaptive model predictive control algorithms. *IFAC Proceedings Volumes*, 44(1):9272–9277, 2011.
- [221] Sidhartha Panda, Binod K Sahu, and Pradeep K Mohanty. Design and performance analysis of PID controller for an automatic voltage regulator system using simplified particle swarm optimization. *Journal of the Franklin Institute*, 349(8):2609–2625, 2012.
- [222] Stephen Boyd, Martin Hast, and Karl J Åström. MIMO PID tuning via iterated lmi restriction. *International Journal of Robust and Nonlinear Control*, 26(8):1718–1731, 2016.
- [223] Jürgen Pannek. Parallelizing a state exchange strategy for noncooperative distributed NMPC. *Systems & Control Letters*, 62(1):29–36, 2013.
- [224] Eduardo Camponogara, Dong Jia, Bruce H Krogh, and Sarosh Talukdar. Distributed model predictive control. *IEEE Control Systems*, 22(1):44–52, 2002.
- [225] László Monostori, Paul Valckenaers, Alexandre Dolgui, Hervé Panetto, Mietek Brdys, and Balázs Csanád Csáji. Cooperative control in production and logistics. *Annual Reviews in Control*, 39:12–29, 2015.
- [226] Ping Liu, Qiang Zhang, and Jürgen Pannek. Capacity adjustment of job shop manufacturing systems with RMTs. In *Software, Knowledge, Information Management & Applications (SKIMA), 2016 10th International Conference on*, pages 175–180. IEEE.

- [227] Sohyung Cho and Alfonso Lazaro. Control theoretic model using PID controller for just-in-time production scheduling. *The International Journal of Advanced Manufacturing Technology*, 51(5-8):699–709, 2010.
- [228] Thorsten Wuest, Christopher Irgens, and Klaus-Dieter Thoben. An approach to monitoring quality in manufacturing using supervised machine learning on product state data. *Journal of Intelligent Manufacturing*, 25(5):1167–1180, 2014.
- [229] Thorsten Wuest, Daniel Weimer, Christopher Irgens, and Klaus-Dieter Thoben. Machine learning in manufacturing: advantages, challenges, and applications. *Production & Manufacturing Research*, 4(1):23–45, 2016.



# A

## APPENDIX

### IMPLEMENTATION

All the numerical experiments were implemented on MATLAB 2014b in the operation system Ubuntu 16.04 with processor Intel® Core™ i5-5200U CPU @ 2.20GHz × 4. The core optimization NMPC routine could be found at <http://www.nmpc-book.com>. The GA function is available in the global optimization toolbox. The BNB20 function could be downloaded freely online via the link <https://de.mathworks.com/matlabcentral/fileexchange/95-bnb>. Parts of code are as follows:

```
1  %fmincon function is called by BNB20 function for solving ...
    mixed-integer nonlinear programming in the framework of MPC, ...
    p-system dimension, N-prediction horizon, u-Number of RMTs, ...
    V-Optimal function value. The detailed interpretation for ...
    other parameters can be found in the above mentioned information.
2  index=ones(p·N,1);
3  options=optimset('fmincon');
4  [u,V]=BNB20(@(u) costfunction(runningcosts, ...
5      terminalcosts,system, N, T, t0, x0, ...
6      u, atol_ode_sim, rtol_ode_sim, type),u0,index',lb, ub,A, ...
7      b, Aeq, beq, ...
8      @(u) nonlinearconstraints(constraints, ...
9      terminalconstraints, ...
      system, N, T, t0, x0, u, ...
      atol_ode_sim, rtol_ode_sim, type),setts, options)
```



```

1 % GA is used for solving mixed-integer nonlinear programming in ...
   the framework of MPC, p-system dimension, N-prediction horizon
2 index=1:p·N;
3 options = gaoptimset(options, ...
4   'PlotFcn', {@gaplotbestf, @gaplotstopping}, ...
5   'MutationFcn', @mutationadaptfeasible, ...
6   'Tolcon', '', 'TolFun', '', 'Generations', '');
7 [u,V,exitflag,output] = ga(@(u) costfunction(runningcosts, ...
8   terminalcosts, system, N, T, t0, x0, ...
9   u, atol_ode_sim, rtol_ode_sim, ...
10  type), index, A, b, [], [], lb, ub, @(u) ...
11  nonlinearconstraints(constraints, terminalconstraints, ...
   system, N, T, t0, x0, u, ...
   atol_ode_sim, rtol_ode_sim, type), index, options)

```

## A.1. TERMINAL ENDPOINT CONSTRAINTS

In Section 6.1.1, we showed the asymptotic stability can be analytically guaranteed with the explicitly derived feedback law  $\kappa_2$  if the Assumptions 3.23 and 3.24 hold. Below, we present a more general way for analysis of asymptotic stability but with additional assumptions A.1 and A.2. As discussed before, the main challenge is to make  $V_N(x) \leq \alpha_2(\|x - x^*\|)$  holds.

### Assumption A.1

There exists a control sequence  $u^*(k)$  such that  $\ell(x^*(k), u^*(k)) \leq \tilde{\alpha}_2(\|x_0 - x^*\|)$  holds,  $k = 0, 1, \dots, N - 1$ . Then

$$V_N(x_0) = J(x_0, u^*(\cdot)) = \sum_{k=0}^{N-1} \ell(x^*(k), u^*(k)) \leq N\tilde{\alpha}_2(\|x_0 - x^*\|) \quad (\text{A.1})$$

where  $x_0 = x(0)$ ,  $\alpha_2(s) = N\tilde{\alpha}_2(s) \in \mathcal{K}$ .

### Assumption A.2

There exists a sequence of positive numbers  $\vartheta_k$  such that  $\|u^*(k) - u^*\| \leq \vartheta_k \|x_0 - x^*\|$ , and  $\eta_k > 0$  such that  $\|x^*(k) - x^*\| \leq \eta_k \|x_0 - x^*\|$ , where  $k = 0, 1, \dots, N - 1$ .

For simplicity of exposition and based on coordinate transformation, we set  $x := x - x^*$ ,  $u := u - u^*$ , below. The inequality holds when  $k = 0$ , for all  $\eta_k \geq 1$ ,

*Proof.* Assume  $\|x^*(k)\| \leq \eta_k \|x(n)\|$ ,  $x(n) = x_0$  holds for all  $0 \leq k \leq N - 2$ . Now we extend and prove it also holds at  $k + 1$ . Recall a class of linear affine dynamics  $x^+ = f(x, u, d) = Ax(k) + Bu(k) + d(k)$ , we have  $\|x^*(k + 1)\| = \|Ax^*(k) + Bu^*(k) + d(k)\|$ ,

where,  $\|x^*(k)\| \in \mathbb{X}_N$  and assume  $\|d(k)\| \leq \varrho\|x(n)\|$ , then

$$\begin{aligned} \|x^*(k+1)\| &\leq \|A\|\|x^*(k)\| + \|B\|\|u^*(k)\| + \|d(k)\| \\ &\leq (\|A\|\eta_k + \|B\|\vartheta_k + \varrho)\|x(n)\| := \eta_{k+1}\|x(n)\|. \end{aligned} \quad (\text{A.2})$$

Consequently, given the following stage cost, we have

$$\begin{aligned} \ell(x(k), u(k)) &= \|x(k)\|_Q^2 + \|u(k)\|_R^2 \\ &\leq \lambda_{\max}(Q)\|x(k)\|_2^2 + \lambda_{\max}(R)\|u(k)\|_2^2 \end{aligned}$$

if Assumption A.2 holds, then

$$\ell(x(k), u(k)) \leq (\eta_k^2 \lambda_{\max}(Q) + \vartheta_k^2 \lambda_{\max}(R))\|x(n)\|_2^2 := \tilde{\alpha}_2(\|x(n)\|). \quad (\text{A.3})$$

□

Then, Assumption A.1 holds and  $V_N(x) \leq \alpha_2(\|x - x^*\|)$  holds.

## A.2. PROOF FOR PROPOSITION 3.32

*Proof.* From the Proposition 3.32, we can obtain

$$V_N(x) - V_{N-1}(x) \leq \frac{\gamma^{N-1}}{(\gamma+1)^{N-2}} V_{N-1}(x) \quad (\text{A.4})$$

Based on the Assumption 3.31, we have

$$V_{k-1}(f(x, \kappa_k(x))) = V_k(x) - \ell(x, \kappa_k(x)) \leq (\gamma+1)\ell(x, \kappa_k(x)) - \ell(x, \kappa_k(x)) = \gamma\ell(x, \kappa_k(x))$$

Set  $k = N$ , we obtain  $V_{N-1}(f(x, \kappa_N(x))) \leq \gamma\ell(x, \kappa_N(x))$ . Now, according to (A.4) and set  $x := f(x, \kappa_N(x))$ , we have

$$V_N(f(x, \kappa_N(x))) - V_{N-1}(f(x, \kappa_N(x))) \leq \frac{\gamma^N}{(\gamma+1)^{N-2}} V_{N-1}(f(x, \kappa_N(x))) \quad (\text{A.5})$$

$$\leq \frac{\gamma^N}{(\gamma+1)^{N-2}} \ell(x, \kappa_N(x)) \quad (\text{A.6})$$

Then according to (3.36), we can conclude that

$$\alpha = 1 - \frac{\gamma^N}{(\gamma+1)^{N-2}} = \frac{(\gamma+1)^{N-2} - \gamma^N}{(\gamma+1)^{N-2}} \quad (\text{A.7})$$

□

### A.3. LINEAR QUADRATIC REGULATOR

Consider a class of linear time-invariant discrete time systems

$$x^+ = Ax(n) + Bu(n) \quad (\text{A.8})$$

with  $x \in \mathbb{R}^n$ ,  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{m \times p}$ ,  $u \in \mathbb{R}^p$ ,  $n \in \mathbb{N}_0$  and the following cost functional

$$J_N(x, u) = \sum_{k=0}^{N-1} x^\top(k)Qx(k) + u^\top(k)Ru(k) + x^\top(N)Q_Nx(N) \quad (\text{A.9})$$

with  $N \in \mathbb{N}_0$ , symmetric positive semi-definite matrices  $Q \in \mathbb{R}^{n \times n}$  and  $Q_N \in \mathbb{R}^{n \times n}$ , and a symmetric positive definite matrix  $R \in \mathbb{R}^{p \times p}$ , the goal is to find an optimal control sequence  $\{u(0), \dots, u(N-1)\}$  such that (A.9) is minimal with the known initial state  $x(0)$ . Based on the Bellmans principle of optimality, we now compute the optimal control sequence in a reverse way. That is, we first compute  $u(N-1)$  and define the resulting cost functional as  $C_1$ :

$$C_1(x(N-1)) = \min_{u(N-1)} u^\top(N-1)Ru(N-1) + x^\top(N)Q_Nx(N)$$

Given the system dynamic,  $u(N-1)$  can be analytically derived by a linear state feedback:

$$u(N-1) = -\overbrace{(B^\top Q_N B + R)^{-1} B^\top Q_N A}^{K_{N-1}} x(N-1) \quad (\text{A.10})$$

Then,

$$\begin{aligned} C_1(x(N-1)) &= (Ax(N-1) + BK_{N-1}x^\top(N-1))Q_N(Ax(N-1) + \\ &\quad BK_{N-1}x(N-1)) + (x^\top(N-1))K_{N-1}^\top RK_{N-1}x(N-1) \\ &= x^\top(N-1) \overbrace{(A + BK_{N-1})^\top Q_N (A + BK_{N-1}) + K_{N-1}^\top RK_{N-1}}^{P_{N-1}} \\ &\quad x(N-1) \end{aligned} \quad (\text{A.11})$$

Similarly, based on Bellman's optimality principle, we can compute  $u(N-2)$

$$C_2(x(N-2)) = \min_{u(N-2)} u^\top(N-2)Ru(N-2) + x^\top(N-1)Q_{N-1}x(N-1) + C_1(x(N-1)) \quad (\text{A.12})$$

Again based on system dynamics and the obtained  $C_1(x(N-1))$ ,  $u(N-2)$  can be likely expressed as  $u(N-2) = K_{N-2}x(N-2)$ . Consequently,  $u(k) = K_k x(k)$ , where  $K_k = -(B^\top P_{k+1} B + R)^{-1} B^\top P_{k+1} A$ . Assume  $P_k = P_{k+1}$ ,  $P_k$  can be derived through solving so

called discrete time algebraic Riccati equation.

$$P_k = A^\top P_k A + A^\top P_k B K_k + Q_k \quad (\text{A.13})$$

#### A.4. INDUCTION PROCESS FOR (6.25)-(6.27)

Since  $K = (B^\top P B + R)^{-1} B^\top P A$ ,  $A_k = A - B K$ , then (6.25) is induced as follow:

$$\begin{aligned} R K - B^\top P A_k &= R K - B^\top P (A - B K) \\ &= R K - B^\top P A + B^\top P B K \\ &= (R + B^\top P B) K - B^\top P A \\ &= (R + B^\top P B) (B^\top P B + R)^{-1} B^\top P A - B^\top P A = 0 \end{aligned} \quad (\text{A.14})$$

and induced (6.26) is expressed as:

$$\begin{aligned} K^\top R - A_k^\top P B \\ &= K^\top R - (A - B K)^\top P B \\ &= K^\top R - A^\top P B + K^\top B^\top P B \\ &= K^\top (R + B^\top P B) - A^\top P B \\ &= A^\top P B (B^\top P B + R)^{-1} (B^\top P B + R) - A^\top P B = 0 \end{aligned}$$

and the last (6.25) is:

$$\begin{aligned} A_k^\top P A_k - P + Q + K^\top R K \\ &= (A - B K)^\top P (A - B K) - P + Q + K^\top R K \\ &= A^\top P A - A^\top P B K - K^\top B^\top P A + K^\top B^\top P B K - P + Q + K^\top R K \end{aligned} \quad (\text{A.15})$$

Substitute discrete Riccati equation (A.13) into (A.15), then (A.15) is simplified as:

$$\begin{aligned} A_k^\top P A_k - P + Q + K^\top R K \\ &= -K^\top B^\top P A + K^\top B^\top P B K + K^\top R K \\ &= K^\top \overbrace{(R K + B^\top P B K - B^\top P A)}^{(\text{A.14})=0} \\ &= 0 \end{aligned}$$

# B

## LIST OF PUBLICATIONS

- Qiang Zhang, Michael Freitag, and Jürgen Pannek. Stability of predictive control in job shop system with reconfigurable machine tools for capacity adjustment. *Logistics Research*, 12 (3), 1-12, 2019.
- Qiang Zhang, Ping Liu, and Jürgen Pannek. Combining MPC and integer operators for capacity adjustment in job-shop systems with RMTs. *International Journal of Production Research*, 57(8), 2498-2513, 2019.
- Qiang Zhang and Jürgen Pannek. Predictive control of a job shop system with RMTs using equilibrium terminal constraints. In *International Conference on Dynamics in Logistics*, pages 361–367. Springer, 2018
- Qiang Zhang, Ping Liu, and Jürgen Pannek. Modeling and predictive capacity adjustment for job shop systems with RMTs. In *Control and Automation (MED), 2017 25th Mediterranean Conference on*, pages 310–315. IEEE, 2017.
- Ping Liu, Qiang Zhang, and Jürgen Pannek. Evaluation of control approaches for capacity adjustment in job shop systems, 9th conference MIM 2019 on *Manufacturing Modelling, Management and Control*, Berlin, accepted, 2019.
- Ping Liu, Qiang Zhang, and Jürgen Pannek. Application of reconfigurable machine tools in the capacity control of job shop systems. *International Journal of Agile Systems and Management*, 11(3):206–221, 2018.
- Ping Liu, Undram Chinges, Qiang Zhang, and Jürgen Pannek. Capacity control in disturbed and time-delayed job shop manufacturing systems with RMTs. *IFAC-PapersOn-Line*, 51(2):807–812, 2018.
- Undram Chinges, Ping Liu, Qiang Zhang, Ingrid Rügge and Jürgen Pannek. Abstract Control Interface for Job Shop Manufacturing Systems with RMTs. In *Software*,

*Knowledge, Information Management & Applications (SKIMA), 2017 11th International Conference on*, pages 1-6. IEEE, 2017.

- Ping Liu, Qiang Zhang, and Jürgen Pannek. Capacity adjustment of job shop manufacturing systems with RMTs. In *Software, Knowledge, Information Management & Applications (SKIMA), 2016 10th International Conference on*, pages 175–180. IEEE, 2016.