Digital twin for industrial systems: data-driven approaches for monitoring and control

Politecnico di Bari

Department of Electrical and Information Engineering
ELECTRICAL AND IFORMATION ENGINEERING
Ph.D. Program
SSD: ING-INF/04 – SYSTEMS AND CONTROL ENGINEERING

Final Dissertation

Digital Twin for Industrial Systems: Data-Driven Approaches for Monitoring and Control

---

by

Eng. Bozza Augusto

Supervisors:

Prof. Eng. Mariagrazia Dotoli
Prof. Eng. Raffaele Carli
Prof. Eng. Graziana Cavone

Coordinator of Ph.D. Program:

Prof. Eng. Mario Carpentieri

Course n°37, 01/11/2021 – 31/10/2024

Politecnico di Bari

LIBERATORIA PER L'ARCHIVIAZIONE DELLA TESI DI DOTTORATO

Al Magnifico Rettore
del Politecnico di Bari

Il sottoscritto BOZZA AUGUSTO nato a LUCERA (FG) il 29/06/1995, residente a LUCERA (FG) in via SILVIO MANCINI, 1 e-mail ABOZZA95@GMAIL.COM, iscritto al 3° anno di Corso di Dottorato di Ricerca in INGEGNERIA ELETTRICA E DELL'INFORMAZIONE ciclo XVII, ed essendo stato ammesso a sostenere l'esame finale con la prevista discussione della tesi dal titolo:

DIGITAL TWIN FOR INDUSTRIAL SYSTEMS: DATA-DRIVEN APPROACHES FOR MONITORING AND CONTROL

DICHIARA

1) di essere consapevole che, ai sensi del D.P.R. n. 445 del 28.12.2000, le dichiarazioni mendaci, la falsità negli atti e l'uso di atti falsi sono puniti ai sensi del Codice penale e delle Leggi speciali in materia, e che nel caso ricorressero dette ipotesi, decade fin dall'inizio e senza necessità di nessuna formalità dai benefici conseguenti al provvedimento emanato sulla base di tali dichiarazioni;
2) di essere iscritto al Corso di Dottorato di ricerca in INGEGNERIA ELETTRICA E DELL'INFORMAZIONE ciclo XVII, corso attivato ai sensi del "*Regolamento dei Corsi di Dottorato di ricerca del Politecnico di Bari"*, emanato con D.R. n.286 del 01.07.2013;
3) di essere pienamente a conoscenza delle disposizioni contenute nel predetto Regolamento in merito alla procedura di deposito, pubblicazione e autoarchiviazione della tesi di dottorato nell'Archivio Istituzionale ad accesso aperto alla letteratura scientifica;
4) di essere consapevole che attraverso l'autoarchiviazione delle tesi nell'Archivio Istituzionale ad accesso aperto alla letteratura scientifica del Politecnico di Bari (IRIS-POLIBA), l'Ateneo archivierà e renderà consultabile in rete (nel rispetto della Policy di Ateneo di cui al D.R. 642 del 13.11.2015) il testo completo della tesi di dottorato, fatta salva la possibilità di sottoscrizione di apposite licenze per le relative condizioni di utilizzo (di cui al sito http://www.creativecommons.it/Licenze), e fatte salve, altresì, le eventuali esigenze di "embargo", legate a strette considerazioni sulla tutelabilità e sfruttamento industriale/commerciale dei contenuti della tesi, da rappresentarsi mediante compilazione e sottoscrizione del modulo in calce (Richiesta di embargo);
5) che la tesi da depositare in IRIS-POLIBA, in formato digitale (PDF/A) sarà del tutto identica a quelle **consegnate**/inviate/da inviarsi ai componenti della commissione per l'esame finale e a qualsiasi altra copia depositata presso gli Uffici del Politecnico di Bari in forma cartacea o digitale, ovvero a quella da discutere in sede di esame finale, a quella da depositare, a cura dell'Ateneo, presso le Biblioteche Nazionali Centrali di Roma e Firenze e presso tutti gli Uffici competenti per legge al momento del deposito stesso, e che di conseguenza va esclusa qualsiasi responsabilità del Politecnico di Bari per quanto riguarda eventuali errori, imprecisioni o omissioni nei contenuti della tesi;
6) che il contenuto e l'organizzazione della tesi è opera originale realizzata dal sottoscritto e non compromette in alcun modo i diritti di terzi, ivi compresi quelli relativi alla sicurezza dei dati personali; che pertanto il Politecnico di Bari ed i suoi funzionari sono in ogni caso esenti da responsabilità di qualsivoglia natura: civile, amministrativa e penale e saranno dal sottoscritto tenuti indenni da qualsiasi richiesta o rivendicazione da parte di terzi;
7) che il contenuto della tesi non infrange in alcun modo il diritto d'Autore né gli obblighi connessi alla salvaguardia di diritti morali od economici di altri autori o di altri aventi diritto, sia per testi, immagini, foto, tabelle, o altre parti di cui la tesi è composta.

Bari, 03/01/2025 Firma Augusto Bozza

Il sottoscritto, con l'autoarchiviazione della propria tesi di dottorato nell'Archivio Istituzionale ad accesso aperto del Politecnico di Bari (POLIBA-IRIS), pur mantenendo su di essa tutti i diritti d'autore, morali ed economici, ai sensi della normativa vigente (Legge 633/1941 e ss.mm.ii.),

CONCEDE

- al Politecnico di Bari il permesso di trasferire l'opera su qualsiasi supporto e di convertirla in qualsiasi formato al fine di una corretta conservazione nel tempo. Il Politecnico di Bari garantisce che non verrà effettuata alcuna modifica al contenuto e alla struttura dell'opera.
- al Politecnico di Bari la possibilità di riprodurre l'opera in più di una copia per fini di sicurezza, back-up e conservazione.

Bari, 03/01/2025 Firma Augusto Bozza

Politecnico di Bari

Department of Electrical and Information Engineering
ELECTRICAL AND IFORMATION ENGINEERING
Ph.D. Program
SSD: ING-INF/04 – SYSTEMS AND CONTROL ENGINEERING

Final Dissertation

Digital Twin for Industrial Systems: Data-Driven Approaches for Monitoring and Control

---

by

Eng. Bozza Augusto

Referees:

Prof. Andrea Iannelli

Prof. Carlos A. Ocampo Martínez

Supervisors:

Prof. Eng. Mariagrazia Dotoli

Prof. Eng. Raffaele Carli

Prof. Eng. Graziana Cavone

Coordinator of Ph.D. Program:

Prof. Eng. Mario Carpentieri

Course n°37, 01/11/2021 – 31/10/2024

Politecnico di Bari

Eng. Augusto Bozza

Digital-Twin for Industrial Systems: Data-Driven Approaches for Monitoring and Control

Thesis submitted for the degree of Philosophiae Doctor

Department of Electrical and Information Engineering
Politecnico di Bari

Tutors
Prof. Eng. *Mariagrazia Dotoli*
Prof. Eng. *Raffaele Carli*
Prof. Eng. *Graziana Cavone*

2025

Dissertation submitted for the degree of *Philosophiae Doctor* in Electrical and Information Engineering (XXXVII cycle).

Title:

Digital-Twin for Industrial Systems: Data-Driven Approaches for Monitoring and Control

Ph.D Candidate:

Eng. *Augusto Bozza*, Politecnico di Bari (Bari, Italy)

Tutors:

Prof. Eng. *Mariagrazia Dotoli*, Politecnico di Bari (Bari, Italy)

Prof. Eng. *Raffaele Carli*, Politecnico di Bari (Bari, Italy)

Prof. Eng. *Graziana Cavone*, Università degli Studi Roma Tre (Roma, Italy)

Coordinator:

Prof. Eng. *Mario Carpentieri*, Politecnico di Bari (Bari, Italy)

External Reviewers:

Prof. *Andrea Iannelli*, Institute for Systems Theory and Automatic Control, Universität Stuttgart (Stuttgart, Germany).

Prof. *Carlos A. Ocampo Martínez*, Automatic Control Department, Universitat Politècnica de Catalunya – BarcelonaTECH (Barcelona, Spain).

Last version:

January 3, 2025

Abstract

Industrial Systems (ISs), ranging from manufacturing lines to more advanced networks, increasingly require smarter and smarter control mechanisms due to the complexities of modern industrial automation. However, traditional model-based techniques often fall short in capturing the complexities of these systems due to the inherent system nonlinearities and the exponential growth of process data. As a result, *Data-Driven* (DD) technology has emerged as a pivotal enabler for the digital transformation of modern ISs, promoting integrated and adaptive control solutions compliant to the so-called *Industry 4.0*. This evolution introduces advanced technologies, including Internet of Things, Machine Learning, and cyber-physical systems –especially *Digital Twin*–, which together allow systems to adaptively respond to real-time production demands and environmental changes.

Despite DD methodologies represent a promising alternative for improving system control and diagnostics, as well preserving process performance by leveraging real-time data, several technical limitations hindering their widespread application in real-world systems still exist. Within such a context, this thesis investigates the potential of DD methods for monitoring and controlling ISs, addressing their most modern challenges. To this aim, two primary research directions are pursued.

The first part of the thesis explores DD dynamic modeling for process control, developing indirect DD control methods for optimizing the process performance. Contributions in this direction include (i) the design of Model Predictive Control techniques for the deep drawing process, which significantly enhances process efficiency and product quality, and (ii) a robust control scheme for input-affine nonlinear systems using Subspace Identification of Nonlinear Dynamics and online Semi-Definite Programming. This control approach offers superior stability and performance for such a class of nonlinear models without approximating their dynamics.

The second part of the thesis investigates DD adaptive control and fault detection methods for preserving the performance of industrial processes. This includes (i) an Adaptive Model Predictive Control for Hydraulic Servo Actuators (HSAs) based on flow control valves, and (ii) a novel Model Reference Adaptive Control framework for multi-chamber HSAs based on pressure control valves. Additionally, (iii) a new Adaptive Constrained Clustering algorithm is introduced for the real-time fault detection of ISs, effectively distinguishing between nominal and non-nominal working conditions in a dynamic operational context.

To those who have believed in me
even when I did not,

Contents

List of Figures

List of Tables

Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of *Philosophiae Doctor* in Electrical and Information Engineering at the *Politecnico di Bari*.

The research presented in this dissertation was conducted at the *Decision and Control Laboratory* of the Politecnico di Bari under the supervision of Professors **Mariagrazia Dotoli** and **Raffaele Carli**, in collaboration with Professor **Graziana Cavone** from *Università degli Studi Roma Tre*, between November 2021 and October 2024.

To the best of my knowledge, this work is original, except where acknowledgments and references are made to previous works. Most part of this thesis reports the results published during this research period in different scientific publications where I am one of the authors. An introductory chapter (Chapter 1) precedes the papers, relating them to one another and providing the necessary background information and motivation for the entire work. An original version of Chapters 2, 3, 5, and 6 have been presented in/submitted to International Journals [1]–[4], while the content of Chapter 4 was published in the proceedings of 2022 IEEE International Conference on Automation Engineering [5]. It is worth noting that [4] was accepted for presentation at the 2023 European Control Conference and was selected by the Program Committee for inclusion in the Special Issue of the European Journal of Control collecting substantial extensions of the best conference papers. The concluding chapter (Chapter 7) summarizes the main outcomes and findings for future developments, while Appendix A includes some theoretical preliminaries.

I would like to express my gratitude to Professors **Mariagrazia Dotoli**, **Raffaele Carli**, and **Graziana Cavone** for their invaluable contributions to the early stages of concept formation, as well as data collection, numerical implementation, analysis of experiments, which significantly shaped the above-mentioned manuscripts.

The experimental dataset used for the system identification in Chapter 2 was provided by the **SMATI (Materials and Innovative Technology) Research Group** at the *Politecnico di Bari*. Moreover, most research activities were supported by **Telematics LAB** at the *Politecnico di Bari*, **CRF (Centro Ricerche Fiat)**, and **Gigant Company Srl**.

Chapter 3 results from an international research collaboration with Professor **Frank Allgöwer** and Doctor **Tim Martin**, from the *Institute of Systems Theory and Automatic Control* at the *University of Stuttgart*, Germany, whose contributions were fundamental. In particular, I would especially like to thank Doctor Tim Martin for his valuable involvement in the early stages of concept development, theoretical results, and numerical implementation of the manuscript.

The experimental dataset used in Chapter 6 was provided by **MASMEC Company SpA**, and I sincerely appreciate their support throughout my research activities. I also acknowledge the collaboration with Doctor **Bahman Askari** in data collection, conceptualization, numerical implementation, and results analysis, as well as the preparation of the original manuscript.

In addition to the works presented in this thesis, I am a co-author of manuscript [6], [7], which focus on the identification, modeling, and control of industrial and mechatronic systems: these works are not included in this thesis due to space constraints. A full list of my co-authored papers is reported hereafter.

List of Papers Written by the Author

International Journal Articles

[1] Cavone, G., Bozza, A., Carli, R., and Dotoli, M., "MPC-Based Process Control of Deep Drawing: An Industry 4.0 Case Study in Automotive," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 3, pp. 1586–1598, 2022. DOI: 10.1109/TASE.2022.3177362.

[2] Bozza, A., Martin, T., Cavone, G., Carli, R., Dotoli, M., and Allgöwer, F., "Online data-driven control of nonlinear systems using semidefinite programming," *IEEE Control Systems Letters*, 2024. DOI: 10.1109/LCSYS.2024.3521645.

[3] Bozza, A., Cavone, G., Carli, R., and Dotoli, M., "PWM-Based Energy-Efficient Adaptive Control for Multi-Chamber Hydraulic Servo Actuators," *IEEE Transactions on Control Systems Tecnology*, **under review**.

[4] Askari, B., Bozza, A., Cavone, G., Carli, R., and Dotoli, M., "An Adaptive Constrained Clustering Approach for Real-Time Fault Detection of Industrial Systems," *European Journal of Control*, p. 100 858, 2023. DOI: 10.1016/j.ejcon.2023.100858.

International Conference Proceedings

[5] Bozza, A., Askari, B., Cavone, G., Carli, R., and Dotoli, M., "An Adaptive Model Predictive Control Approach for Position Tracking and Force Control of a Hydraulic Actuator," in *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, 2022, pp. 1029–1034. DOI: 10.1109/CASE49997.2022.9926645.

[6] Bozza, A., Cavone, G., Carli, R., Mazzoccoli, L., and Dotoli, M., "An MPC-based Approach for the Feedback Control of the Cold Sheet Metal Forming Process," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, 2021, pp. 286–291. DOI: 10.1109/CASE49439.2021.9551602.

[7] Bozza, A., Cavone, G., Carli, R., and Dotoli, M., "A Power Electronic Converters-Inspired Approach for Modeling PWM Switched-Based Nonlinear Hydraulic Servo Actuators," in *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2023, pp. 2477–2482. DOI: 10.1109/SMC53992.2023.10394065.

Acknowledgements

The higher the mountain, harsher the climbing, and bigger the team. Pursuing this Ph.D. has been a truly life-changing experience, and surely I would have given up many times without the unwavering support and guidance of so many people along the way. Hence, I would like to spend some words to thank those who have encouraged, motivated, and endured me, had my back, and all the other countless things that have kept me from giving up or going insane.

I would like to begin by expressing my heartfelt gratitude to Professor *Mariagrazia Dotoli* for serving as my primary supervisor during my activities. Her suggestions were invaluable for providing me with the best way to tackle all the troubles I encountered throughout this challenging path. Likewise, her guidance and example has certainly contributed to foster my growth from both professional as well personal perspectives. I would also like to extend my gratitude to Professors *Raffaele Carli* and *Graziana Cavone* for the huge patience and guidance offered me while their supervision activity. They consistently have advised me on the next steps to take, and even when things did not go as planned, they always found a way to keep me motivated. Without their unwavering support, it would have been definitely a lot harder reaching this ambitious goal.

Further, I greatly appreciated the opportunity to collaborate with Professor *Frank Allgöwer* and Doctor *Tim Martin*. Their insightful suggestions were always precious, broadening my perspective and enriching my scientific practice.

I would also like to thank all the members, past and present, of the *D&C Lab research group* whom I have had the pleasure of meeting over these years. I really enjoyed coming to work, attending conferences, and participating in various group activities with all of them.

And last, but not least, I want to express my gratitude to my girlfriend *Amira* and all my family, particularly my parents *Rino* and *Annarita* and my sister *Federica* with her husband *Francesco*, for always supporting and enduring me, especially during the worst moments of this challenging experience. Without their presence, none of this would have been possible. Thank you for always believing in me and helping me to reach this goal. Your encouragement and guidance meant more to me than words can ever express.

Acronyms

AC *Alternative Current*. 67, 116
ACC *Adaptive Constrained Clustering*. 88, 90, 105
AEDM *Average Equivalent Discrete-time Model*. 62
AI *Artificial Intelligence*. 1
AIC *Akaike's Information Criterion*. 18, 113
AMPC *Adaptive Model Predictive Control*. 9, 49, 50, 105
ARMAX *AutoRegressive Moving Average eXogenous*. 110
ARX *AutoRegressive eXogenous*. 110

BD *Big Data*. 1
BHF *Blank Holder Force*. 8, 15, 16

CBM *Condition Based Monitoring*. 88
CN *Connection and Integration*. 3

DC *Direct Current*. 67, 116
DD *Data-Driven*. 1, 17, 36, 88, 104
DDR *Data-Driven Robust*. 37
DFCU *Digital Flow Control Unit*. 62
DPI *Decoupled Proportional-Integral*. 9, 62
DS *Data Stream*. 89
DT *Digital Twin*. 2, 17, 104

ERP *Enterprise Resource Planning*. 1

FPE *Akaike's Final Prediction Error*. 18, 113
FTC *Fault-Tolerant Control*. 5

HSAs *Hydraulic Servo Actuators*. 9, 49, 61, 104
HW *Hammerstein-Wiener*. 8, 17, 111

IOT *Internet of Things*. 1
ISs *Industrial Systems*. 1, 88, 104

LMIs *Linear Matrix Inequalities*. 36
LQ *Linear Quadratic*. 64
LTI *Linear Time-Invariant*. 50, 107
LTV *Linear Time-Varying*. 50
LTVKF *Linear Time-Varying Kalman Filter*. 52

MDL *Rissanen's Minimum Description Length*. 18, 113
MES *Manufacturing Execution System*. 1
MIMO *Multiple-Input Multiple-Output*. 50, 66, 107
MISO *Multiple-Input Single-Output*. 62, 65
ML *Machine Learning*. 1, 89
MPC *Model Predictive Control*. 6, 15, 17, 50, 61, 64
MRAC *Model Reference Adaptive Control*. 62, 114
MSE *Mean Square Error*. 56, 78

NARX *Nonlinear Auto Regressive eXogenous*. 18

OPC UA *Platform Communications Unified Architecture*. 24

PCM *Pulse Code Modulation*. 63
PE *Persistently Exciting*. 108
PhE *Physical Entity*. 3
PHM *Prognostics and Health Management*. 88
PID *Proportional-Integral-Derivative*. 15, 17, 50
PLC *Programmable Logic Controller*. 1, 24
PNM *Pulse Number Modulation*. 63
PWM *Pulse Width Modulation*. 61, 62

SCADA *Supervisory Control And Data Acquisition*. 1
SDP Semi-Definite Programming. 8, 36, 104
SIMO *Single-Input Multiple-Output*. 8, 17, 19
SINDy *Subspace Identification of Nonlinear Dynamics*. 37, 112
SISO *Single-Input Single-Output*. 18, 107
SS *Silhouette Score*. 97
SSE *Sum of Squared Errors*. 97
SWIL *software-in-the-loop*. 8, 24

TCP/IP *Transmission Control Protocol/Internet Protocol*. 24

VC *Virtual Commissioning*. 24
VE *Virtual Entity*. 3

ZOH *Zero-Order Hold*. 71

Chapter 1

Introduction

Despite a unified definition has not yet been formalized, *Industrial Systems* (ISs) can refer to a production line, an entire workshop, or any type of machining system characterized by systematization and automation features. The former involves organizing manufacturing resources and related elements to construct an organic whole, while the latter pertains to the use of automated procedures in the process [1]. Effectively, an IS is generally referred to a complex network operating automated control systems [2].

Industrial automation, defined as automation for ISs, generally involves the five-levels pyramidal classification shown in Fig. 1.1. At the base of the pyramid is the *field level*, consisting of physical devices (e.g., sensors and actuators) used to generate data and handle the physical operations of the industrial facility. The second level of automation (i.e., *control level*) is responsible for controlling the processes and is usually based on Programmable Logic Controllers (PLCs). Further, the *supervisory level* implements Supervisory Control And Data Acquisition (SCADA) systems to monitor and control the IS while collecting information remotely. The *planning level* leverages the Manufacturing Execution System (MES) to oversee the entire manufacturing process, from raw material usage to on-time product delivery. At the top of the automation pyramid is the *management level*, which utilizes the Enterprise Resource Planning (ERP) software –a comprehensive management system that integrates and manages all business processes within the factory [3].

The data exchange is typically limited to adjacent levels and this hierarchical classification does not support the integration of multiple vendors. Therefore, these layers are not fully interconnected, leading to inefficiencies and suboptimal decision-making. To address these issues, enabling technologies from the ongoing fourth Industrial Revolution have emerged as key factors in improving integration and efficiency [4].

1.1 From Industry 4.0 to Industry 5.0: New Frontiers in Industrial Automation

The term *Industry 4.0* (or *fourth Industrial Revolution*) was coined in 2011 by Wolfgang Wahlster, a full professor in Computer Science at Saarland University. It refers to the digital transformation of industrial processes exploiting automation, interconnection, and real-time process optimization [5]. The key enabling technologies include Internet of Things (IOT), Machine Learning (ML), Artificial Intelligence (AI), Big Data (BD), cloud computing, cyber-physical systems, and cybersecurity [6], [7]. To this aim, the fourth Industrial Revolution is based on the digitization of industrial processes and machines for enhancing their operational efficiency, productivity, and automation compared to previous eras [8]. This transformation results in smart, fully connected, and *Data-Driven* (DD) ISs.

As a result, ISs have evolved into the concept of *smart ISs*, defined as *"fully-integrated, collaborative and responsive operations that respond in real-time to meet changing demands and conditions in the factory, in the supply network, and in customer needs via data-driven understanding, reasoning, planning, and execution of all aspects of manufacturing processes, facilitated by the pervasive use of advanced sensing, modeling, simulation, and analytics technologies"* [9]. By utilizing advanced sensing, data processing, and decision-making technologies, these systems are capable of producing manufacturing operations that respond on demand independently [10].

Based on the above definition, the key features of such systems include *digitization* (i.e., the use of multiple sensing systems, simulation, and analysis), *cyberization* (i.e., on the

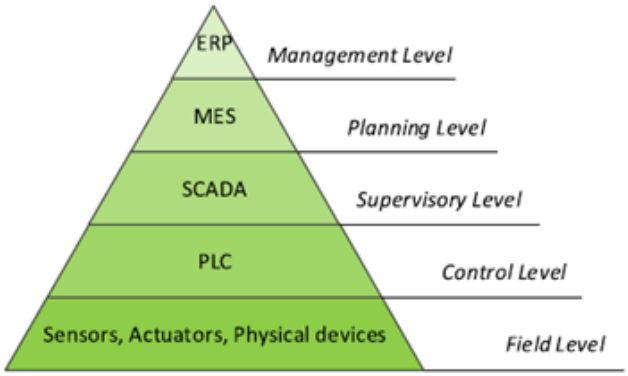

Figure 1.1: Five-level pyramidal classification of industrial automation [3].

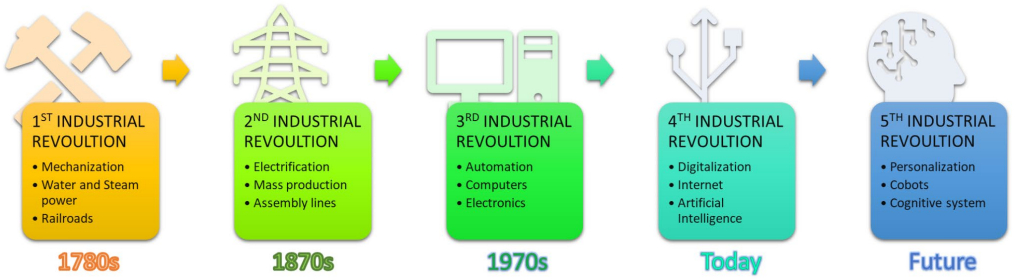

Figure 1.2: Stages of Industrial Revolution [4].

one hand, the interconnection of all the key manufacturing resources aimed at information exchange and instruction provision, on the other hand, on the systems automation), and *intellectualization* (i.e., the application of perception, analysis, decision making, and related functions based on intelligent methods and adaptive responses to changes in production requirements and manufacturing environment) [1].

Moreover, as the world struggles to fully leverage the potential of Industry 4.0, some practitioners and researchers are starting to envision the next Industrial Revolution, referred to as *Industry 5.0* [5] (Fig. 1.2). While Industry 4.0 focuses on digital connection of industrial machines to facilitate the data flow and achieve optimal performance, Industry 5.0 aims at reintroducing humans into these systems (such as through human-robot collaboration), ultimately making manufacturing processes more sustainable [11], [12] and human-friendly [13].

1.2 Monitoring and Controlling Industrial Systems in the Digital Twin Era

The real-time digitization of the production environment has emerged as a significant challenge in addressing the goals of the fourth Industrial Revolution. To this aim, *Digital Twin* (DT) has been used by research and industries as an enabling technology for developing the so-called *DT-based ISs*. In [1], a novel definition of these systems is proposed as *"smart manufacturing systems where the physical entities are paired to digital models. The system analysis, decision-making, and control are based on digital models with strong observability, representability, and computability, making the manufacturing process monitoring more transparent, the analysis more thorough, and the control more accurate"*. Compared to traditional smart ISs, DT-based ISs incorporate additional properties of *fusion* (i.e., a multi-disciplines approach with high-fidelity multi-physic models) and *interaction* (i.e., between physical and virtual entities) [14]. With these

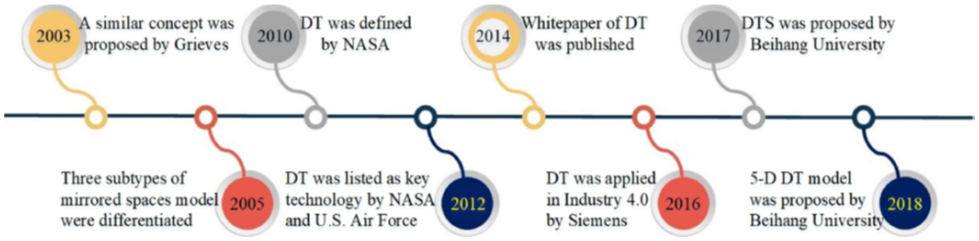

Figure 1.3: Digital Twin history timeline [18].

perception and cognitive abilities, DTs enhance automation, monitoring, and control of ISs during production more effectively than traditional approaches, providing an innovative and holistic perspective [15].

1.2.1 Basics of Digital Twin

The concept of twinning a physical model was introduced for the first time by NASA during the Apollo mission in the 1960s [16]. Nonetheless, the history of DT officially began in 2003 thanks to Michael Grieves, a researcher and Executive Director at Digital Twin Institute, who defined it as *"a virtual and digital representation equivalent to a physical product"* [17]. Then, in 2012, NASA further formalized the DT concept, defining it as *"a Multi-physics, multi-scale, probabilistic, ultra-fidelity simulation that reflects, in a timely manner, the state of a corresponding twin based on the historical data, real-time sensor data, and physical model"*. The theoretical development of DTs can be divided through three stages: *formation*, *incubation* and *growth*, as illustrated in Fig. 1.3 [18]. The *formation stage*, which spans from 2003 to 2010, saw limited publications on the topic; the *incubation stage*, ranging in period 2010-2016, saw increased interest and exploration; and, since 2016 the *growth stage* has propelled DTs from conceptual frameworks to practical applications [19].

Despite many applications of DTs are documented in the related literature, only a few delve deeply into the underlying technological aspects. A comprehensive classification is given in [20], where DTs are defined as virtual machines (or computer-based models) capable of *simulating* and *mirroring* or, *twinning*, the behaviour of any physical entity (e.g., an IS, product, or object). Therefore, a DT represents not only a simple model for simulations of the real system, but it is considered as a dynamic, intelligent, and evolving representation of a physical component [21].

To this aim, the entire life-cycle of the physical counterpart under monitoring and control needs to be continuously observed and supervised. Thus, the so-called *twinning process* operates on continuous interaction, communication, and synchronization between the physical system, its virtual model, and the surrounding environment. As a consequence, DT theory covers a large range of fields, including BD, ML, and AI, IoT, mechanics, control theory, and so on [20].

1.2.2 Models and Features

Although a unified framework has yet to be agreed upon by researchers, one of the most comprehensive architectures is proposed in [19], encompassing the following five dimensions:

- Physical Entity (PhE)
- Virtual Entity (VE)
- Data
- Connection and Integration (CN)
- Services.

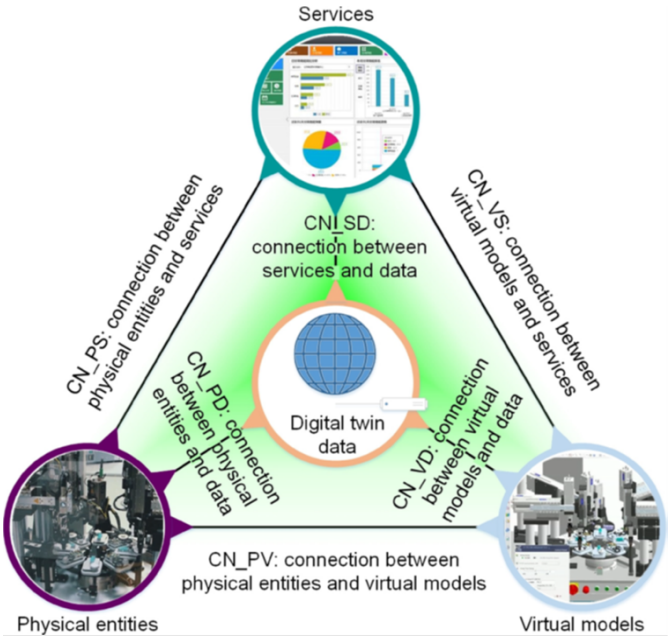

Figure 1.4: Digital Twin framework of [19] proposed by Qui *et al.* [18].

In this framework, the PhE represents the physical system (e.g., an IS or component), which is necessary for creating its corresponding VE. The latter, in turn, supports simulation, decision making, control, and monitoring (i.e., services) by exchanging data through CN tools (see Fig. 1.4).

Nonetheless, significant technological challenges remain in implementing a true DT application. The related literature mostly focuses on *data exchanges*, *modeling*, and *simulation methods*.

Connection modeling theories are useful for maintaining a constant connection between PhE, VE, and DT data. A typical connection model includes data transmission, format conversion, and data source protection. Specifically, interactions between PhE-PhE and VE-VE are discussed in [19]–[22]: PhE-PhE connection allows VE to be dynamically optimized along with PhE in real-time, while VE-VE connection considers networks of multiple VEs that share information, thus enabling multi-DT collaboration. Moreover, other types of DT connections are discussed in [18], including links between each entity previously described (i.e., PhE-DT Data, PhE-DT Services, VE-DT Data, VE-DT Services, and DT Services-DT Data). To achieve effective real-time cyber-physical synchronization, strong data optimization processes are needed to extract salient values from raw data streaming from the DT [20].

Modeling is at the core of design a DT, utilizing tools such as semantic data models and physical models. Semantic data models are trained by known inputs and outputs, and rely on AI methods. On the contrary, physical models require a comprehensive understanding of physical properties and their mutual interactions. Thus, a multi-physics modeling approach is essential for building high-fidelity DT models [23]. Nonetheless, a key issue in DT modeling lies in balancing a simplified virtual model with the complex (real) behaviour of the PhE. A practical solution uses flexible modeling approaches in a modular way. For example, authors in [24] propose to add black-box modules to the main simulation model, activating different models only as needed. To balance computational efficiency and accuracy, it is essential to identify which components are critical for the system functionality and adjust each component's modeling level accordingly, before creating a DT model [25]. Alternatively, authors in [26] propose physics-based models to achieve high precision assembly production. Overall, DT modeling usually combines physics-informed and black-box (or grey-box) approaches, merging physics and data information to develop high-fidelity DT models across various modeling levels.

Finally, DT simulation involves real-time, bidirectional interaction between the PhE and its corresponding VE. To this aim, authors in [27] present a high-level dynamic framework for efficient data exchange between heterogeneous systems. Furthermore, an anchor-point to unify data from mechatronic components across interdisciplinary domains is introduced in [28]. Differently from traditional simulations, DT-based simulations use real-time data directly acquired from the physical system [29]. Among the enabling technologies, image recognition and laser measurement are frequently recommended for capturing parameters of the PhE [30]. Multi-physics and multi-scale simulations still represent one of the most important views of DTs, emphasizing integration across different levels of detail and life-cycle phases within a holistic framework [31]. A DT enables interfacing with different models and data in a multi-scale manner while maintaining consistency. Nevertheless, most of the existing works regarding DT multi-physics simulation usually treat integration of different models as a simple input-output relationship, which, while simplified, still yields effective results [32].

1.2.3 Digital Twin for Industrial Systems

Within the Industry 4.0 context, DT represents a key enabling factor for enhancing industrial automation through holistic simulations and virtual mirroring of physical systems [33]. While technological advancements have achieved high-levels of manufacturing automation, today's manufacturing elements are increasingly context-aware, requiring the ability to communicate with their environments to make intelligent and autonomous decisions [34]. This evolution has led to numerous new concepts, paradigms, and frameworks within the field.

A DT can contribute to industrial processes throughout the entire life-cycle by supporting the following key tasks:

- *Real-time state monitoring*: by comparing simulated data from the VE with real data collected from the PhE, DTs can help identify failure modes. Historical, real-time, and predicted data are integrated to track past performance, monitor current states, and predict future outcomes with high-fidelity models. To this aim, information perception [35], augmented reality [36], and three-dimensional visualization technologies [37] assist in understanding complex situations and making optimization decisions.
- *Production control*: DTs enable intelligent and real-time control by connecting the PhE to VE, potentially reducing the reliance on a central MES [33].
- *Fault detection, diagnosis, and predictive maintenance*: combining data and physics-based methods, DTs offer cooperation of multi-physics models, enabling real-time high-fidelity environment-aware responses of the system. Key techniques in this domain include deep transfer learning [38], dynamic Bayesian network [39], and nonlinear dynamics [40].

Upon the above discussion, it is clear that DTs enable the implementation of smart control strategies, with real-time data perception and high accuracy modeling, meeting the demands of today's complex manufacturing systems [41], [42]. Moreover, in the service phase, DTs help manage decentralized products that are out of control from manufacturers and suppliers, overcoming challenges related to data access and continuous closed-loop streaming. In addition, while users prioritize product reliability, manufacturers and suppliers are mainly concerned with real-time insights for operational status and maintenance strategies [23]. DT integration with onboard sensors allows preemptive identification of degradation and potential anomalies [19].

Automatic Fault-Tolerant Control (FTC) represents an interesting way to perform closed-loop control actions for DT-based ISs. Such an approach allows process factories to continue scheduled operations even if any fault occurs before a complete failure, guaranteeing closed-loop control system robustness [43]. FTC performance depends on the design of control gain reconfiguration, which comprise *adaptive* and *Model Predictive*

Control (MPC) strategies [44]. For instance, observer-based controllers, such as those proposed in [45], enable flexible configurations using process data. Nonetheless, most of the FTC related literature considers bias and gain faults, without fully addressing apparatus failures, such as actuator malfunctions, which can render a controller uncontrollable and increase the difficulties of the control system reconfiguration. Other examples of adaptive failure compensation schemes are based on the Linear–Quadratic control for discrete-time systems with unknown actuator failures [46]. Traditional control schemes that use preset reconstruction gains to improve control performance cannot automatically adjust the control gain according to the process diagnosis results. On the contrary, most FTC-based studies aim at guaranteeing the stability of production under faulty conditions, while ignoring the security of process systems [47].

As a result, DT-based ISs offer resilience by activating self-healing responses to mitigate damage and degradation and adjusting manufacturing loads to prevent overload. Despite substantial theoretical progress, practical implementation of DTs for real-time smart control in complex manufacturing environments remains an area with significant room for development.

1.3 Data-Driven Technology as a Key Enabler for Industrial System Digitization

The previous sections provide a general description of DT foundations and describe how its key enabling technologies contribute to the development of smart ISs within the fourth Industrial Revolution era. However, traditional model-based methods are unpractical for describing the state space of complex systems, due to the exponential growth of process data.

Therefore, DD technology emerges as a potential key enabling factor for implementing DT-based ISs [48]. Most studies in this area focus on *condition monitoring* and *system diagnosis* using multivariate statistics. For example, two surveys on the fundamentals of DD methods for system monitoring cover principal component analysis, partial least squares, and generalized likelihood ratio [49], [50]. As for the system diagnosis, additional DD methods are summarized in [51], including Fisher discriminant analysis and support vector machine. Notwithstanding, these multivariate statistical analysis and ML methods often require a large dataset of multi-category fault data, which is difficult to collect; in addition, most related techniques are limited in describing the dynamical system state and are not suited for real-time diagnosis. Besides condition monitoring and system diagnosis, DD techniques can help providing also complementary tools for *process control* and *control performance maintenance*, leveraging the most cutting-edge dynamic modeling techniques for complex systems [52].

1.3.1 Data-Driven Process Control Approaches

In recent decades, model-based process control techniques (i.e., those based on the dynamical model of the plant) have gained attention, motivated by the growing availability of first-principles models and the advent of state-space control theory. Nonetheless, first-principles models are often complex and difficult to derive, thus making the selection of accurate and easy-to-use dynamical models for industrial processes challenging. This complexity has led to the widespread use of *system identification* procedures [53]. Current system identification techniques are often limited to linear and deterministic model structures. In addition, despite linearized models are often satisfying for processes when operating in small regions around steady states, the control performance can deteriorate if operation modes vary significantly. This results in employing nonlinear controllers able to handle system uncertainties [54].

To address the inherent complexities of process dynamics (e.g., nonlinearity and uncertainty), DD control strategies have recently been adopted [55], [56] also for monitoring and controlling ISs [57]. These techniques rely on the availability of large volumes of

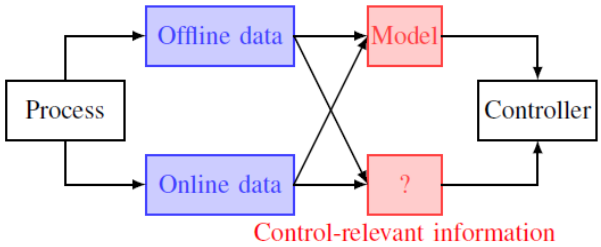

Figure 1.5: Data-Driven Process Control Strategies [52].

data collected (either *offline* or *online*) and stored from the real system. Once enough data are available, various system identification techniques are used to obtain dynamical models for control purposes. Specifically, the related literature distinguishes two different categories of DD control, namely *indirect* and *direct*. In the first case, data are completely collected offline and, then, used to identify (or estimate) a model dynamics. Conversely, the second approach considers online data collection to directly update the controller in a proper manner (e.g., direct adaptive control) [58].

A typical process can be modeled by the following nonlinear control-affine dynamics [52]:

$$\begin{cases} \dot{\boldsymbol{x}}(t) = f\big(\boldsymbol{x}(t)\big) + g\big(\boldsymbol{x}(t)\big)\boldsymbol{u}(t) \\ \boldsymbol{y}(t) = h\big(\boldsymbol{x}(t)\big) \end{cases} \tag{1.1}$$

where $\boldsymbol{x}(t) \in \mathcal{X} \subseteq \mathbb{R}^n$, $\boldsymbol{u}(t) \in \mathcal{U} \subseteq \mathbb{R}^m$, and $\boldsymbol{y}(t) \in \mathcal{X} \subseteq \mathbb{R}^p$ represent respectively the vector of n states, m inputs, and p outputs, while $f : \mathcal{X} \to \mathbb{R}^n$, $g : \mathcal{U} \to \mathbb{R}^m$, and $h : \mathcal{X} \to \mathbb{R}^p$ are smooth state, input, and output functions, respectively. Moreover, we assume that system (1.1) is controllable and observable [59]. A comprehensive overview of DD approaches for process control in systems with the form of (1.1), namely *Data-Driven Dynamic Modeling* and *Data-Driven Performance Maintenance*, is provided in [52] (see Fig. 1.5).

1.4 Thesis Positioning and Scientific Contributions

DD process control approaches guarantee high performance not only for control, but also for prediction, monitoring, diagnosing, and optimization of industrial processes. These techniques are effective for both linear and nonlinear systems in continuous and discrete-time domains, accommodating single and multiple input-output dynamics. The above-discussed state-of-the-art highlights DD process control as a promising tool for the development of DT applications in ISs, marking it as a key enabling factor of Industry 4.0. Nonetheless, several technical challenges must be addressed before these techniques can be widely applied to real-world systems.

As a result, due to the noteworthy effort needed to design effective DD approaches for IS monitoring and control, this thesis pursues two primary research directions:

1. ***Data-Driven Dynamic Modeling for Process Control:*** the first research direction investigates DD process control approaches that leverage dynamic modeling. To this aim, two different *indirect DD methods* are exploited to design optimization-based control system architectures.
2. ***Data-Driven Adaptive Control and Fault Detection for Process Performance Preservation:*** this second research direction focuses on methods for preserving performance of DD process controllers. To this aim, *adaptive control* and *fault detection* techniques are investigated.

This thesis presents the results of research published by the author across conference proceedings and journal articles. The document is structured into two main parts –described in the sequel– each corresponding to the research directions outlined above.

1.4.1 Data-Driven Dynamic Modeling for Process Control

The first level of DD process control focuses on methods based on dynamic modeling. It involves properly designing a (nonlinear) model-based controller for the underlying process once an accurate (nonlinear) dynamical model of the plant, as represented by (1.1), is identified or estimated. Several regression models are employed in the related literature for this purpose, ranging from classic Box-Jenkins' models to the more advanced models that capture nonlinear dynamics, e.g., Hammerstein-Wiener (HW), Subspace and Sparse identification methods, Koopman operator theory, and Neural networks [52], [60] (see Appendix A.1).

In this context, the first contribution of the thesis introduces process control schemes to design optimization-based regulators using dynamical models derived from system identification.

To this aim, indirect DD control methods for industrial processes are investigated in **Chapter 2**. Specifically, we propose an MPC-based control scheme for deep drawing (or cold metal forming) process. Such a process refers to a metalworking procedure aimed at getting a cold metal sheet plastically deformed in accordance to a Blank Holder Force (BHF) applied against a predefined mold. Despite being well-established in industry, this procedure is susceptible to several issues affecting the quality of the final stamped workpiece. Unlike conventional approaches, we present an MPC-based process control architecture, where the process is modeled as a Single-Input Multiple-Output (SIMO) HW model that relates the BHF to the draw-in of some critical sheet points around the die. The proposed methodology aims at improving the overall efficiency, safety, security, fault tolerance, and quality of both the stamped part and the whole forming process. The effectiveness of this control scheme is demonstrated through a DT-based software-in-the-loop (SWIL) simulation on a real case study, with performance comparisons showing significant improvements over a multiple Proportional-Integral-Derivative -based control architecture inspired by existing literature.

In **Chapter 3** we proposes a novel DD control method for nonlinear input-affine systems, a class commonly used in industrial processes. More precisely, using the Subspace Identification of Nonlinear Dynamics [61] regression technique on noisy offline data, we design a robust controller for such a class of systems. This method utilizes a basis function and consists of solving an *online* Semidefinite Programming (SDP) problem, similar to [62]. However, differently from related works based on SDP, the proposed method does not require approximations of the nonlinear dynamics, outperforming other literature-inspired controllers for strong nonlinear systems. This results in a DD control algorithm able to solve efficiently a convex optimization problem to online infer the unknown nonlinear dynamics. Moreover, closed-loop stability is explicitly ensured within the SDP problem formulation, which enforces a proper Lyapunov function's descent over time.

1.4.2 Data-Driven Adaptive Control and Fault Detection for Process Performance Preservation

Activities such as detecting and diagnosing all potential faults within the plant, controller tuning issues, or equipment malfunctioning, to prevent deterioration in controller performance, are related to preserve the performance of ISs [63]. These activities also rely on controller monitoring, which is based on real-time data retrieved during online operations. Specifically, data collected after the controller initiates online operations can be used in different ways to maintain the controller performance [64], [65]. Among these, *adaptive control* (see Appendix A.2) for parameters estimation and adaptive *fault analysis* are worth of mention.

Adaptive control methods for nonlinear systems in the form of (1.1) –especially relevant in process control applications– have been extensively studied since the 1990s [66]. When the functions f, g, and h are not known exactly, some unknown parameters must be determined online during control. To this aim, adaptation laws (i.e., governing equations for updating parameter estimates) need to be designed. Besides, DD methods are also

commonly used in *fault analysis* since existing first-principles models often fall short in fault detection and diagnosis [67]. Of course, early and accurate faults detection and analysis is crucial for ISs, leading to the use of classification problems based on statistics and ML methods [68]. This requires distinguishing between faulty and non-faulty data, and categorizing measured data into different fault categories based on this classification [69].

Consequently, the second part of this thesis aims at developing adaptive methods for both controlling and fault detection of Hydraulic Servo Actuators (HSAs), which are essential mechatronic components in many ISs.

Regarding the first objective, an Adaptive Model Predictive Control (AMPC) approach for the position tracking and force control of HSAs based on *flow control valves* is presented in **Chapter 4**. The actuator is modeled with a multiple-input multiple-output relation between the flow-rates entering the piston chambers and the piston rod position and force. Such a nonlinear dynamics is approximated, at each iteration, with a linear time-varying model through the iterative linearization approach. Thus, leveraging this approximation, the AMPC adapts its internal plant model for effectively provisioning a predicted control action on the system. The effectiveness of the proposed adaptive control architecture is shown through numerical simulation experiments on different scenarios.

In **Chapter 5**, another adaptive control framework is proposed for digital multi-chamber HSAs based on *pressure control valves*. This chapter presents a twofold contribution. First, a novel multiple-input single-output Average Equivalent Discrete-time Model based on a Pulse Width Modulation strategy is defined, inspired by power electronic converters (see Appendix A.3). Second, a two-level architecture is designed for the force tracking and pressure control. Combining a Decoupled Proportional-Integral (DPI) regulator with a decentralized Model Reference Adaptive Control, this controller provides a streamlined and practical alternative to existing methods for controlling HSAs. Our approach is applied to a four-chamber HSA and compared with an MPC-based cascaded control-loop, guaranteeing higher energy efficiency while streamlining the control architecture, even with multiple loads. In addition, we show how the proposed adaptive control strategy effectively copes with the system uncertainties, demonstrating its practical effectiveness.

As for the second objective, **Chapter 6** proposes a novel Adaptive Constrained Clustering algorithm for real-time fault detection of ISs. In fact, ISs are prone to severe faults during their work, but thanks to the widespread availability of sensors in Industry 4.0, DD methods play an important role in fault diagnosis and prognosis of critical components. The proposed algorithm clusters incoming monitoring data into groups, distinguishing nominal and non-nominal operating conditions. The solution includes a partitioned-based data stream clustering method to perform real-time fault detection. Unlike typical unsupervised methods (e.g., K-means), which do not ensure controlled clustering outcomes, our approach integrates a large volume of unlabeled data with a small amount of labeled information, adapting offline constrained clustering to the real-time scenario. Finally, simulations on an HSA used in a real industrial process demonstrate the advantages of the suggested framework in terms of responsiveness, processing speed, accuracy, and results interpretability compared to baseline methods (e.g., Stream K-means).

References

[1] Liu, S., Zheng, P., and Bao, J., "Digital twin-based manufacturing system: A survey based on a novel reference model," *Journal of Intelligent Manufacturing*, vol. 35, no. 6, pp. 2517–2546, 2024.

[2] Knapp, E. D., *Industrial Network Security: Securing critical infrastructure networks for smart grid, SCADA, and other Industrial Control Systems*. Elsevier, 2024.

[3] Babayigit, B. and Abubaker, M., "Industrial internet of things: A review of improvements over traditional scada systems for industrial automation," *IEEE Systems Journal*, vol. 18, no. 1, pp. 120–133, 2023.

[4] Raja Santhi, A. and Muthuswamy, P., "Industry 5.0 or industry 4.0 s? introduction to industry 4.0 and a peek into the prospective industry 5.0 technologies," *International Journal on Interactive Design and Manufacturing (IJIDeM)*, vol. 17, no. 2, pp. 947–979, 2023.

[5] Xu, X., Lu, Y., Vogel-Heuser, B., and Wang, L., "Industry 4.0 and industry 5.0—inception, conception and perception," *Journal of manufacturing systems*, vol. 61, pp. 530–535, 2021.

[6] Xu, L. D., "Industry 4.0—frontiers of fourth industrial revolution," *Systems Research and Behavioral Science*, vol. 37, no. 4, pp. 531–534, 2020.

[7] Raja Santhi, A. and Muthuswamy, P., "Influence of blockchain technology in manufacturing supply chain and logistics," *Logistics*, vol. 6, no. 1, p. 15, 2022.

[8] Thames, L. and Schaefer, D., "Software-defined cloud manufacturing for industry 4.0," *Procedia cirp*, vol. 52, pp. 12–17, 2016.

[9] Lu, Y., Xu, X., and Wang, L., "Smart manufacturing process and system automation–a critical review of the standards and envisioned scenarios," *Journal of Manufacturing Systems*, vol. 56, pp. 312–325, 2020.

[10] Vogel-Heuser, B., Fay, A., Schaefer, I., and Tichy, M., "Evolution of software in automated production systems: Challenges and research directions," *Journal of Systems and Software*, vol. 110, pp. 54–84, 2015.

[11] Nahavandi, S., "Industry 5.0—a human-centric solution," *Sustainability*, vol. 11, no. 16, p. 4371, 2019.

[12] Demir, K. A., Döven, G., and Sezen, B., "Industry 5.0 and human-robot co-working," *Procedia computer science*, vol. 158, pp. 688–695, 2019.

[13] Vlacic, L., Huang, H., Dotoli, M., *et al.*, "Automation 5.0: The key to systems intelligence and industry 5.0," *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 8, pp. 1723–1727, 2024.

[14] Eckhart, M. and Ekelhart, A., "Towards security-aware virtual environments for digital twins," in *Proceedings of the 4th ACM workshop on cyber-physical system security*, 2018, pp. 61–72.

[15] Dai, Y., Wang, H., Khan, F., and Zhao, J., "Abnormal situation management for smart chemical process operation," *Current opinion in chemical engineering*, vol. 14, pp. 49–55, 2016.

[16] Zhuang, C., Liu, J., and Xiong, H., "Digital twin-based smart production management and control framework for the complex product assembly shop-floor," *The International Journal of Advanced Manufacturing Technology*, vol. 96, no. 1-4, pp. 1149–1163, 2018.

[17] Grieves, M., "Digital twin: Manufacturing excellence through virtual factory replication," *White paper*, vol. 1, no. 2014, pp. 1–7, 2014.

[18] Qi, Q., Tao, F., Hu, T., *et al.*, "Enabling technologies and tools for digital twin," *Journal of Manufacturing Systems*, 2019.

[19] Tao, F., Zhang, H., Liu, A., and Nee, A. Y., "Digital twin in industry: State-of-the-art," *IEEE Transactions on industrial informatics*, vol. 15, no. 4, pp. 2405–2415, 2018.

[20] Barricelli, B. R., Casiraghi, E., and Fogli, D., "A survey on digital twin: Definitions, characteristics, applications, and design implications," *IEEE Access*, vol. 7, pp. 167 653–167 671, 2019.

[21] Tao, F., Sui, F., Liu, A., *et al.*, "Digital twin-driven product design framework," *International Journal of Production Research*, vol. 57, no. 12, pp. 3935–3953, 2019.

[22] Tao, F., Cheng, Y., Cheng, J., Zhang, M., Xu, W., and Qi, Q., "Theories and technologies for cyber-physical fusion in digital twin shop-floor," *Computer integrated manufacturing systems*, 2017.

[23] Liu, M., Fang, S., Dong, H., and Xu, C., "Review of digital twin about concepts, technologies, and industrial applications," *Journal of manufacturing systems*, vol. 58, pp. 346–361, 2021.

[24] Negri, E., Fumagalli, L., Cimino, C., and Macchi, M., "Fmu-supported simulation for cps digital twin," *Procedia manufacturing*, vol. 28, pp. 201–206, 2019.

[25] Aivaliotis, P., Georgoulias, K., Arkouli, Z., and Makris, S., "Methodology for enabling digital twin using advanced physics-based modelling in predictive maintenance," *Procedia Cirp*, vol. 81, pp. 417–422, 2019.

[26] Sun, X., Bao, J., Li, J., Zhang, Y., Liu, S., and Zhou, B., "A digital twin-driven approach for the assembly-commissioning of high precision products," *Robotics and Computer-Integrated Manufacturing*, vol. 61, p. 101 839, 2020.

[27] Schroeder, G. N., Steinmetz, C., Pereira, C. E., and Espindola, D. B., "Digital twin data modeling with automationml and a communication methodology for data exchange," *IFAC-PapersOnLine*, vol. 49, no. 30, pp. 12–17, 2016.

[28] Talkhestani, B. A., Jazdi, N., Schloegl, W., and Weyrich, M., "Consistency check to synchronize the digital twin of manufacturing automation based on anchor points," *Procedia Cirp*, vol. 72, pp. 159–164, 2018.

[29] Tan, Y., Yang, W., Yoshida, K., and Takakuwa, S., "Application of iot-aided simulation to manufacturing systems in cyber-physical system," *Machines*, vol. 7, no. 1, p. 2, 2019.

[30] Qi, Q., Tao, F., Hu, T., *et al.*, "Enabling technologies and tools for digital twin," *Journal of Manufacturing Systems*, vol. 58, pp. 3–21, 2021.

[31] Tuegel, E. J., Ingraffea, A. R., Eason, T. G., and Spottswood, S. M., "Reengineering aircraft structural life prediction using a digital twin," *International Journal of Aerospace Engineering*, vol. 2011, no. 1, p. 154 798, 2011.

[32] Boschert, S. and Rosen, R., "Digital twin—the simulation aspect," *Mechatronic futures: Challenges and solutions for mechatronic systems and their designers*, pp. 59–74, 2016.

[33] Rosen, R., Von Wichert, G., Lo, G., and Bettenhausen, K. D., "About the importance of autonomy and digital twins for the future of manufacturing," *IFAC-PapersOnLine*, vol. 48, no. 3, pp. 567–572, 2015.

[34] Zuehlke, D., "Smartfactory–from vision to reality in factory technologies," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 14 101–14 108, 2008.

[35] Zheng, Y., Yang, S., and Cheng, H., "An application framework of digital twin and its case study," *Journal of ambient intelligence and humanized computing*, vol. 10, pp. 1141–1153, 2019.

[36] Zhu, Z., Liu, C., and Xu, X., "Visualisation of the digital twin data in manufacturing by using augmented reality," *Procedia Cirp*, vol. 81, pp. 898–903, 2019.

[37] Zhao, H., Liu, J., Xiong, H., *et al.*, "3d visualization real-time monitoring method for digital twin workshop," *Computer Integrated Manufacturing Systems*, vol. 25, no. 06, pp. 1432–1443, 2019.

[38] Xu, Y., Sun, Y., Liu, X., and Zheng, Y., "A digital-twin-assisted fault diagnosis using deep transfer learning," *Ieee Access*, vol. 7, pp. 19 990–19 999, 2019.

[39] Li, C., Mahadevan, S., Ling, Y., Choze, S., and Wang, L., "Dynamic bayesian network for aircraft wing health monitoring digital twin," *Aiaa Journal*, vol. 55, no. 3, pp. 930–941, 2017.

[40] Wang, J., Ye, L., Gao, R. X., Li, C., and Zhang, L., "Digital twin for rotating machinery fault diagnosis in smart manufacturing," *International Journal of Production Research*, vol. 57, no. 12, pp. 3920–3934, 2019.

[41] Vogel-Heuser, B., Hartl, F., Wittemer, M., *et al.*, "Long living human-machine systems in construction and production enabled by digital twins: Exploring applications, challenges, and pathways to sustainability," *at-Automatisierungstechnik*, vol. 72, no. 9, pp. 789–814, 2024.

[42] Vogel-Heuser, B., Lahrsen, B., Wilch, J., Ji, F., Zhang, M., and Neumann, E.-M., "Methods to enable evolvable digital twins for flexible automated production systems," *Asia-Pacific Journal of Operational Research*, 2024.

[43] Blanke, M., Kinnaert, M., Lunze, J., *et al.*, *Introduction to diagnosis and fault-tolerant control*. Springer, 2016.

[44] Lunze, J. and Richter, J. H., "Reconfigurable fault-tolerant control: A tutorial introduction," *European journal of control*, vol. 14, no. 5, pp. 359–386, 2008.

[45] Yin, S., Luo, H., and Ding, S. X., "Real-time implementation of fault-tolerant control systems with performance optimization," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 5, pp. 2402–2411, 2013.

[46] Teixeira, A. M., Araújo, J., Sandberg, H., and Johansson, K. H., "Distributed sensor and actuator reconfiguration for fault-tolerant networked control systems," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 4, pp. 1517–1528, 2017.

[47] He, R., Chen, G., Dong, C., Sun, S., and Shen, X., "Data-driven digital twin technology for optimized control in process systems," *ISA transactions*, vol. 95, pp. 221–234, 2019.

[48] Yin, S., Li, X., Gao, H., and Kaynak, O., "Data-based techniques focused on modern industry: An overview," *IEEE Transactions on industrial electronics*, vol. 62, no. 1, pp. 657–667, 2014.

[49] Ding, S. X., Zhang, P., Jeinsch, T., Ding, E. L., Engel, P., and Gui, W., "A survey of the application of basic data-driven and model-based methods in process monitoring and fault diagnosis," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 12 380–12 388, 2011.

[50] Yin, S., Ding, S. X., Haghani, A., Hao, H., and Zhang, P., "A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark tennessee eastman process," *Journal of process control*, vol. 22, no. 9, pp. 1567–1581, 2012.

[51] Severson, K., Chaiwatanodom, P., and Braatz, R. D., "Perspectives on process monitoring of industrial systems," *Annual Reviews in Control*, vol. 42, pp. 190–200, 2016.

[52] Tang, W. and Daoutidis, P., "Data-driven control: Overview and perspectives," in *2022 American Control Conf. (ACC)*, IEEE, 2022, pp. 1048–1064.

[53] Ljung, L., "System identification," in *Signal analysis and prediction*, Springer, 1998, pp. 163–173.

[54] Su, Q., Ganesh, S., Moreno, M., *et al.*, "A perspective on quality-by-control (qbc) in pharmaceutical continuous manufacturing," *Computers & Chemical Engineering*, vol. 125, pp. 216–231, 2019.

[55] Mishra, V. K., Iannelli, A., and Bajcinca, N., "A data-driven approach to system invertibility and input reconstruction," in *2023 62nd IEEE Conference on Decision and Control (CDC)*, IEEE, 2023, pp. 671–676.

[56] Yin, M., Iannelli, A., and Smith, R. S., "Stochastic data-driven predictive control: Regularization, estimation, and constraint tightening," *IFAC-PapersOnLine*, vol. 58, no. 15, pp. 79–84, 2024.

[57] Hou, Z.-S. and Wang, Z., "From model-based control to data-driven control: Survey, classification and perspective," *Information Sciences*, vol. 235, pp. 3–35, 2013.

[58] Song, B. and Iannelli, A., "The role of identification in data-driven policy iteration: A system theoretic study," *International Journal of Robust and Nonlinear Control*, 2024.

[59] Isidori, A., *Nonlinear control systems: an introduction*. Springer, 1985.

[60] Soudbakhsh, D., Annaswamy, A. M., Wang, Y., *et al.*, "Data-driven control: Theory and applications," in *2023 American Control Conf. (ACC)*, IEEE, 2023, pp. 1922–1939.

[61] Fasel, U., Kaiser, E., Kutz, J. N., Brunton, B. W., and Brunton, S. L., "Sindy with control: A tutorial," in *2021 60th IEEE Conf. on Decision and Control (CDC)*, IEEE, 2021, pp. 16–21.

[62] Dai, T. and Sznaier, M., "Nonlinear data-driven control via state-dependent representations," in *2021 60th IEEE Conf. on Decision and Control (CDC)*, IEEE, 2021, pp. 5765–5770.

[63] Gao, X., Yang, F., Shang, C., and Huang, D., "A review of control loop monitoring and diagnosis: Prospects of controller maintenance in big data era," *Chinese Journal of Chemical Engineering*, vol. 24, no. 8, pp. 952–962, 2016.

[64] Jiang, Y., Yin, S., and Kaynak, O., "Data-driven monitoring and safety control of industrial cyber-physical systems: Basics and beyond," *IEEE Access*, vol. 6, pp. 47 374–47 384, 2018.

[65] Zhong, K., Han, M., and Han, B., "Data-driven based fault prognosis for industrial systems: A concise overview," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 2, pp. 330–345, 2019.

[66] Annaswamy, A. M. and Fradkov, A. L., "A historical perspective of adaptive control and learning," *Annual Reviews in Control*, vol. 52, pp. 18–41, 2021.

[67] Venkatasubramanian, V., Rengaswamy, R., Kavuri, S. N., and Yin, K., "A review of process fault detection and diagnosis: Part iii: Process history based methods," *Computers & chemical engineering*, vol. 27, no. 3, pp. 327–346, 2003.

[68] *Fault detection and diagnosis in industrial systems*. Springer Science & Business Media, 2000.

[69] Qin, S. J., "Survey on data-driven industrial process monitoring and diagnosis," *Annual reviews in control*, vol. 36, no. 2, pp. 220–234, 2012.

Part I:

Data-Driven Dynamic Modeling for Process Control

Chapter 2

Deep Drawing Process Control Based on Indirect Data-Driven Approaches

Abstract

The *deep drawing* (or metal forming) process is a metalworking procedure aimed at getting a cold metal sheet plastically deformed in accordance with a predefined mould. Although this procedure is well-established in industry, it is still susceptible to several issues affecting the quality of the final stamped part. In order to reduce defects of workpieces, process control approaches can be performed. Typically, such methods employ simple Proportional-Integral-Derivative (PID) regulators for steering the Blank Holder Force (BHF), tuned through trial and error procedures based on the punch force error evaluation. Thus, resulting in non efficient process control strategies. Besides, a single PID cannot handle constraints on the process variables. Therefore, differently from the state of the art, in this chapter we propose a process control architecture for deep drawing based on an *indirect Data-Driven approach*. Specifically, we employ offline data to identify such a metal forming process by a Hammerstein-Wiener model, resulting in a single-input multiple-output dynamics that relates the BHF with the draw-in of n different critical points around the die. Thus, upon this identified dynamics, a Model Predictive Control (MPC) framework is designed for the sake of avoiding the workpiece defects due to the abnormal sliding of the metal sheet during the forming stage. Finally, the effectiveness of the proposed process controller is shown on a real case study through a software-in-the-loop simulation based on a Digital Twin, where the performance achieved by the MPC-based system is analyzed in detail and compared against the results obtained through an ad-hoc defined multiple PID-based control architecture.

Contents

2.1 Introduction

Deep drawing (or metal forming) is a metalworking process used to impress a predefined shape to a metal sheet. The deformation of the metal is performed at re-crystallization temperature, which is usually equal to the ambient one for many metals [1]. In such a type of forming process, the metal sheet gets deformed plastically while it is pushed down into the die cavity, conforming it to its shape [1], and with no alterations in its thickness [2]. A schematic representation of a typical stamping press used for cold sheet metal forming is shown in Fig. 2.1. It consists of a *punch* (or *upper die*), a *die*, and a *blank holder*, which holds the metal sheet in place during the punch stroke (i.e., while the punch is lowered into the die). The blankholder design can also include appropriate *drawbeads* to help regulate the material flow into the die. Deep drawing is typically characterized

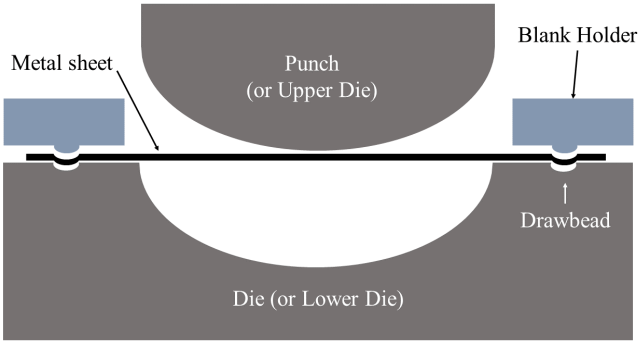

Figure 2.1: Schematic representation of a typical stamping press configuration.

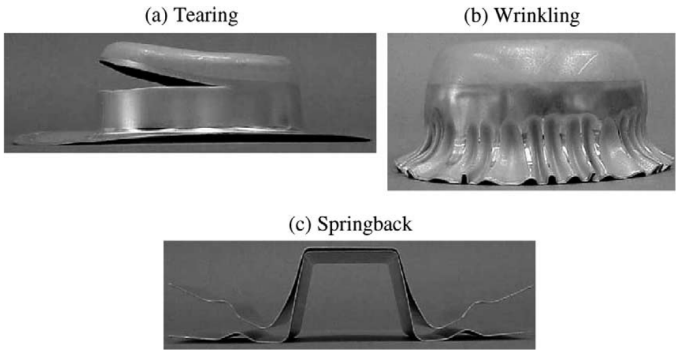

Figure 2.2: The most common problems in the cold metal sheet forming [6].

by high pressures (~ 10 kPa), high speeds (up to 8 mm/s), and very short durations (less than 5 s). This results in high production rates, low labour costs, but difficulties in effectively controlling in real time the process [3]–[5].

Several quality issues affect the stamped metal products: the most common problems include *wrinkling* (due to compression stresses), *tearing* (due to tensile stresses), and *springback* (due to elasticity) [6], as illustrated in Fig. 2.2. During the forming phase, the occurrence of the listed defects can be identified by monitoring specific forming variables, i.e., the punch force, wrinkle height, draw-in, and friction force, which are all influenced by the Blank Holder Force (BHF). In effect, the workpiece quality depends on how much the BHF presses on the blank holder: if it is too high, then the flow of material into the die is restricted and tearing is likely to occur in that region; if the BHF is too low, then excessive material flow can lead to wrinkling. Moreover, the BHF plays a crucial role in controlling springback, since it can change the internal force and metal flow resistance of a plate [7]. In fact, springback is due to deformations of the material in the elastic-plastic field. This is a critical aspect of the drawing process, especially in the automotive sector, where high accuracy is often required [2].

Due to the presence of the above described issues, the industrial deep drawing process is lagging behind in the full implementation of the zero-defect manufacturing paradigm. In fact, traditional methods used to pragmatically enhance the products quality typically consist in batch checking the stamped parts features. In case of defects in the examined part, the entire batch is backward checked, with consequent loss of time and a high number of defective parts. More recent and innovative approaches in Industry 4.0 consider the application of automatic control techniques [8], [9], which can be classified into *machine control* and *process control* [10], [11]. The first class of techniques aims at monitoring and controlling the press variables (e.g., feedback and control of the BHF, imposing the tracking of a predetermined trajectory), resulting in an open-loop control with respect to the process features, whereas the latter class aims at monitoring the forming parameters of the process and controlling the machine variables (e.g., feedback of the punch stroke

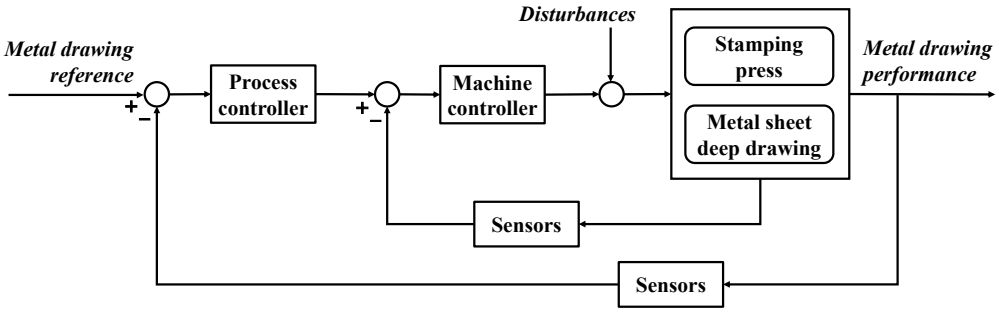

Figure 2.3: Block diagram of a typical process control closed-loop.

and control of the BHF), thus resulting in a closed-loop control of the process parameters. Differently from machine control, process control can allow to online monitor the state of the stamped part during the formation phase and to provide corrective actions in case of abnormal behaviors of the metal sheet, thus ensuring a more precise and higher quality control of the process [10]. As depicted in Fig. 2.3, machine and process control can be combined. Thus, the machine control feedback loop can be nested inside the process control one. In this manner, the process controller performs as a high-level control and provides a very important contribution to disturbance rejection.

In this context, the goal of this chapter is to provide a process control methodology for deep drawing, exploiting *indirect Data-Driven* (DD) approaches. Therefore, once an appropriate dynamics of the deep drawing process is identified through offline data, a well-defined model-based controller is properly designed. Differently from the state of the art, on the one hand, we model the forming stroke with a Single-Input Multiple-Output (SIMO) *Hammerstein-Wiener* (HW) model, providing a relationship between the BHF (i.e., the single input) and the draw-in of n different monitored points (i.e., the multiple outputs) around the die. On the other hand, we aim at proposing a *Model Predictive Control* (MPC) methodology that can provide higher performance with respect to standard controllers. The effectiveness of the proposed process control approach is shown on a real case study for a "T-shaped" metal part (within the national research program PICO&PRO, whose activities have been carried out in collaboration with both some industries and research centers). The results are first validated through a simulation setup based on *Digital Twin* (DT), where the performance achieved by the MPC-based system is analyzed in detail and compared against the results obtained through an ad-hoc designed multiple Proportional-Integral-Derivative (PID)-based control architecture. Then, the proposed approach is further tested in a real scenario, in collaboration with the SMATI (Materials and Innovative Technology) Research Group and Telematics LAB of the Politecnico di Bari, CRF (Centro Ricerche Fiat), and Gigant Company Srl. demonstrating the effectiveness on a real metal sheet.

2.2 Literature Review and Chapter Contribution

In order to properly design an effective process controller for the deep drawing process, [12] indicates two key specifications to satisfy. The former regards modeling the deep drawing process taking into account model uncertainties and disturbances. This means to properly define a mathematical correlation between the selected input and output process variables. It is also important to consider that the model and its uncertainties are strictly correlated to the specific case study, i.e., depending on material properties, workpiece shape, press size, and so on. Thus, modeling uncertainties means including small variations in the forming system (such as in blank size and thickness, material properties, and tooling) and external input to the forming system (such as lubrication). The second considers process controller performance. Thus, the design should guarantee high tracking guarantees regardless of uncertainties and process disturbance, resulting in

satisfying disturbance rejection requirements. Therefore, the literature review presented in the sequel is focused on two main aspects of the deep drawing process control, i.e., models and procedures used for the system identification, as well as the techniques adopted for process control, with particular attention to the application of MPC to metal forming.

2.2.1 Deep Drawing Process Modeling

As it emerges in [13], in deep drawing most of forming variables directly depend on the BHF values during the forming phase. The typical process variables indicated in literature are *wrinkling*, *punch force*, and *draw-in* [1]. The ability to sense the occurrence of wrinkles is potentially useful; however, its measure is limited, because wrinkles locations are not known a priori. Differently, the monitoring of the punch force is more viable and various contributions model the deep drawing process relating the BHF to the punch force [6], [12], [14], [15]. Nevertheless, it is proven that the draw-in of the metal sheet can better reveal the correct execution of the deep drawing process [2], [7]. In fact, more recent works consider the deep drawing process as a relationship between the BHF as input and the draw-in as output. In the related works, the authors aim at obtaining the optimal profile for the BHF and the draw-in of the material via off-line multi-variable optimization. The obtained results can then be used as reference signals for process control.

As for deep drawing modeling, two main techniques are viable. On the one hand, the process can be represented considering physics-informed modeling approaches. These can be particularly reliable, but also complex and time consuming, due to the non linear nature of the process. On the other hand, the process can be considered as a black box and DD methods can be applied to identify, resulting in linear or nonlinear regression. However, system modeling could be less precise then physics-based procedures, but also simpler and faster, which are valuable features in the industrial sector. Moreover, various assessment tools are available to compare and test the identified models, including Akaike's Information Criterion (AIC), Akaike's Final Prediction Error (FPE), and Rissanen's Minimum Description Length (MDL) [16] (see Appendix A.1).

Although system identification is widely used in manufacturing (see e.g., [17]–[19]), to the best of our knowledge, it has never been adopted for deep drawing processes. Several examples of physics-based modeling approaches are proposed in the related literature [6], [12], [14], [15], where nonlinear Single-Input Single-Output (SISO) first-order dynamical models are obtained for relating the BHF and the punch force. Conversely, within the nonlinear identification context, among several types of nonlinear regression models, Nonlinear Auto Regressive eXogenous (NARX) and HW models are largely appreciated for the identification of nonlinear systems [20]–[23] (see Appendix A). The former represent one of the most traditional nonlinear input-output models, whereas HW are composed by linear and nonlinear subsystems. Since, the forming process performs a plastic deformation of the material, the workpiece deformation is almost linear until its fracture. This means that nonlinearities do not influence the process dynamics under certain conditions. Hence, HW models can be suitably employed to simplify the representation procedure.

2.2.2 Deep Drawing Process Control

Process control aims at improving the overall process performance in the presence of disturbances and ensuring good part quality while preventing process failures [13], [24]–[26]. In general, process control architectures present a feedback control loop for monitoring online the press variables based on the error between process variables and the related reference values. In this context, [14], [15] proposes the use of PID controllers. In [6], authors consider a proportional-integral control with a feed-forward action. Similarly, [27] suggests the use of standard controllers tuned by an automatic controller and some model reference adaptive control approaches. Although the results achieved by the aforementioned works are satisfactory in terms of reference tracking and stamped part quality (also in presence of disturbances), these approaches suffer from the main drawbacks of standard regulators. PID controllers –widely used in several industrial

control applications– constitute a good and cheap choice, but they are able to address only simple SISO dynamics. Thus, the application of PID control to multi-variable systems requires the definition of ad-hoc control architectures. Moreover, since PIDs cannot properly handle constraints on the process variables, the tuning phase could be difficult, being heavily dependent on user experience and competency. Thus, the tuning activity can require a large number of tests, often producing sub-optimal control strategies [28].

For the sake of overcoming the above-discussed limits, different strategies have been developed in the related literature for the deep drawing process control. In particular, more recently thanks to the increase of computing power and the development of novel sensing and actuating technologies, optimal feedback control has gained more attention. For example, [29] introduces a process optimal control method for the forming monitoring to reduce the springback effect, based on evolutionary strategy. Furthermore, an optimal control of the BHF by monitoring the draw-in of the metal sheet during the deep drawing phase is proposed in [30]. the authors use a nonlinear model based on the Finite Element Method to model the forming process, and a nonlinear least square optimization algorithm to design the control system.

In more recent works, a further evolution of optimal control is presented in [13], i.e., a MPC-based architecture. Specifically, differently from optimal feedback control, MPC considers the use of a model of the system during the online control to predict the system behavior and compute the most appropriate control actions while optimizing a proper performance indicator. The application of MPC to the deep drawing feedback control presents various advantages. First, MPC is a multi-variable real-time control strategy that allows driving the outputs of the system by taking into account the interactions between system variables and variables constraints. Then, assuming the availability of a good process model, the MPC tuning phase can be carried out in an intuitive and simple way, not necessarily by an expert user. For instance, [31], [32] propose a linear MPC-based process controller to monitor the step depth of the workpiece geometrical contour tool-path in incremental sheet forming. The aim is to reduce the pillow effect, sheet bending effect, and springback.

2.2.3 Chapter Contribution

In view of the above discussion, the goal of this chapter is to present an indirect DD process control architecture based on MPC for the proactive real-time control of deep drawing process.

To this aim, on the one hand, a multi-variable HW model is obtained for describing the deep drawing process. Differently from the state of the art, that generally proposes SISO models putting in relation the BHF and the punch force only, we consider a more complete Single-Input Multiple-Output (SIMO) HW dynamical model that relates the BHF (as the single input) and the draw-in at n different locations around the die (as the multiple output), whose monitoring can reveal the eventual presence of defects and cracks in the whole stamped part. The obtained model is identified through offline data, and ensures a good compromise between the process controller design easiness and the system representation accuracy. Particularly, the latter allows an effective evaluation of the performance of the designed process controller by means of simulation tests. On the other hand, the MPC-based approach allows the strict fulfillment of the requirements on process variables, thus overcoming the issues previously discussed. Moreover, the computational time of the MPC is sufficiently limited and compatible with the short duration of the deep drawing process.

2.3 The Proposed Deep Drawing Process Control

In this section we present the proposed DD process control approach for deep drawing. We use offline data to identify the dynamics with a HW model having the BHF as input and the draw-in of n critical points of the metal sheet profile around the die as output.

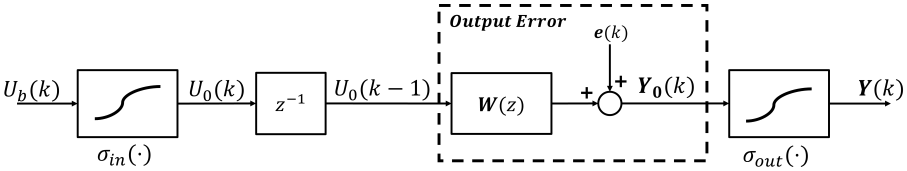

Figure 2.4: Block diagram of the Hammerstein-Wiener model representing the nonlinear dynamical process.

Subsequently, we provide the definition of an MPC controller for the considered deep drawing process.

2.3.1 Process Model Identification

As discussed in the previous section, the draw-in is generally used to extract meaningful information on the state of the forming part. Consequently, we model the deep drawing process as a black box having the BHF (uniformly distributed on the blank holder) as the input variable – denoted as U_b – and the draw-in of n critical points of the metal sheet profile as the output variables – denoted as $\boldsymbol{Y} = (Y_1, \ldots, Y_n)^\top$ –, thus assuming the system as a SIMO system. In particular, we represent its dynamical behavior by means of a discrete-time nonlinear model, which allows a more precise modeling of the process with respect to possible simplified linearized modeling approaches. More in detail, in this section we propose the use of the HW model, which describes nonlinear dynamical systems using a linear time-invariant dynamical subsystem sandwiched between two static nonlinear subsystems as illustrated in Fig. 2.4.

The linear SIMO block is represented by a transfer matrix (indicated by $\boldsymbol{W}(z)$ in Fig. 2.4), whilst the input and output non-linearities (indicated by functions $\sigma_{in}(\cdot)$ and $\sigma_{out}(\cdot)$ in Fig. 2.4) are modeled with polynomial estimators:

$$U_0(k) = \sigma_{in}\big(U_b(k)\big) \tag{2.1a}$$

$$\boldsymbol{Y}(k) = \sigma_{out}\big(\boldsymbol{Y}_0(k)\big) \tag{2.1b}$$

where $U_0(k)$ and $\boldsymbol{Y}_0 = (Y_{01}, Y_{02}, \ldots, Y_{0n})^\top$ are the input and the outputs vector of the SIMO linear subsystem. Note that the transfer matrix $\mathbf{W}(z)$ has the following form:

$$\mathbf{W}(z) = \begin{pmatrix} \mathrm{W}_{11}(z) \\ \mathrm{W}_{21}(z) \\ \vdots \\ \mathrm{W}_{n1}(z) \end{pmatrix} \tag{2.2}$$

where each element represents the scalar transfer function of the SISO system described by the BHF as input and the draw-in profile of each critical point as output. Alternatively, the SIMO linear subsystem can be reformulated in accordance with the state-space representation:

$$\begin{cases} \boldsymbol{X_0}(k+1) = \mathbf{A}\boldsymbol{X_0}(k) + \mathbf{B}U_0(k) \\ \boldsymbol{Y}_0(k) = \mathbf{C}\boldsymbol{X_0}(k) + \mathbf{D}U_0(k) \end{cases} \tag{2.3}$$

where $\mathbf{A} \in \mathbb{R}^{n_p \times n_p}$, $\mathbf{B} \in \mathbb{R}^{n_p \times 1}$, $\mathbf{C} \in \mathbb{R}^{q \times n_p}$, and $\mathbf{D} \in \mathbb{R}^{n_p \times 1}$ are respectively the state, input, output, and input-output matrix, while $\boldsymbol{X}_0 = \big(\boldsymbol{X}_1^\top, \boldsymbol{X}_2^\top, \ldots, \boldsymbol{X}_n^\top\big)^\top$ is the state variable vector whose components are $\boldsymbol{X}_1 = \big(X_{11}, X_{12}, \ldots, X_{1p}\big)^\top$, $\boldsymbol{X}_2 = \big(X_{21}, X_{22}, \ldots, X_{2p}\big)^\top$, $\ldots$, $\boldsymbol{X}_n = \big(X_{n1}, X_{n2}, \ldots, X_{np}\big)^\top$ with p as the order of the HW model.

The model identification procedure requires various preliminary data management steps for a proper data life-cycle, consisting of *Data acquisition*, *Data pre-processing*, and *Data analysis*. Firstly, different experiments must be carried out on the press and, during the forming stroke, real data must be collected, analyzed, integrated, and visualised by

using some data management techniques. During the *Data acquisition*, the BHF, i.e., the input variable, and the draw-in at n different points around the die, i.e., the output variables, are collected and stored. We highlight that critical points can change in number and position depending on the shape to be formed. The data retrieved from the press are organized in different datasets suitable for the subsequent steps. Once all data is gathered, following the approach suggested by Baethi *et al.* in [33] and Tao *et al.* in [34], a *Data pre-processing* activity must be performed. It consists in the detection of outliers for the data cleaning, a cubic spline interpolation [35] for the data oversampling, and a filtering step by using the Savitzky-Golay filter [36] for the data smoothing. Finally, the *Data analysis* is to be implemented to extract important information from time-series and make them useful by using some visualization data plot and graphics [37].

Finally, we remark that, at the end of the identification procedure, the correctness of the model is proved by simulation, monitoring both its response to the input signal and its output error with respect to the real one. In particular, this phase is also exploited to compare HW models with a different order p, so that the best performing one is chosen to represent the considered deep drawing process.

2.3.2 MPC Design

The proposed MPC-based control system block diagram is reported in Fig. 2.3. It considers a single MPC block that simultaneously controls the draw-in of the n blank critical points by manipulating the BHF uniformly applied on the whole blank surface. At each sample time, the controller computes the Control Variables (CVs), the State Variables (SVs), and the Output Variables (OVs) related to the given control horizon H_u and prediction horizon H_p respectively in the vector $\boldsymbol{u}(t) := \big(U_b(t+1)^\top, \dots, U_b(t+H_u)^\top\big)^\top$, $\boldsymbol{x}(t) := \big(\boldsymbol{X}(t+1)^\top, \dots, \boldsymbol{X}(t+H_p)^\top\big)^\top$, and $\boldsymbol{y}(t) := \big(\boldsymbol{Y}_i(t+1)^\top, \dots, \boldsymbol{Y}_i(t+H_p)^\top\big)^\top$. The values of CVs and OVs are subject to physical limitation, as indicated by the following constraints related to time step k:

$$\mathrm{U}_b^{\mathrm{lb}} \leq U_b(k) \leq \mathrm{U}_b^{\mathrm{ub}} \tag{2.4a}$$

$$\mathrm{Y}_i^{\mathrm{lb}}(k) \leq Y_i(k) \leq \mathrm{Y}_i^{\mathrm{ub}}(k), \quad \forall i = 1, \dots, n \tag{2.4b}$$

where $\mathrm{U}_b^{\mathrm{lb}}$ and $\mathrm{Y}_i^{\mathrm{lb}}(k)$ and $\mathrm{U}_b^{\mathrm{ub}}$ and $\mathrm{Y}_i^{\mathrm{ub}}(k)$ denote the lower and upper bounding for the CVs and the OVs, respectively. The MPC block has the following objective function:

$$\boldsymbol{J}\big(\boldsymbol{u}(t), \boldsymbol{y}(t)\big) = \boldsymbol{J}_{\boldsymbol{y}}\big(\boldsymbol{u}(t), \boldsymbol{y}(t)\big) + \boldsymbol{J}_{\boldsymbol{\Delta u}}\big(\boldsymbol{u}(t), \boldsymbol{y}(t)\big). \tag{2.5}$$

Each term in (2.5) includes weights (w_i^y and $w^{\Delta u}$) used to balance the competing objectives and described in the sequel. Moreover, Kalman Filter is used as state observer to update the model dynamics at each time step. $\boldsymbol{J}_{\boldsymbol{y}}(\boldsymbol{u}(t), \boldsymbol{y}(t))$ is the term referring to the output reference tracking and it is defined as follows:

$$\boldsymbol{J}_{\boldsymbol{y}}(\boldsymbol{u}(t), \boldsymbol{y}(t)) = \sum_{k=t}^{t+H_p-1} \left(\sum_{i=1}^{n} w_i^y \big(Y_i(k+1) - Y_i^{\mathrm{ref}}(k+1)\big)^2 \right).$$

$\boldsymbol{J}_{\boldsymbol{\Delta u}}\big(\boldsymbol{u}(t), \boldsymbol{y}(t)\big)$ is the term used for the move suppression controlled variable. During the execution, small CV adjustments (moves) are computed. The MPC controller uses the following scalar performance measure for move suppression controlled variable:

$$\boldsymbol{J}_{\boldsymbol{\Delta u}}\big(\boldsymbol{u}(t), \boldsymbol{y}(t)\big) = \sum_{k=t}^{t+H_u-1} \Big(w^{\Delta u} \big(U(k) - U(k-1)\big)^2 \Big).$$

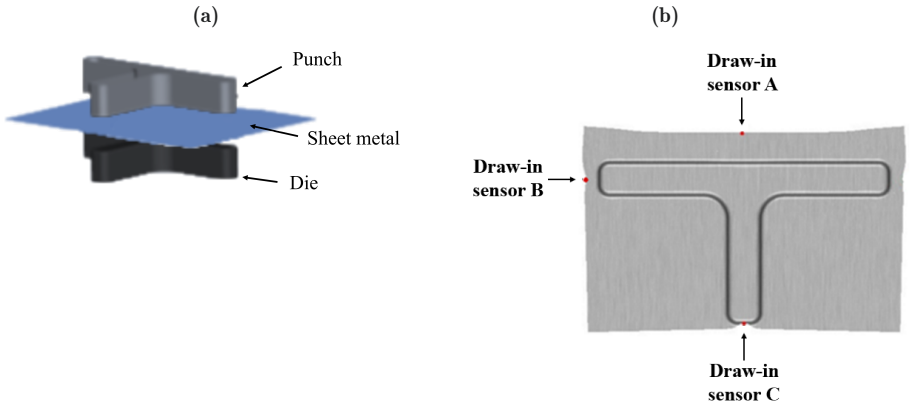

Figure 2.5: "T-shape" steel component: deep drawing mould (a) and mechanical scheme indicating the critical points A, B, and C (b).

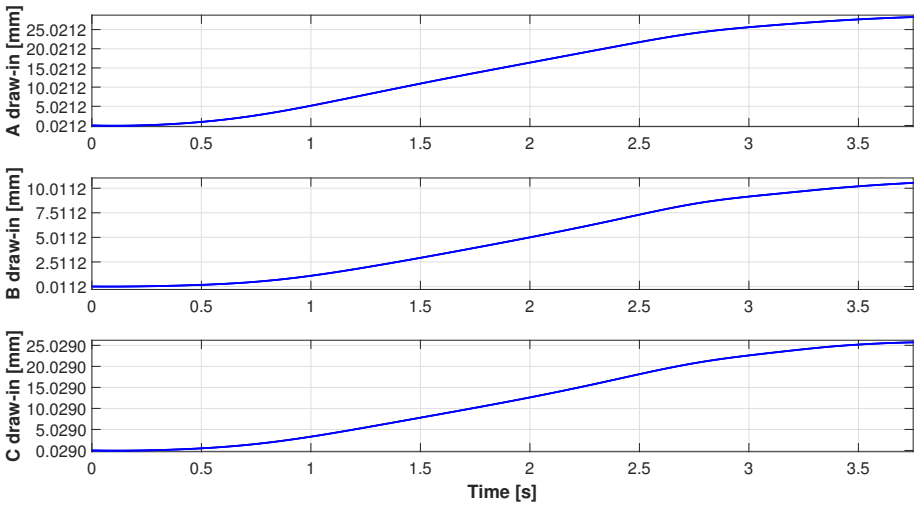

Figure 2.6: Draw-in of critical points A, B, and C over time during the forming stroke under the optimal conditions.

Hence the following optimization problem is solved by the MPC regulator:

$$\min_{\boldsymbol{u}(k),\,\boldsymbol{y}(k),\,\boldsymbol{x}(k)} \quad \boldsymbol{J}\big(\boldsymbol{u}(t),\boldsymbol{y}(t)\big) \tag{2.6a}$$

$$\text{s.t.} \begin{cases} (2.1a), (2.1b), (2.3) & \forall k = t+1,\dots,t+H_p-1 \\ (2.4a) & \forall k = t+1,\dots,t+H_u \\ (2.4b) & \forall k = t+1,\dots,t+H_p. \end{cases} \tag{2.6b}$$

More precisely, at each sampling time k, the controller obtains all measurements $\boldsymbol{y}(k)$, computes the required CVs $\boldsymbol{u}(k)$ solving (2.6), and applies to the plant only the first obtained sample.

2.4 Case Study

In this section we apply the proposed MPC-based approach to a real case study referred to the manufacturing of the "T-shape" workpiece shown in Fig. 2.5. This is a structural component of a car bodywork which is realized in HR 440Y580T-FB-UC steel (2 mm thick) and produced at the Gigant company (Bologna, Italy) for Fiat cars.

Table 2.1: Servo-hydraulic press work-cycle: steps and corresponding duration.

	Step	**Duration** [s]
	Approach stroke	2.39
	Fluid compression	0.42
	Deep drawing	3.88
	Fluid decompression	0.26
	Ascent stroke	2.70
Tot. time		9.66

Table 2.2: Technical specification of the hydraulic piston.

Parameter	**Unit**	**Value**
N. of hydraulic actuators	-	2
Max. force per single actuator	kN	1500
Sensitivity	kN	± 60
Response time	ms	100
Speed	m/s	8

2.4.1 Setup of Process Control Experiments Based on Digital Twin

The metal forming of the considered component is currently performed using a servo-hydraulic press equipped with a machine controller. The considered press is composed by a sled and a passive pressboard. The sled, surmounted by the upper die, is moved both by a hydraulic axle and an electric axle. Both axles are moved by two actuators: hydraulic pistons are employed for the former, and electric motors for the latter. During the deep drawing phase, only the upper die is pushed down and the BHF acts on the draw-in of the metal sheet. The corresponding complete work-cycle includes five stages (i.e., approach stroke, fluid compression, deep drawing, fluid decompression, and ascent stroke), whose durations are reported in Table 2.1. The core step of the whole forming process is deep drawing: during this phase, the punch runs downwards of 30 mm providing a stroke at a speed of 8 mm/s. The total load, provided by each single hydraulic actuator, remains constant. In this step only the hydraulic axle works, but the electric axle can allow a force adjustment within a small range of values, overlapping with the force transmitted by hydraulic pistons to the sled. Table 2.2 shows the technical specifications of the hydraulic piston utilized as actuator during the deep drawing phase. Since the electrical axis can work during this step (if necessary), the maximum force available to the sled is equal to $(2 \times 1500) + 280$ kN $= 3280$ kN.

In the framework of the PICO&PRO research project, the press is being enhanced by adding a process controller. To this aim, on the one hand, the press has been equipped with three laser displacement sensors aimed at measuring in real-time the draw-in of the metal sheet edge in the critical points A, B, and C illustrated in Fig. 2.5 (b), during the stamping process. Note that the choice of these points is the result of a robust off-line multi-variable optimization based on stamping experiments and numerical simulations conducted on the addressed process. As an additional outcome of this analysis, the optimal draw-in profiles for the critical points A, B, and C are determined in reference to the optimal BHF – denoted as U_b^{ref} – equal to 588 kN (see Fig. 2.6). On the other hand, the process controller has been designed in accordance to the methodology described in Section 2.3 and has been tested in a DT framework described in the sequel. The sampling time is set to 200 ms. Note that the detailed features of the identified model are described in Appendix 2.A, whilst the performance of the designed MPC controller as well as the

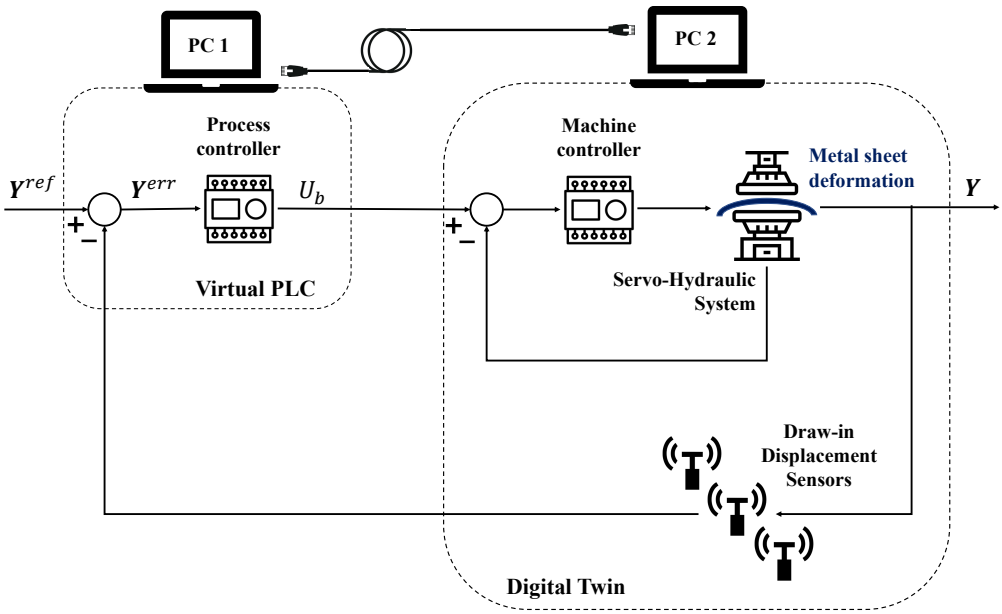

Figure 2.7: Software-in-the-loop framework used for the closed-loop MPC-based control system simulation.

values of parameters are reported in Appendix 2.B.

In the Industry 4.0 framework, the use of the DT is widely adopted, especially in the manufacturing and automotive context [38], [39]. In effect, the development of a cyber-physical system in addition to the deployment of a complex network of DT interacting with monitored and controlled elements is an important key enabling technology for the optimal design, commissioning, and maintenance of stamping presses [40]. In fact, especially in the context of production systems commissioning, the DT is commonly used to leverage on the benefit derived from the recent concept of Virtual Commissioning (VC). VC enables the full validation and verification of a complex manufacturing system by performing a simulation involving a virtual plant and a real controller, thus reducing the time to market, lowering costs, and increasing productivity [41]. Simulations based on a software-in-the-loop (SWIL) approach are generally used to virtually test the control system before the physical implementation of the plant is finished [42], [43]. Following this trend, in this case study a SWIL simulation -based architecture is employed to test the proposed closed-loop control system, using the DT of the press instead of its physical implementation. Figure 2.7 shows the complete architectural framework implemented for the SWIL simulation of the deep drawing control system. In particular, the designed MPC algorithm is implemented in a virtual Programmable Logic Controller (PLC) that is in turn deployed on the PC1 digital platform: here the process controller runs, computing the optimal BHF and communicating such a control action (i.e., U_b) to the inner machine control loop. Viceversa, the PC2 digital platform is dedicated to the DT of the press including the digital models of the servo-hydraulic system, the deep drawing process (i.e., the metal sheet that is getting deformed), the machine controller, and the three draw-in displacement sensors. The communication between PC1 and PC2 is based on the Open Platform Communications Unified Architecture (OPC UA), whilst the communication between PC1 and the DT of the sensors is managed by the Transmission Control Protocol/Internet Protocol (TCP/IP). The PLC used for the virtual simulation is the Siemens S7-1500 with CPU 1515T-2 PN, programmed in the TIA Portal V15 environment, while the virtual commissioning of the PLC is performed using the S7-PLCSIM Advanced V3.0 software [44].

Table 2.3: MPC-based control system computational and communication performance over Monte Carlo simulations (values expressed in [ms]).

	T_s	TCP/IP delay	MPC computation time	OPC UA delay
Mean	213.1	29.1	1.8	9.5
Maximum	215.8	34.0	2.8	15.1

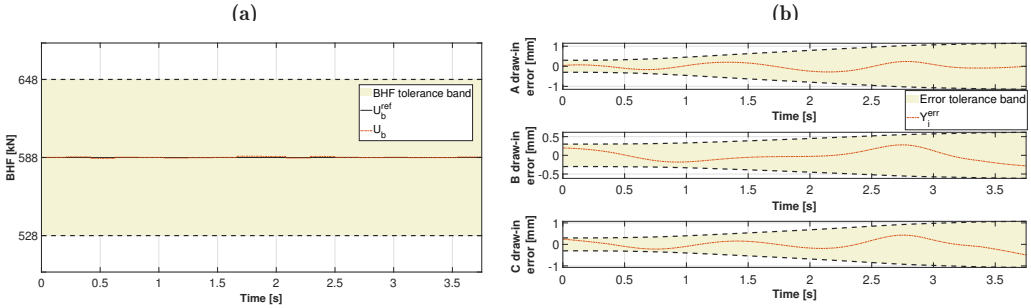

Figure 2.8: MPC-based control system results: BHF profile vs. bounding range (a); draw-in error profile of critical points A, B, and C vs. tolerance band (b).

2.4.2 Process Control Result

Through the DT-based SWIL simulation framework shown in Fig. 2.7, the MPC-based closed-loop control system tuned as described in Appendix 2.B is simulated. The effectiveness of the proposed control method is tested considering the closed-loop system reference trajectory tracking capability. Thus, the BHF error with respect to the optimal value $\mathrm{U}_b^{\mathrm{ref}}$ (i.e., 588 kN) is obtained as $U_b^{\mathrm{err}} = (\mathrm{U}_b^{\mathrm{ref}} - U_b)$, while the draw-in error with respect to the reference value $\mathbf{Y}^{\mathrm{ref}}$ is obtained as $\boldsymbol{Y}^{\mathrm{err}} = (\mathbf{Y}^{\mathrm{ref}} - \boldsymbol{Y})$. In all simulations, external process disturbances are taken into account by considering a step disturbance affecting the control input and characterized by an amplitude equal to 4% of the reference value $\mathrm{U}_b^{\mathrm{ref}}$.

The experimental results are reported in Fig. 2.8. In particular, Fig. 2.8 (a) shows the reference trajectory tracking capability, while Fig. 2.8 (b) the output error trajectory tracking. More in detail, the components of the error vector $\boldsymbol{Y}^{\mathrm{err}}$ (i.e., Y_i^{err} for $i = \{1, 2, 3\}$ indicating the error between the actual and the reference draw-in at the critical points A, B, and C) get the mean values of 0.9%, 0.8%, and 0.7%, respectively. The BHF reference trajectory tracking error U_b^{err} is 0.1%. It is worthwhile noting that the obtained draw-in errors, indicated by dotted curves in Fig. 2.8 (b), are inside the tolerance band, represented by shaded area bounded by the dashed lines in Fig. 2.8 (b), thus ensuring that the stamped workpiece is defect free, despite the presence of the modeled external disturbances. A Monte Carlo simulation with 1000 runs is carried out in order to validate the runtime performance of the deployed control system in the SWIL framework described above. In reference to a single sampling period, the mean and maximum values of the main computational and communication efforts (i.e., the actual sampling time, the MPC algorithm elaboration time, the TCP/IP, and OPC UA communication delays) are reported in Table 2.3, demonstrating that, despite its computational complexity, the approach can be practically applied for the real-time control of fast processes.

Additionally, the proposed MPC is tested in a real scenario, where a metal sheet (with the same features described before) is deformed by using a press having two hydraulic actuators, produced by the Gigant Industries Company. The whole testing setup is shown in Fig. 2.9, where all the principal components are highlighted. Specifically, at process control level, the MPC runs in MATLAB®, while a Siemens S7-1500 PLC is involved at machine control level. The draw-in of points A, B, and C is acquired by means of three laser displacement sensors provided by Keyence Corporation and managed by the

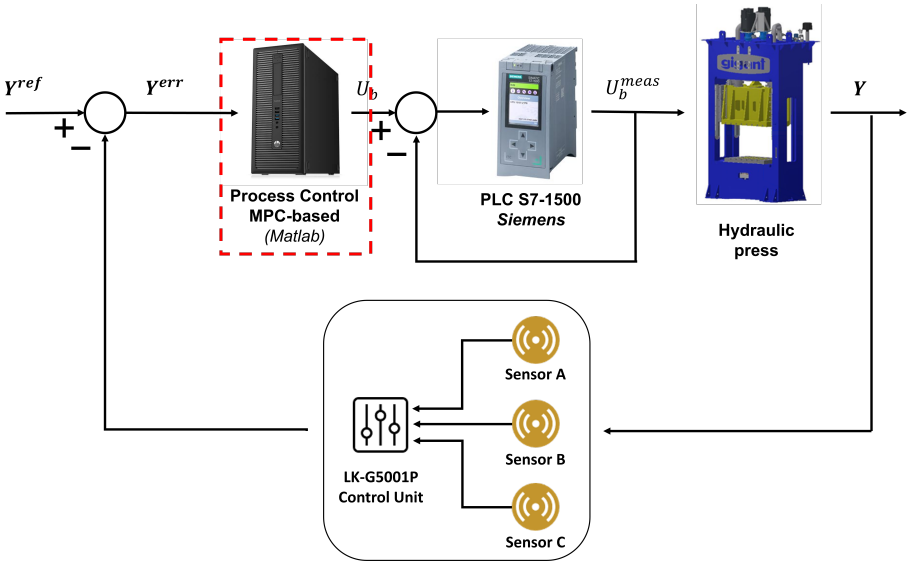

Figure 2.9: Setup configuration for the real experiments.

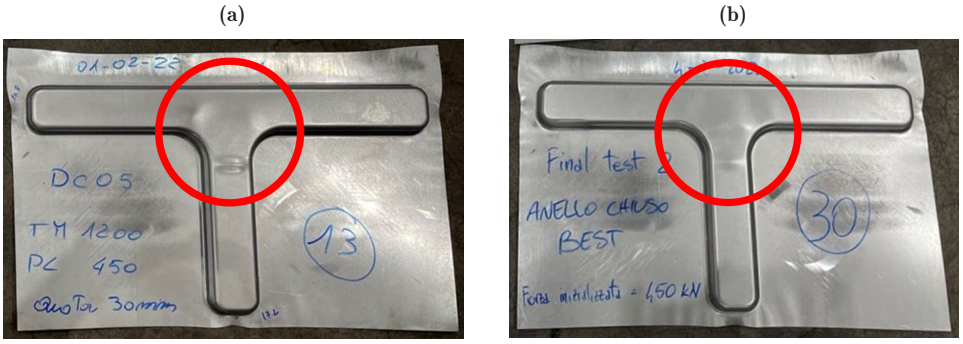

Figure 2.10: "T-shaped" metal sheet obtained without (a) and with (b) the implementation of the MPC-based process control.

LK-G5001P Control Unit [45]. The MPC-based PC uses the OPC UA and the TCP/IP protocol to communicate with the PLC and the LK-G5001P Control Unit, respectively. In Fig. 2.10 we can appreciate the effectiveness of the proposed process control on the quality of the stamped part. The red circle highlights the part of the "T-shaped" most susceptible to defects during the drawing phase. By comparing the stamped workpiece obtained without any process control (Fig. 2.10 (a)) and with the proposed MPC (Fig. 2.10 (b)), the quality improvement is evident.

2.4.3 Comparison with a Multiple PID-Based Control Architecture

With the aim of evaluating and better highlighting the advantages of our approach with respect to the related literature, we provide a comparison between the results obtained by the proposed technique and those achieved by a baseline method, using the PID control approach. To this aim, since the forming process is identified as a SIMO system, a custom multiple PID-based control architecture, whose block diagram is shown in Fig. 2.12, is used for the comparison analysis. Each $\text{PID}_i(z)$ (with $i = 1, 2, 3$) – in series to a saturation block – computes a PID time-discrete control action contributing to the overall control input $U_b^{\text{ref}}(k)$ (i.e., the BHF) as follows:

$$U_b^{\text{PID}}(k) = \gamma_1 U_1^{\text{PID}}(k) + \gamma_2 U_2^{\text{PID}}(k) + \gamma_3 U_3^{\text{PID}}(k) \tag{2.7}$$

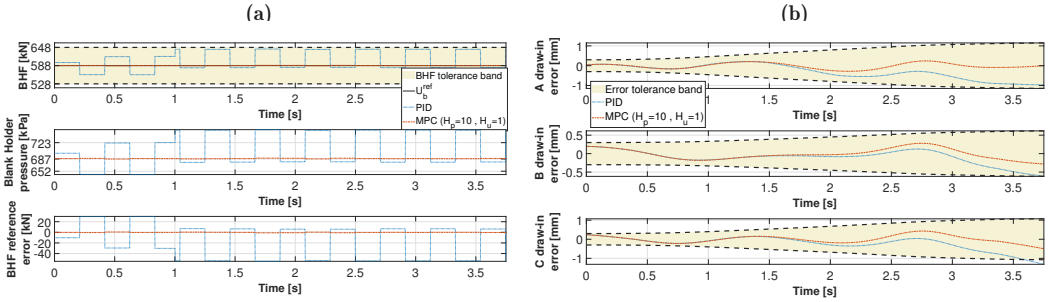

Figure 2.11: Closed-loop system response comparison between the multiple PID-based and the MPC-based ($H_p = 10, H_u = 1$) controllers. Input reference tracking capability (a) and output error of critical points A, B, and C vs. tolerance band (b).

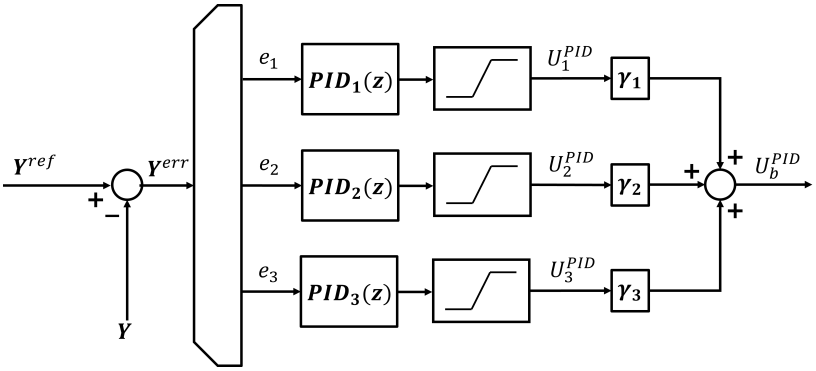

Figure 2.12: Block diagram of the proposed multiple PID-based controller architecture.

where γ_1, γ_2, and γ_3 are weighting factors adopted in the linear combination. Note that, in this approach, the metal forming process is considered as composed by n different SISO systems (one for each measured draw-in and identified with a HW model) that have to be controlled individually. The tuning phase of this PID-based controller has a twofold drawback. On the one hand, the tuning is required for each subsystem, resulting in an effort increase. On the other hand, the final control action needed by the whole system has to be determined through a proper weighted sum represented by Eq. (2.7), where the weights have to be appropriately determined. This implies the execution of a non negligible *trial-and-error* procedure which contributes to making the tuning phase difficult. Moreover, the tuned parameters must be modified whenever any characteristic of the plant changes, resulting in a non scalable architecture, in contrast with the MPC approach. These considerations imply that the tuning phase of the PID-based controller is definitely harder than the MPC design.

The PID-based controller described above is tested in the DT control architecture shown in Fig. 2.7, where the MPC controller is replaced by the PID controller. Also in this case, a step disturbance affecting the control input and characterized by an amplitude equal to 4% of the reference value is considered. Figure 2.11 shows the different closed-loop control responses obtained by the multiple PID-based controller against the MPC based controller. From the results comparison it is evident that the PID-based controller does not guarantee a robust closed-loop stability and an albeit small deviation of the BHF with respect to the optimal trajectory U_b^{ref} occurs, see Fig. 2.11 (a). Not surprisingly, the benefit of the proposed MPC methodology is to include the machine control loop into the process control closed-loop in order to perform a constant BHF reference value for the inner loop as the machine controller operates, and achieve the desired draw-in trajectories. Indeed, as shown in Fig. 2.11, the MPC-based process control allows keeping the errors related both to the control output within the tolerance band (represented by the shaded area in the figure) better than the PID-based approach. In addition, it is worthwhile

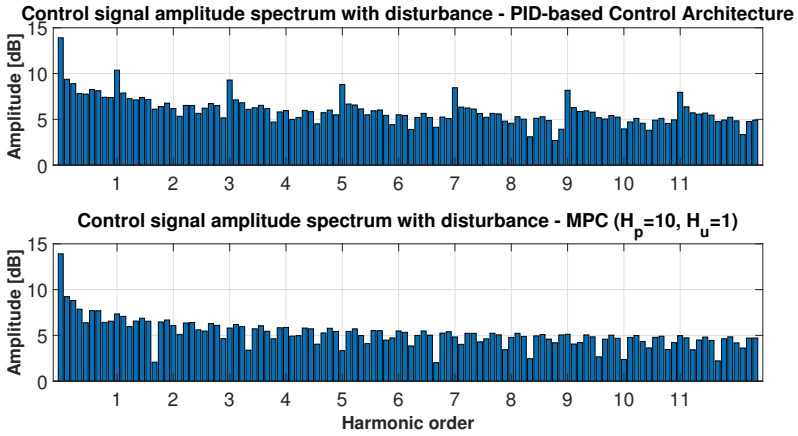

Figure 2.13: BHF FFT amplitudes spectrum comparison between the PID-based and MPC-based controllers.

noting that in the case of the PID-based approach the draw-in error of the critical point C goes out of the band, as shown in the bottom subplot of Fig. 2.11 (b). Moreover, the BHF value computed by the process controller does not oscillate around the reference value, leading a higher closed-loop stability than the multiple PID case. Finally, we remark that the MPC based control system requires a lower control effort than the multiple PID-based approach. To this aim, the comparison of the control effort of the BHF signal obtained by the multiple PID and the MPC -based control system is analyzed using the Fast Fourier Transformation (FFT). Analyzing the FFT amplitude spectra reported in Fig. 2.13, it is apparent that the high-order harmonics content obtained by the MPC approach is more moderate than the one achieved by the PID-based one, thus its control effort is lower.

2.5 Conclusions

In this chapter we present an indirect DD process control method for deep drawing. The main goal of the proposed approach is to optimize the quality of the process, avoiding defects in the stamped part. A comprehensive methodology is developed in detail, ranging from the process identification through an HW model using offline data to the MPC design. Specifically, the chapter contribution is twofold. On the one hand, the proposed approach fills a gap in the existing literature, where –in reference to metal forming– classical PID controllers are commonly employed, even though they cannot properly handle constraints and require a difficult tuning phase, thus not always producing an optimal control signal. Conversely, our approach achieves an excellent control quality of the multi-variable constrained process under a simple fast-tuning phase and short controlling time period. On the other hand, the application to a deep drawing real press highlights the effectiveness of the proposed methodology in designing a suitable control system able to make the stamped parts free of defects. Finally, the proposed MPC is tested first in a DT-based SWIL framework and, then, in a real scenario, where a "T-shaped" metal sheet is deformed by a real hydraulic press, showing the effectiveness of the proposed process control. Specifically, the improvement in the quality of the stamped part –compared to that achieved without any process control– is evident. In addition, the practical feasibility of the MPC-based architecture in demonstrated, despite its computational complexity and the fast dynamics of the involved process.

Future developments could focus on the extension of the proposed approach to explicit MPC, since it allows reduce the computational time needed to determine the control actions. Another research development might involve the proactive integration of adaptive approaches into the control architecture for making the system able to autonomously understand the process operating conditions.

Table 2.A.1: Mean value and variance (expressed in [ms]) of time-series that compose the training and the validation datasets.

		Mean value	Variance
Training dataset	draw-in A	14.4708	110.3194
	draw-in B	4.8239	15.5817
	draw-in C	12.1045	92.4157
Validation dataset	draw-in A	14.4580	110.5453
	draw-in B	4.8443	15.6440
	draw-in C	12.0870	92.4777

2.A Appendix A: Identification of the Deep Drawing Process

For the sake of identifying the deep drawing process model, several experiments are conducted in accordance with three different values of the BHF (i.e., 400, 700, and 588 kN) on the press. Figure 2.5 (b) shows the draw-in of the A, B, and C critical points, measured and recorded during the deep drawing phase. It is important to point out that the duration of each phase, reported in Table 2.1, is indicative. Specifically, for the considered case, the duration of the deep drawing phase (step 3 of the press work-cycle) is 3.75 s on average and a stroke of 30 mm at a speed of 8 mm/s is run on average by the punch. First, the collected data are arranged in a training and validation dataset, one per each experiment. As shown in Table 2.A.1, both training and validation datasets have the same statistics, thus confirming the correctness of the datasets definition.

Subsequently, as indicated in Section 2.3.1, an identification analysis is performed in order to define the parameters of the HW representing the considered deep drawing process. In particular, the analyzed process is suitably identified through a second-order HW model, i.e., $p = 2$. The input and output non-linear functions indicated by $\sigma_{in}(\cdot)$ and $\sigma_{out}(\cdot)$ (Fig. 2.4), respectively, are modeled with a first-degree polynomial, whilst the transfer matrix $\mathbf{W}(z)$ of the linear SIMO block given in (2.2) is obtained as follows:

$$\mathbf{W}(z) = \begin{pmatrix} \frac{-0.98z^{-1}-z^{-2}}{1-1.99z^{-1}+0.99z^{-2}} \\ \frac{-0.93z^{-1}-z^{-2}}{1-1.99z^{-1}+z^{-2}} \\ \frac{-0.95z^{-1}-z^{-2}}{1-1.99z^{-1}+z^{-2}} \end{pmatrix} \tag{2.A.1}$$

where each element represents the transfer function of each SISO system described by the BHF as input and the A, B, and C draw-in profiles as outputs. Both numerator and denominator polynomials have second order, since the model is of the second order. From the output-error polynomial form of the linear SIMO block, the corresponding state-space representation given in (2.3) is obtained as follows:

$$\begin{cases} \boldsymbol{X}_0(k+1) = \mathbf{A}\boldsymbol{X}_0(k) + \mathbf{B}U_0(k) \\ \boldsymbol{Y}_0(k) = \mathbf{C}\boldsymbol{X}_0(k) + \mathbf{D}U_0(k) \end{cases} \tag{2.A.2}$$

where $\boldsymbol{X}_0 = \left(\boldsymbol{X}_A^\top, \boldsymbol{X}_B^\top, \boldsymbol{X}_C^\top\right)^\top$ is the state variable vector with $\boldsymbol{X}_A = \left(X_{A1}, X_{A2}\right)^\top$, $\boldsymbol{X}_B = \left(X_{B1}, X_{B2}\right)^\top$, $\boldsymbol{X}_C = \left(X_{C1}, X_{C2}\right)^\top$, $\boldsymbol{Y}_0 = \left(Y_A, Y_B, Y_C\right)^\top$ is the output vector, and U_0 is the input. The state, input, output, and input-output matrix $\mathbf{A}, \mathbf{B}, \mathbf{C}$, and $\mathbf{D}$

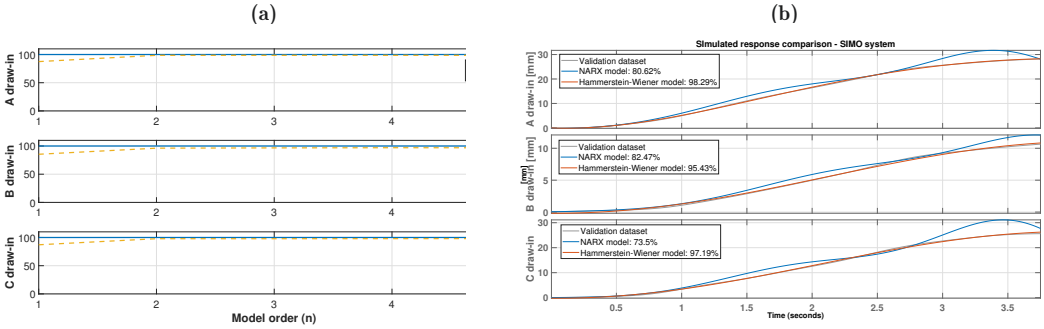

Figure 2.A.1: Validation fitting index (a) and simulated response (b), comparison between HW and NARX models.

are respectively defined as:

$$\mathbf{A} = \begin{pmatrix} 0.999 & 0.002 & 0 & 0 & 0 & 0 \\ -0.002 & 0.999 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.002 & 0 & 0 \\ 0 & 0 & -0.002 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0.002 \\ 0 & 0 & 0 & 0 & -0.002 & 0.999 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} 0.3537 \\ -0.001 \\ 0.354 \\ 447.400 \\ 0.354 \\ 429.300 \end{pmatrix} \tag{2.A.3}$$

$$\mathbf{C} = \begin{pmatrix} -2.80 & -32.12 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.096 & -0.002 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.066 & -0.002 \end{pmatrix} \quad \mathbf{D} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

The HW model described above is implemented in the MATLAB® engineering software using the IDENTIFICATION TOOLBOX®. With the aim of highlighting the advantages of the second order HW model with respect to other approaches, we provide a comparison with the results obtained by the NARX model. The comparison results are depicted in Fig. 2.A.1. Specifically, Fig. 2.A.1 (a) shows that, although the NARX model fits the validation dataset with a confidence of 100%, it suffers from over-parameterization (due to the inherent structural-identifiability issue of this class of models). Effectively, Fig. 2.A.1 (b) demonstrates that the second-order HW model has a higher close-fitting simulated response compared with respect to the NARX model. For each draw-in output, the HW model presents a behaviour that is very similar to the actual one.

Moreover, as a further validation analysis, an external disturbance is superimposed to the input BHF used in the deep drawing process. Figure 2.A.2 shows the results obtained by the system simulation conducted using a disturbance step with increasing amplitude (1%, 2%, 3%, and 4% of the input BHF value). As expected, the results demonstrate the advantages of a closed-loop control system for the metal forming process. Conversely, in Fig. 2.A.3 the simulated system responses are shown for a BHF whose value corresponds to different working conditions (i.e., 400 kN and 700 kN). It is apparent that the obtained HW model correctly represents the metal forming process only when the BHF gets the nominal value (i.e., 588 kN). Fig. 2.A.3 (a) indicates the case of a lower value of BHF, where the model tends to under-estimate its response; conversely, Fig. 2.A.3 (b) illustrates the case of a BHF value of 700 kN, for which the model tends to over-estimate its response.

2.B Appendix B: MPC Tuning

The MPC-based process controller defined by (2.6) is designed and tuned with the MPC DESIGNER®, inheriting the structure of the identified process in the scheme of Fig. 2.4.

The input constraints are limited by the hydraulic piston sensitivity (±60 kN), while the output constraints are limited within a tolerance band centered in the optimal draw-in

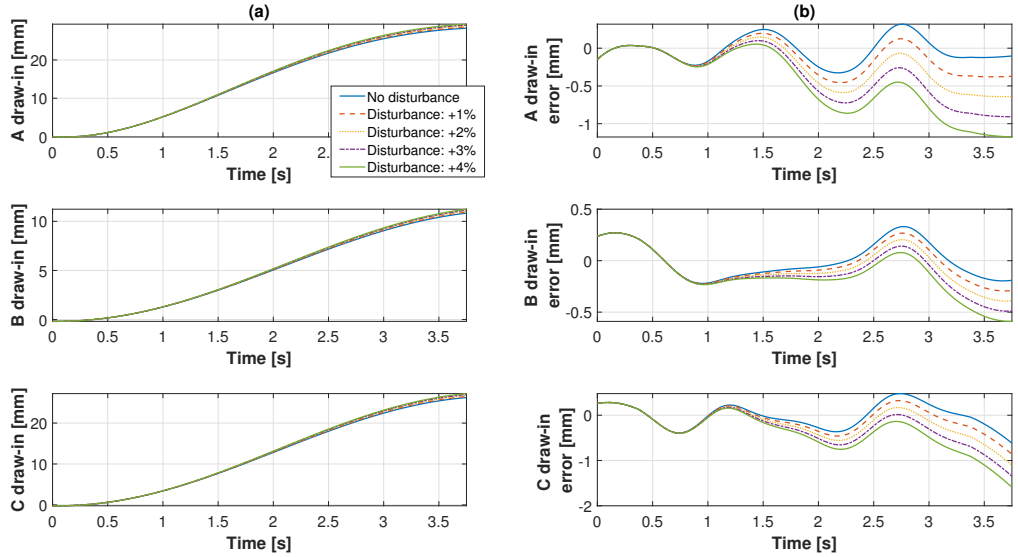

Figure 2.A.2: System simulated response comparison (a) and draw-in errors (b) in presence of a step disturbance (amplitude equal to 1%, 2%, 3%, and 4% of the BHF value).

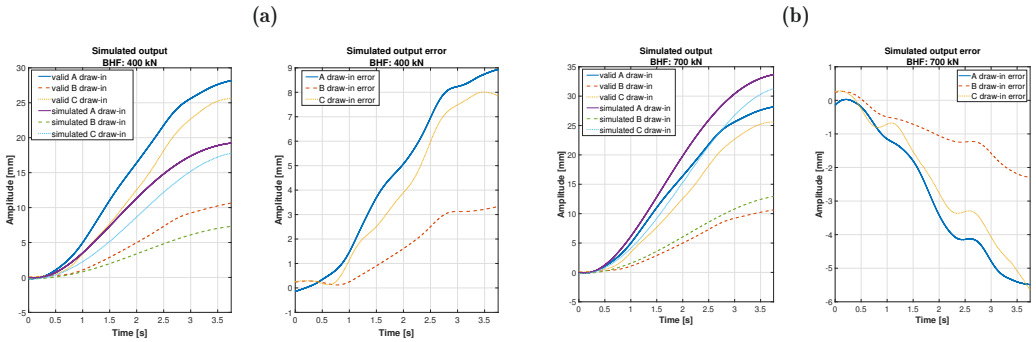

Figure 2.A.3: Simulated outputs and output errors for the identified SIMO system with an input BHF equal to of 400 kN (a) and 700 kN (b).

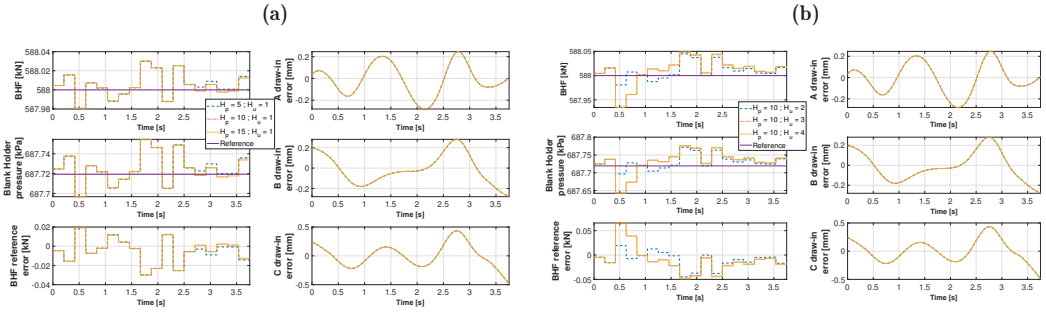

Figure 2.A.4: MPC Closed-loop system response for $H_u = 1$ and $H_p = (5, 10, 15)$ (a) and for $H_p = 10$ and $H_u = (2, 3, 4)$ (b).

profiles Y_1^{ref}, Y_2^{ref}, and Y_3^{ref}. The tolerance band is defined by parameter α, whose value is an outcome of the robust analysis of the stamping experiments. All the constraint parameters values are reported in Table 2.B.1. The values of the other MPC parameters (i.e., H_p, and H_u, the variables weights, the variables rate weights, and α) are detailed

Table 2.B.1: MPC input and output constraints.

Parameter	Unit	Value
$\mathrm{U}_b^{\mathrm{lb}}$, $\mathrm{U}_b^{\mathrm{ub}}$	kN	528, 648
$\mathrm{Y}_1^{\mathrm{lb}}(k)$, $\mathrm{Y}_1^{\mathrm{ub}}(k)$	mm	$(1-\alpha)Y_1^{\mathrm{ref}}(k)$, $(1+\alpha)Y_1^{\mathrm{ref}}(k)$
$\mathrm{Y}_2^{\mathrm{lb}}(k)$, $\mathrm{Y}_2^{\mathrm{ub}}(k)$	mm	$(1-\alpha)Y_2^{\mathrm{ref}}(k)$, $(1+\alpha)Y_2^{\mathrm{ref}}(k)$
$\mathrm{Y}_3^{\mathrm{lb}}(k)$, $\mathrm{Y}_3^{\mathrm{ub}}(k)$	mm	$(1-\alpha)Y_3^{\mathrm{ref}}(k)$, $(1+\alpha)Y_3^{\mathrm{ref}}(k)$

Table 2.B.2: MPC parameters.

Parameter	Unit	Value
H_p	-	10
H_u	-	1
T_s	ms	200
w_1^y	-	1
w_2^y	-	1
w_3^y	-	1
$w^{\Delta u}$	-	0
α	-	0.03

in Table 2.B.2. The MV nominal input value U_b is fixed at 588 kN and for the output reference trajectory (indicated as Y_1^{ref}, Y_2^{ref}, and Y_3^{ref}) the optimal draw-in displacement shown in Fig. 2.6 is considered. Moreover, since the inner machine control loop works with two hydraulic pistons having a response time of 100 ms, the MPC-based process controller sampling time T_S is settled at 200 ms. With regard to the control horizon H_p and the prediction horizon H_u, since they are considered as tuning parameters, their changes are evaluated during the testing phase. Figure 2.A.4 (a) shows the closed-loop control system simulation for three different values of prediction horizon ($H_p = \{5, 10, 15\}$) and a constant value of control horizon ($H_u = 1$), while Fig. 2.A.4 (b) illustrates the closed-loop system dynamics for increasing values of the control horizon ($H_u = \{2, 3, 4\}$) and a constant value of the prediction horizon ($H_p = 10$). Changes in H_p does not affect the behaviour of the input variable, whereas, as H_u grows, the CV signal tends to be more and more oscillating around the nominal value.

References

[1] Lim, Y., Ulsoy, A. G., and Venugopal, R., *Process control for sheet-metal stamping.* Springer, 2013.

[2] Palmieri, M. E., Lorusso, V. D., and Tricarico, L., "Robust optimization and kriging metamodeling of deep-drawing process to obtain a regulation curve of blank holder force," *Metals*, vol. 11, no. 2, p. 319, 2021.

[3] Gopalakrishnan, K., Goldberg, K., Bone, G. M., *et al.*, "Unilateral fixtures for sheet-metal parts with holes," *IEEE Trans. Autom. Sc. Eng.*, vol. 1, no. 2, pp. 110–120, 2004.

[4] Ng, Y.-M. H., Yu, M., Huang, Y., and Du, R., "Diagnosis of sheet metal stamping processes based on 3-d thermal energy distribution," *IEEE Trans. Autom. Sc. Eng.*, vol. 4, no. 1, pp. 22–30, 2007.

[5] Sheng, W., Chen, H., Xi, N., and Chen, Y., "Tool path planning for compound surfaces in spray forming processes," *IEEE Trans. Autom. Sc. Eng.*, vol. 2, no. 3, pp. 240–249, 2005.

[6] Hsu, C.-W., Ulsoy, A., and Demeri, M., "Development of process control in sheet metal forming," *J. Mater. Process. Technol.*, vol. 127, no. 3, pp. 361–368, 2002.

[7] Su, C., Zhang, K., Lou, S., Xu, T., and Wang, Q., "Effects of variable blank holder forces and a controllable drawbead on the springback of shallow-drawn ta2m titanium alloy boxes," *The International Journal of Advanced Manufacturing Technology*, vol. 93, no. 5, pp. 1627–1635, 2017.

[8] Dotoli, M., Fay, A., Miśkowicz, M., and Seatzu, C., "An overview of current technologies and emerging trends in factory automation," *International Journal of Production Research*, vol. 57, no. 15-16, pp. 5047–5067, 2019.

[9] Cavone, G., Epicoco, N., and Dotoli, M., "Process re-engineering based on colored petri nets: The case of an italian textile company," in *2020 28th Mediterranean Conference on Control and Automation (MED)*, IEEE, 2020, pp. 856–861.

[10] Siegert, K., Ziegler, M., and Wagner, S., "Closed loop control of the friction force. deep drawing process," *Journal of materials processing technology*, vol. 71, no. 1, pp. 126–133, 1997.

[11] Bozza, A., Cavone, G., Carli, R., Mazzoccoli, L., and Dotoli, M., "An MPC-based Approach for the Feedback Control of the Cold Sheet Metal Forming Process," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, 2021, pp. 286–291. DOI: 10.1109/CASE49439.2021.9551602.

[12] Hsu, C.-W., Ulsoy, A., and Demeri, M., "An approach for modeling sheet metal forming for process controller design," *J. Manuf. Sci. Eng.*, vol. 122, no. 4, pp. 717–724, 2000.

[13] Allwood, J., Duncan, S., Cao, J., *et al.*, "Closed-loop control of product properties in metal forming," *CIRP Annals*, vol. 65, no. 2, pp. 573–596, 2016.

[14] Hsu, C.-W., Ulsoy, A., and Demeri, M., "Process controller design for sheet metal forming," in *Proceedings of the 1999 American Control Conference (Cat. No. 99CH36251)*, IEEE, vol. 1, 1999, pp. 192–196.

[15] Hsu, C.-W., Ulsoy, A., and Demeri, M., "An approach for modeling sheet metal forming for process controller design," *J. Manuf. Sci. Eng.*, vol. 122, no. 4, pp. 717–724, 2000.

[16] Ljung, L., "System identification," in *Signal analysis and prediction*, Springer, 1998, pp. 163–173.

[17] Tunc, L. T. and Budak, E., "Identification and Modeling of Process Damping in Milling," *Journal of Manufacturing Science and Engineering*, vol. 135, no. 2, Mar. 2013, 021001. DOI: 10.1115/1.4023708. eprint: https://asmedigitalcollection.asme.org/manufacturingscience/article-pdf/135/2/021001/6262466/manu_135_2_021001.pdf. [Online]. Available: https://doi.org/10.1115/1.4023708.

[18] Rao, P., Bukkapatnam, S., Beyca, O., Kong, Z. (, and Komanduri, R., "Real-Time Identification of Incipient Surface Morphology Variations in Ultraprecision Machining Process," *Journal of Manufacturing Science and Engineering*, vol. 136, no. 2, Jan. 2014, 021008. DOI: 10.1115/1.4026210. eprint: https://asmedigitalcollection.asme.org/manufacturingscience/article-pdf/136/2/021008/6263211/manu_136_02_021008.pdf. [Online]. Available: https://doi.org/10.1115/1.4026210.

[19] Sammons, P. M., Bristow, D. A., and Landers, R. G., "Two-Dimensional Modeling and System Identification of the Laser Metal Deposition Process," *Journal of Dynamic Systems, Measurement, and Control*, vol. 141, no. 2, Oct. 2018, 021012. DOI: 10.1115/1.4041444. eprint: https://asmedigitalcollection.asme.org/dynamicsystems/article-pdf/141/2/021012/6129245/ds_141_02_021012.pdf. [Online]. Available: https://doi.org/10.1115/1.4041444.

[20] Perrusquía, A. and Yu, W., "Identification and optimal control of nonlinear systems using recurrent neural networks and reinforcement learning: An overview," *Neurocomputing*, vol. 438, pp. 145–154, 2021.

[21] Li, J., Zheng, W. X., Gu, J., and Hua, L., "A recursive identification algorithm for wiener nonlinear systems with linear state-space subsystem," *Circuits, Systems, and Signal Processing*, vol. 37, no. 6, pp. 2374–2393, 2018.

[22] Zhao, W., Zheng, W. X., and Bai, E.-W., "A recursive local linear estimator for identification of nonlinear arx systems: Asymptotical convergence and applications," *IEEE Transactions on Automatic Control*, vol. 58, no. 12, pp. 3054–3069, 2013.

[23] Zhao, W.-X., Chen, H.-F., and Zheng, W. X., "Recursive identification for nonlinear arx systems based on stochastic approximation algorithm," *IEEE Transactions on Automatic Control*, vol. 55, no. 6, pp. 1287–1299, 2010.

[24] Ulsoy, A. G., Koren, Y., and Rasmussen, F., "Principal developments in the adaptive control of machine tools," *ASME Journal of Dynamic Systems, Measurement, and Control*, 1983.

[25] Ulsoy, A. G. and Koren, Y., "Control of machining processes," *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, vol. 115, no. 2B, pp. 301–308, 1993.

[26] Rashap, B. A., Elta, M. E., Etemad, H., *et al.*, "Control of semiconductor manufacturing equipment: Real-time feedback control of a reactive ion etcher," *IEEE Trans. Semicond. Manuf.*, vol. 8, no. 3, pp. 286–297, 1995.

[27] Lim, Y., Venugopal, R., and Ulsoy, A. G., "Auto-tuning and adaptive control of sheet metal forming," *Control Eng. Pract.*, vol. 20, no. 2, pp. 156–164, 2012.

[28] Borase, R. P., Maghade, D., Sondkar, S., and Pawar, S., "A review of pid control, tuning methods and applications," *Int. J. Dyn. Control.*, pp. 1–10, 2020.

[29] Liu, W., Liang, Z., Huang, T., Chen, Y., and Lian, J., "Process optimal control of sheet metal forming springback based on evolutionary strategy," in *2008 7th World Congress on Intelligent Control and Automation*, IEEE, 2008, pp. 7940–7945.

[30] Endelt, B., Tommerup, S., and Danckert, J., "A novel feedback control system–controlling the material flow in deep drawing using distributed blank-holder force," *J. Mater. Process. Technol.*, vol. 213, no. 1, pp. 36–50, 2013.

[31] Hao, W. and Duncan, S., "Constrained model predictive control of an incremental sheet forming process," in *IEEE Int. Conf. on Control Applications (CCA)*, IEEE, 2011, pp. 1288–1293.

[32] Lu, H., Kearney, M., Li, Y., Liu, S., Daniel, W. J., and Meehan, P. A., "Model predictive control of incremental sheet forming for geometric accuracy improvement," *The International Journal of Advanced Manufacturing Technology*, vol. 82, no. 9-12, pp. 1781–1794, 2016.

[33] Baheti, R. and Gill, H., "Cyber-physical systems," *The impact of control technology*, vol. 12, no. 1, pp. 161–166, 2011.

[34] Tao, F., Zhang, H., Liu, A., and Nee, A. Y., "Digital twin in industry: State-of-the-art," *IEEE Transactions on industrial informatics*, vol. 15, no. 4, pp. 2405–2415, 2018.

[35] De Boor, C., "A practical guide to splines springer-verlag," *New York*, 1978.

[36] Savitzky, A. and Golay, M. J., "Smoothing and differentiation of data by simplified least squares procedures.," *Analytical chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.

[37] Keivanpour, S. and Ait Kadi, D., "Strategic eco-design map of the complex products: Toward visualisation of the design for environment," *International Journal of Production Research*, vol. 56, no. 24, pp. 7296–7312, 2018.

[38] Rosen, R., Von Wichert, G., Lo, G., and Bettenhausen, K. D., "About the importance of autonomy and digital twins for the future of manufacturing," *IFAC-PapersOnLine*, vol. 48, no. 3, pp. 567–572, 2015.

[39] Vachálek, J., Bartalskỳ, L., Rovnỳ, O., Šišmišová, D., Morháč, M., and Lokšík, M., "The digital twin of an industrial production line within the industry 4.0 concept," in *2017 21st international conference on process control (PC)*, IEEE, 2017, pp. 258–262.

[40] Carli, R., Cavone, G., Dotoli, M., Epicoco, N., Manganiello, C., and Tricarico, L., "Ict-based methodologies for sheet metal forming design: A survey on simulation approaches," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, IEEE, 2019, pp. 128–133.

[41] Fur, S., Riedel, O., and Verl, A., "Hybrid commissioning of production plants," in *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2021, pp. 1–6. DOI: 10.1109/ETFA45728.2021.9613256.

[42] Scholz, M., Oberschachtsiek, S., Donhauser, T., and Franke, J., "Software-in-the-loop testbed for multi-agent-systems in a discrete event simulation: Integration of the java agent development framework into plant simulation," in *2017 IEEE International Systems Engineering Symposium (ISSE)*, 2017, pp. 1–6. DOI: 10.1109/SysEng.2017.8088320.

[43] Jaensch, F., Csiszar, A., Sarbandi, J., and Verl, A., "Reinforcement learning of a robot cell control logic using a software-in-the-loop simulation as environment," in *2019 Second International Conference on Artificial Intelligence for Industries (AI4I)*, 2019, pp. 79–84. DOI: 10.1109/AI4I46381.2019.00027.

[44] *Siemens products website*, https://new.siemens.com/it/it/prodotti/ software.html, Accessed: 2022-01-15.

[45] Corporation, K., *Ultra high-speed/high-accuracy laser displacement sensor - user manual*, 2022.

Chapter 3

Robust Control of Nonlinear Systems Based on Online Data-Driven Methods

III

Abstract

This chapter proposes a novel robust Data-Driven (DD) method for controlling unknown input-affine nonlinear systems. First, we estimate the system dynamics from noisy data offline through Subspace Identification of Nonlinear Dynamics. Then, at each time step during runtime, we exploit this estimation to deduce a feedback-linearization control law that robustly regulates all the systems consistent with the data. Notably, the control law is derived by solving a Semidefinite Programming (SDP) online. Moreover, closed-loop stability is ensured by constraining a Lyapunov function to descend in each time step using a linear-matrix-inequality representation. Unlike related DD control approaches for nonlinear systems based on SDP, our approach does not require any approximation of the nonlinear dynamics, while requiring the knowledge of a library of candidate basis functions. Finally, we validate our theoretical contributions by simulations for stabilization and tracking, outperforming another DD literature-inspired controller.

Contents

3.1 Introduction

Model-based control –which exploits the knowledge of the plant dynamics– has largely been motivated by the growing availability of first-principles models and the emergence of state-space control theory [1]. However, first-principles laws are often difficult to apply, requiring both time and expert knowledge. On the contrary, system identification [2] has been widely used as an indirect *Data-Driven* (DD) approach [3] to obtain models from measured trajectories, while these methods often do not provide guarantees for the closed-loop. For these reasons, alternative DD control techniques are becoming more and more popular, thanks to their capability of designing a controller from data, avoiding the full knowledge of the model [4].

While many examples of DD control with stability guarantees are given in the literature for linear systems (e.g., *Willem's Fundamental Lemma* [5]), nonlinear control is more challenging [6]. Indeed, more data are required than in the linear case, and non-convex optimization problems often arise in the controller synthesis. Various approaches have been developed in this field, including nonlinear adaptive control [7], adaptive Model Predictive Control [8], and set-membership estimation [9]. To avoid non-convex optimization that is typically inherited by nonlinearities, one of the most recent research lines considers *Semidefinite Programming* (SDP). SDP has become popular for control problems since it can handle complex constraints characterized by *Linear Matrix Inequalities* (LMIss) [10].

Moreover, a recent research [11] also considers the application of SDP for computing control inputs online. However, these methods require an approximation of the nonlinearity, such as polynomial approximations [12], embedding into Linear Parameter-Varying dynamics [13], and using the Koopman operator [14]. On the other hand, *feedback linearization* represents a viable approach for canceling out system nonlinearities, aimed at linearizing the dynamics via the feedback control action. Within the DD context, feedback linearization is utilized for noisy-free data [15], for single input systems [16], and based on online training data update [17].

Data-Driven Robust (DDR) control for nonlinear systems refers to *set membership*, in the sense that a controller robustly stabilizes a set of all the systems consistent with data [18]. For instance, the cited paper suggests a DDR control based on Linear Quadratic Regulation, rewritten as an SDP. Even providing a low-complexity computational solution, it approximates the nonlinearity of the dynamics with evident limitations. Other examples are given by [19] and [20], which propose DDR controllers based on Sum-of-Squares optimization. However, these approaches suffer from scaling issues and are limited to polynomial and rational dynamics. Further, [21] utilizes the Koopman operator to transform the nonlinear dynamics into a linear form for a robust controller design. Notwithstanding, the truncation error of this transformation is generally unknown. Then, in [22] and [23], a DD regulator is designed using the Taylor's expansion to approximate the unknown nonlinear dynamics, but its accuracy is limited for highly nonlinear systems, affecting the control performance.

In this chapter, we propose an online DD control approach for unknown input-affine nonlinear systems. Our method first estimates offline the system dynamics through Subspace Identification of Nonlinear Dynamics (SINDy) [24] applied on noisy data. Then, at runtime, we exploit this estimation to set-up an SDP problem (similar to [18]) to robustly controlling the set of all the systems consistent with the data via feedback-linearization. More specifically, closed-loop stability is explicitly enforced by an LMI constraint ensuring the descend of a Lyapunov function over time for all the systems within the set. Thereby, the optimization problem is convex and enables the online inference of the unknown coefficients of the dynamics. To this aim, we require the knowledge of a library of candidate basis functions, as suggested in [15], [18], [22], [25], [26]. However, in contrast to these works, our method does not require nonlinear approximation, achieving a less conservative robust performance with respect to the set of systems consistent with the noisy data.

3.1.1 Notation

The state and input vectors are respectively indicated by $\boldsymbol{x} = \big(x_1, \dots, x_n\big)^\top \in \mathbb{R}^n$ and $\boldsymbol{u} = \big(u_1, \dots, u_n\big)^\top \in \mathbb{R}^n$. The symbol $\boldsymbol{x}^+$ is the state vector at the next time point, while $\boldsymbol{x}(k)$ is the k-th retrieved data sample. Moreover, we denote the Euclidean norm of a vector $\boldsymbol{v} \in \mathbb{R}^n$ by $\|\boldsymbol{v}\|_2$ and the induced matrix norms for the 1- and Euclidean vector norm by $\|\mathbf{M}\|_1$ and $\|\mathbf{M}\|_2$, respectively. Furthermore, we respectively name $\mathbf{I}$ and $\mathbf{0}$ the identity and the zero matrix of suitable dimensions. We use the superscript $^\top$ for denoting the transpose of a matrix. The abbreviation $\star^\top \mathbf{BA}$ indicates the quadratic form $\mathbf{A}^\top \mathbf{BA}$ of the matrices $\mathbf{A}$ and $\mathbf{B}$ of suitable dimensions. Finally, $\succ$ ($\succeq$) and $\prec$ ($\preceq$) denote the positive (semi-) and negative (semi-) definiteness of a matrix.

3.2 Problem Statement

Consider the input-affine nonlinear discrete-time system

$$\boldsymbol{x}^+ = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}, \tag{3.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n \times n}$ describe the unknown state transition and input functions, respectively. To infer the unknown dynamics, we suppose the knowledge of basis functions containing the dynamics.

Assumption 3.2.1

The unknown state transition and input functions can be written upon the following parameterized representations

$$f(\boldsymbol{x}) = \boldsymbol{\Xi}_1 \mathbf{F}(\boldsymbol{x})$$
$$g(\boldsymbol{x}) = \boldsymbol{\Xi}_2 \mathbf{G}(\boldsymbol{x}),$$

where $\mathbf{F} : \mathbb{R}^n \to \mathbb{R}^{n_f}$ *and* $\mathbf{G} : \mathbb{R}^n \to \mathbb{R}^n \times n$ *are the known state transition and input basis functions, respectively. Moreover, the functions, given by the rows of matrix* $\mathbf{F}(\boldsymbol{x})$ *and* $\mathbf{G}(\boldsymbol{x})$*, are linearly independent. The coefficient matrices* $\boldsymbol{\Xi}_1 \in \mathbb{R}^{n \times n_f}$ *and* $\boldsymbol{\Xi}_2 \in \mathbb{R}^{n \times n}$ *are unknown.* □

By matrix multiplication, we can rewrite (3.1) as

$$\boldsymbol{x}^+ = \boldsymbol{\Xi}\boldsymbol{\Theta}(\boldsymbol{x}, \boldsymbol{u}), \tag{3.2}$$

where we summarize the nonlinear basis functions into

$$\boldsymbol{\Theta}(\boldsymbol{x}, \boldsymbol{u}) = \begin{pmatrix} \mathbf{F}(\boldsymbol{x}) \\ \mathbf{G}(\boldsymbol{x})\boldsymbol{u} \end{pmatrix}$$

and the unknown coefficient matrices into $\boldsymbol{\Xi} = \begin{pmatrix} \boldsymbol{\Xi}_1 & \boldsymbol{\Xi}_2 \end{pmatrix}$. Due to the linear independence of the basis functions according to Assumption 3.2.1, the coefficient matrix $\boldsymbol{\Xi}$ is unique. Assumption 3.2.1 requires some insights into the unknown dynamics, which might be available from first-principles, e.g., for electrical or mechanical systems. For instance, $\mathbf{F}(\boldsymbol{x})$ and $\mathbf{G}(\boldsymbol{x})$ can have the following form:

$$\mathbf{F}(\boldsymbol{x}) = \begin{pmatrix} \boldsymbol{x}^\top & \begin{pmatrix} x_1^2, \dots, x_n^2 \end{pmatrix}^\top & \sin(x_1) & \cdots \end{pmatrix}^\top$$
$$\mathbf{G}(\boldsymbol{x}) = \begin{pmatrix} 1 & 0 & \cdots \\ 0 & 1 + x_1^2 & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}.$$

To infer $\boldsymbol{\Xi}$, we further ask for $\mathrm{N}+1$ noisy data samples $\left\{\boldsymbol{x}(k), \boldsymbol{u}(k)\right\}_{k=1}^{\mathrm{N}+1}$ satisfying the open-loop dynamics (3.1):

$$\boldsymbol{x}(k+1) = f\big(\boldsymbol{x}(k)\big) + g\big(\boldsymbol{x}(k)\big)\boldsymbol{u}(k) + \boldsymbol{d}(k) \tag{3.3}$$

$\forall k = 1, \dots, \mathrm{N}$, with *unknown* noise $\boldsymbol{d}(k)$. Since these data lack information for arbitrary noise realizations, we also require the following general noise characterization, as in [12] and [18].

Assumption 3.2.2

Let the noise realizations $\boldsymbol{d}(k), \forall k = 1, \dots, \mathrm{N}$*, satisfy* $\|\boldsymbol{d}(k)\|_2^2 \leq \epsilon_d$ *with known* $\epsilon_d \geq 0$*.* □

The goal of the remainder of this chapter is to find a control input $\boldsymbol{u}$ leading the nonlinear system (3.1) to track a reference signal $\boldsymbol{x}_\mathrm{r}$ in a stable fashion.

3.3 Data-Driven Control Problem Formulation

The proposed control law is based on feedback linearization, where the unknown coefficient matrices are first estimated offline utilizing the so-called SINDy regression. Then, to guarantee stability regarding the reference –even with a not precise coefficient inference– the state feedback is updated at each time step by solving an SDP problem online.

3.3.1 Offline Regression

Since typically only few nonlinearities in $\boldsymbol{\Theta}(\boldsymbol{x},\boldsymbol{u})$ are non-zero, we set-up the following SINDy-based sparse regression problem [24]:

$$\min_{\widehat{\boldsymbol{\Xi}}} \quad \sum_{k=1}^{\mathrm{N}} \frac{1}{2}\left\|\boldsymbol{x}(k+1)-\widehat{\boldsymbol{\Xi}}\boldsymbol{\Theta}\big(\boldsymbol{x}(k),\boldsymbol{u}(k)\big)\right\|_2^2+\gamma\left\|\widehat{\boldsymbol{\Xi}}\right\|_1. \tag{3.4}$$

Here, $\widehat{\boldsymbol{\Xi}}\in\mathbb{R}^{n\times(n_f+n)}$ is the estimated sparse coefficient matrix, while γ is a Pareto-based hyper-parameter used for best balancing model complexity and accuracy, by promoting the sparsity of the coefficient estimation.

In order to introduce a feedback law for the full state linearization once problem (3.4) is solved, the following invertibility assumption is required.

Assumption 3.3.1 (*cf. [25], Ass. 4*)

Let the coefficient matrix $\widehat{\boldsymbol{\Xi}}_2$ and matrix $\mathbf{G}(\boldsymbol{x}),\forall\boldsymbol{x}\in\mathbb{R}^n$, be invertible. □

Assumption 3.3.1 ensures, among others, that the inverse of $\widehat{\boldsymbol{\Xi}}_2\mathbf{G}(\boldsymbol{x})$ exists, which is common for input-affine systems [27], e.g., robotic manipulators or coupled multi-tank systems. Furthermore, while the invertibility of $\mathbf{G}(\boldsymbol{x})$ can be checked offline, we can ensure the invertibility of $\widehat{\boldsymbol{\Xi}}_2$ by a small perturbation.

Under this assumption, we can choose the following feedback-linearization control law for (3.1) to assure a good tracking

$$\widehat{\boldsymbol{u}}(\boldsymbol{x})=\big(\widehat{\boldsymbol{\Xi}}_2\mathbf{G}(\boldsymbol{x})\big)^{-1}\big(-\widehat{\boldsymbol{\Xi}}_1\mathbf{F}(\boldsymbol{x})-\boldsymbol{K}\boldsymbol{e}+\boldsymbol{x}_{\mathrm{r}}^{+}\big), \tag{3.5}$$

where $\boldsymbol{K}\in\mathbb{R}^{n\times n}$ and $\boldsymbol{e}=\boldsymbol{x}-\boldsymbol{x}_{\mathrm{r}}$ are the to-be-chosen control gain matrix and the tracking error vector, respectively. Indeed, by adding and subtracting the quantity $\widehat{\boldsymbol{\Xi}}_2\mathbf{G}(\boldsymbol{x})\widehat{\boldsymbol{u}}(\boldsymbol{x})$ to the "true" dynamics (3.2) and, then, by substituting the control law (3.5), we obtain the tracking error dynamics

$$\boldsymbol{e}^{+}=-\boldsymbol{K}\boldsymbol{e}+\big(\boldsymbol{\Xi}_1-\widehat{\boldsymbol{\Xi}}_1\big)\mathbf{F}(\boldsymbol{x})+\big(\boldsymbol{\Xi}_2-\widehat{\boldsymbol{\Xi}}_2\big)\mathbf{G}(\boldsymbol{x})\widehat{\boldsymbol{u}}(\boldsymbol{x}). \tag{3.6}$$

Hence, in the ideal case of $\big(\widehat{\boldsymbol{\Xi}}_1,\widehat{\boldsymbol{\Xi}}_2\big)\to\big(\boldsymbol{\Xi}_1,\boldsymbol{\Xi}_2\big)$, then (3.6) becomes $\boldsymbol{e}^{+}=-\boldsymbol{K}\boldsymbol{e}$, resulting in an asymptotically stable tracking error dynamics, for a Schur $-\boldsymbol{K}$.

However, due to the inexact inference on the regression matrix, a static choice of $\boldsymbol{K}$, made offline, does not necessarily lead to a stable tracking error. Thus, we define the Lyapunov function candidate $V(\boldsymbol{e})=\frac{1}{2}\boldsymbol{e}^{\top}\mathbf{P}\boldsymbol{e}$, with $\mathbf{P}\succ 0$, and propose an online control scheme to guarantee its decay over time.

3.3.2 Projection of the Offline Regression

To propose the online control scheme, we first consider the projection of estimation $\widehat{\boldsymbol{\Xi}}$ on the new variable $\widetilde{\boldsymbol{\Xi}}=\begin{pmatrix}\widetilde{\boldsymbol{\Xi}}_1 & \widetilde{\boldsymbol{\Xi}}_2\end{pmatrix}\in\mathbb{R}^{n\times(n_f+n)}$ from the set of coefficients, for which the control input (3.5) decreases the Lyapunov function for all the systems consistent with the retrieved data samples (3.3). Moreover, by utilizing the control structure (3.5), the subsequently proposed control algorithm also provides an online adapted inference of the coefficients $\boldsymbol{\Xi}$.

Lemma 3.3.1 (*Set membership*)

The set of all coefficients consistent with data (3.3), under the noise characterization of Assumption 3.2.2, is given by

$$\Sigma=\left\{\boldsymbol{\Xi}:\begin{pmatrix}\mathbf{I}\\ \boldsymbol{\Xi}^{\top}\end{pmatrix}^{\top}\boldsymbol{\Delta}(k)\begin{pmatrix}\mathbf{I}\\ \boldsymbol{\Xi}^{\top}\end{pmatrix}\preceq 0,k=1,\dots,\mathrm{N}\right\} \tag{3.7}$$

with

$$\boldsymbol{\Delta}(k) = \begin{pmatrix} -\epsilon_d \mathbf{I} + \boldsymbol{x}(k+1)\boldsymbol{x}(k+1)^\top & -\boldsymbol{x}(k+1)\boldsymbol{\Theta}(k)^\top \\ -\big(\boldsymbol{x}(k+1)\boldsymbol{\Theta}(k)^\top\big)^\top & \boldsymbol{\Theta}(k)\boldsymbol{\Theta}(k)^\top \end{pmatrix}$$

and $\boldsymbol{\Theta}(k) = \boldsymbol{\Theta}\big(\boldsymbol{x}(k), \boldsymbol{u}(k)\big)$. □

Proof 3.3.1

By the Dualization Lemma (see [28], 2.10), the noise bound from Assumption 3.2.2 is equivalent to

$$\begin{pmatrix} \mathbf{I} \\ \boldsymbol{d}(k)^\top \end{pmatrix}^\top \begin{pmatrix} -\epsilon_d \mathbf{I} & 0 \\ 0 & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{I} \\ \boldsymbol{d}(k)^\top \end{pmatrix} \preceq 0, \quad k = 1, \dots, \mathrm{N}.$$

Replacing $\boldsymbol{d}(k)$ *by the noise term from* (3.3) *yields* Σ. □

Note that Σ represents a set membership of $\boldsymbol{\Xi}$, containing, among all, also the *true* coefficients $\boldsymbol{\Xi}$.

Then, inspired by the feedback law (3.5), the projection $\widetilde{\boldsymbol{\Xi}}$ is obtained by solving the optimization problem

$$\min_{\widetilde{\boldsymbol{\Xi}}_1, \text{invertible } \widetilde{\boldsymbol{\Xi}}_2, \boldsymbol{K}} \quad \left\| \widehat{\boldsymbol{\Xi}} - \widetilde{\boldsymbol{\Xi}} \right\|_2^2 \tag{3.8a}$$

$$\text{s.t.} \begin{cases} V(\boldsymbol{e}^+) \leq V(\boldsymbol{e}) \\ \boldsymbol{e}^+ = -\boldsymbol{K}\boldsymbol{e} + \left(\boldsymbol{\Xi} - \widetilde{\boldsymbol{\Xi}}\right) \begin{pmatrix} \mathbf{F}(\boldsymbol{x}) \\ \mathbf{G}(\boldsymbol{x})\widetilde{\boldsymbol{u}}(\boldsymbol{x}) \end{pmatrix} \\ \widetilde{\boldsymbol{u}}(\boldsymbol{x}) = \big(\widetilde{\boldsymbol{\Xi}}_2 \mathbf{G}(\boldsymbol{x})\big)^{-1} \big(-\widetilde{\boldsymbol{\Xi}}_1 \mathbf{F}(\boldsymbol{x}) - \boldsymbol{K}\boldsymbol{e} + \boldsymbol{x}_{\mathrm{r}}^+\big), \end{cases} \tag{3.8b}$$

where constraints (3.8b) need to be valid $\forall \boldsymbol{\Xi} \in \Sigma$. Since the variability of Σ only depends on the noise bound, the control law $\widetilde{\boldsymbol{u}}(\boldsymbol{x})$ is robust for all the admissible systems, which differ each other only by the noise variance. This ensures that, although the true $\boldsymbol{\Xi}$ is unknown, the Lyapunov function robustly decays for all the coefficients matrices within Σ, and, consequently, also for the true system.

3.4 The Proposed Online Data-Driven Control Algorithm

In this section, leveraging the result of Lemma 3.3.1, we first demonstrate that (3.8) can be effectively solved as an SDP. To this aim, the following lemma delivers a sufficient LMI condition for satisfying constraints (3.8b). Second, we provide the stability guarantees for the closed-loop dynamics in a concluding theorem.

Lemma 3.4.1 (*Sufficient LMI condition*)

If there exist matrices $\widetilde{\boldsymbol{M}}_1 \in \mathbb{R}^{n \times n_f}$ *and* $\widetilde{\boldsymbol{M}}_2 \in \mathbb{R}^{n \times n}$, *invertible matrix* $\widetilde{\boldsymbol{\Xi}}_2^{-1} \in \mathbb{R}^{n \times n}$, *and scalars* $\alpha_1, \dots, \alpha_{\mathrm{N}} \geq 0$ *such that the LMI constraint*

$$\begin{pmatrix} \boldsymbol{\Upsilon} & \begin{pmatrix} 0 \\ \boldsymbol{\Theta}(\boldsymbol{x}) \end{pmatrix} \\ \begin{pmatrix} 0 & \boldsymbol{\Theta}(\boldsymbol{x})^\top \end{pmatrix} & -1 \end{pmatrix} \preceq 0 \tag{3.9}$$

with

$$\boldsymbol{\Upsilon} = \begin{pmatrix} \boldsymbol{x}_r^+ (\boldsymbol{x}_r^+)^\top - \boldsymbol{e}^\top \mathbf{P} \boldsymbol{e} \mathbf{P}^{-1} & -\boldsymbol{x}_r^+ \boldsymbol{\Theta}(\boldsymbol{x})^\top \\ -\big(\boldsymbol{x}_r^+ \boldsymbol{\Theta}(\boldsymbol{x})^\top\big)^\top & 0 \end{pmatrix} - \sum_{k=1}^{\mathrm{N}} \alpha_k \boldsymbol{\Delta}(k)$$

and

$$\boldsymbol{\Theta}(\boldsymbol{x}) = \begin{pmatrix} \mathbf{F}(\boldsymbol{x}) \\ -\widetilde{\boldsymbol{M}}_1 \mathbf{F}(\boldsymbol{x}) - \widetilde{\boldsymbol{M}}_2 \boldsymbol{e} + \widetilde{\boldsymbol{\Xi}}_2^{-1} \boldsymbol{x}_r^+ \end{pmatrix} \tag{3.10}$$

holds, then (3.8b) *is satisfied* $\forall \boldsymbol{\Xi} \in \Sigma$, *and for* $\widetilde{\boldsymbol{\Xi}}_1 = \widetilde{\boldsymbol{\Xi}}_2 \widetilde{\boldsymbol{M}}_1$, $\widetilde{\boldsymbol{\Xi}}_2 = \left(\widetilde{\boldsymbol{\Xi}}_2^{-1}\right)^{-1}$, *and* $\boldsymbol{K} = \widetilde{\boldsymbol{\Xi}}_2 \widetilde{\boldsymbol{M}}_2$. □

Proof 3.4.1

To prove the claim, we first apply the Schur complement on (3.9) *resulting in the equivalent condition*

$$\begin{pmatrix} \boldsymbol{x}_r^+ (\boldsymbol{x}_r^+)^\top - \boldsymbol{e}^\top \mathbf{P} \boldsymbol{e} \mathbf{P}^{-1} & -\boldsymbol{x}_r^+ \boldsymbol{\Theta}(\boldsymbol{x})^\top \\ -\big(\boldsymbol{x}_r^+ \boldsymbol{\Theta}(\boldsymbol{x})^\top\big)^\top & \boldsymbol{\Theta}(\boldsymbol{x}) \boldsymbol{\Theta}(\boldsymbol{x})^\top \end{pmatrix} - \sum_{k=1}^{\mathrm{N}} \alpha_k \boldsymbol{\Delta}(k) \preceq 0.$$

Pre- and post-multiplying $\begin{pmatrix} \mathbf{I} & \boldsymbol{\Xi} \end{pmatrix}$ *and its transpose, respectively, yields*

$$\begin{aligned} &\begin{pmatrix} \mathbf{I} \\ \boldsymbol{\Xi}^\top \end{pmatrix}^\top \begin{pmatrix} \boldsymbol{x}_r^+ (\boldsymbol{x}_r^+)^\top - \boldsymbol{e}^\top \mathbf{P} \boldsymbol{e} \mathbf{P}^{-1} & -\boldsymbol{x}_r^+ \boldsymbol{\Theta}(\boldsymbol{x})^\top \\ -\big(\boldsymbol{x}_r^+ \boldsymbol{\Theta}(\boldsymbol{x})^\top\big)^\top & \boldsymbol{\Theta}(\boldsymbol{x}) \boldsymbol{\Theta}(\boldsymbol{x})^\top \end{pmatrix} \begin{pmatrix} \mathbf{I} \\ \boldsymbol{\Xi}^\top \end{pmatrix} \\ &\preceq \sum_{k=1}^{\mathrm{N}} \alpha_k \begin{pmatrix} \mathbf{I} \\ \boldsymbol{\Xi}^\top \end{pmatrix}^\top \boldsymbol{\Delta}(k) \begin{pmatrix} \mathbf{I} \\ \boldsymbol{\Xi}^\top \end{pmatrix}. \end{aligned}$$

By the construction of Σ *in* (3.7) *and* $\alpha_1, \ldots, \alpha_{\mathrm{N}} \geq 0$, *the right-hand side of the previous condition is negative definite for all* $\boldsymbol{\Xi} \in \Sigma$. *Therefore, we can imply that*

$$\star^\top \begin{pmatrix} \boldsymbol{x}_r^+ (\boldsymbol{x}_r^+)^\top - \boldsymbol{e}^\top \mathbf{P} \boldsymbol{e} \mathbf{P}^{-1} & -\boldsymbol{x}_r^+ \boldsymbol{\Theta}(\boldsymbol{x})^\top \\ -\big(\boldsymbol{x}_r^+ \boldsymbol{\Theta}(\boldsymbol{x})^\top\big)^\top & \boldsymbol{\Theta}(\boldsymbol{x}) \boldsymbol{\Theta}(\boldsymbol{x})^\top \end{pmatrix} \begin{pmatrix} \mathbf{I} \\ \boldsymbol{\Xi}^\top \end{pmatrix} \preceq 0 \tag{3.11}$$

$\forall \boldsymbol{\Xi} \in \Sigma$, *which corresponds to an S-procedure argument. To proceed, note that* (3.11) *can also be written as*

$$\star^\top \begin{pmatrix} \boldsymbol{x}_r^+ (\boldsymbol{x}_r^+)^\top - \boldsymbol{e}^\top \mathbf{P} \boldsymbol{e} \mathbf{P}^{-1} & \boldsymbol{x}_r^+ \\ (\boldsymbol{x}_r^+)^\top & 1 \end{pmatrix} \begin{pmatrix} \mathbf{I} \\ -\big(\boldsymbol{\Xi} \boldsymbol{\Theta}(\boldsymbol{x})\big)^\top \end{pmatrix} \preceq 0. \tag{3.12}$$

Moreover, we observe that $\boldsymbol{\Theta}(\boldsymbol{x}) = \boldsymbol{\Theta}\big(\boldsymbol{x}, \widetilde{\boldsymbol{u}}(\boldsymbol{x})\big)$ *as*

$$\begin{aligned} &- \widetilde{\boldsymbol{M}}_1 \mathbf{F}(\boldsymbol{x}) - \widetilde{\boldsymbol{M}}_2 \boldsymbol{e} + \widetilde{\boldsymbol{\Xi}}_2^{-1} \boldsymbol{x}_r^+ \\ &= \mathbf{G}(\boldsymbol{x}) \mathbf{G}(\boldsymbol{x})^{-1} \widetilde{\boldsymbol{\Xi}}_2^{-1} \big(- \widetilde{\boldsymbol{\Xi}}_2 \widetilde{\boldsymbol{M}}_1 \mathbf{F}(\boldsymbol{x}) - \widetilde{\boldsymbol{\Xi}}_2 \widetilde{\boldsymbol{M}}_2 \boldsymbol{e} + \boldsymbol{x}_r^+ \big) \\ &= \mathbf{G}(\boldsymbol{x}) \Big(\widetilde{\boldsymbol{\Xi}}_2 \mathbf{G}(\boldsymbol{x}) \Big)^{-1} \big(- \widetilde{\boldsymbol{\Xi}}_1 \mathbf{F}(\boldsymbol{x}) - \boldsymbol{K} \boldsymbol{e} + \boldsymbol{x}_r^+ \big). \end{aligned}$$

Since $\boldsymbol{x}^+ = \boldsymbol{\Xi} \boldsymbol{\Theta}\big(\boldsymbol{x}, \widetilde{\boldsymbol{u}}(\boldsymbol{x})\big)$, *condition* (3.12) *can be written as*

$$\star^\top \begin{pmatrix} \boldsymbol{x}_r^+ (\boldsymbol{x}_r^+)^\top - \boldsymbol{e}^\top \mathbf{P} \boldsymbol{e} \mathbf{P}^{-1} & \boldsymbol{x}_r^+ \\ (\boldsymbol{x}_r^+)^\top & 1 \end{pmatrix} \begin{pmatrix} \mathbf{I} \\ -(\boldsymbol{x}^+)^\top \end{pmatrix} \preceq 0.$$

Finally, since $\mathbf{P} \succ 0$, *we can apply the Dualization Lemma such that*

$$\begin{aligned} 0 \leq &\begin{pmatrix} \boldsymbol{x}^+ \\ 1 \end{pmatrix}^\top \begin{pmatrix} -\mathbf{P} & \mathbf{P} \boldsymbol{x}_r^+ \\ (\mathbf{P} \boldsymbol{x}_r^+)^\top & \boldsymbol{e}^\top \mathbf{P} \boldsymbol{e} - (\boldsymbol{x}_r^+)^\top \mathbf{P} \boldsymbol{x}_r^+ \end{pmatrix} \begin{pmatrix} \boldsymbol{x}^+ \\ 1 \end{pmatrix} \\ &= \boldsymbol{e}^\top \mathbf{P} \boldsymbol{e} - (\boldsymbol{x}^+ - \boldsymbol{x}_r^+)^\top \mathbf{P} (\boldsymbol{x}^+ - \boldsymbol{x}_r^+) = V(\boldsymbol{e}) - V(\boldsymbol{e}^+) \end{aligned}$$

is satisfied for all $\boldsymbol{\Xi} \in \Sigma$.

It remains to be argued that (3.9) *is an LMI condition. To this end, note that* $\boldsymbol{x}, \boldsymbol{x}_r$, *and* $\boldsymbol{x}_r^+$ *are known constants. Therefore, the nonlinearity* $\boldsymbol{\Theta}\big(\boldsymbol{x}, \widetilde{\boldsymbol{u}}(\boldsymbol{x})\big)$ *is evaluated at the current state. Thus,* $\boldsymbol{\Theta}(\boldsymbol{x})$ *is linear with respect to the optimization variables* $\widetilde{\boldsymbol{M}}_1, \widetilde{\boldsymbol{M}}_2$, *and* $\widetilde{\boldsymbol{\Xi}}_2^{-1}$. *Since condition* (3.9) *is additionally linear regarding to* $\boldsymbol{\Theta}(\boldsymbol{x})$, *then* (3.9) *is an LMI.* □

Lemma 3.4.1 establishes an LMI constraint to verify whether $V(\boldsymbol{e})$ decreases in the next time step, for all the coefficient matrices robustly within Σ. It also provides a necessary condition for $k = 1$. Interestingly, the nonlinear dynamics can be directly evaluated without any approximation.

Furthermore, to achieve a guaranteed decay rate of the Lyapunov function $V(\boldsymbol{e}^+) \leq \delta V(\boldsymbol{e})$ with some $0 < \delta < 1$, we replace $\boldsymbol{\Upsilon}$ in LMI (3.9) by

$$\begin{pmatrix} \boldsymbol{x}_r^+(\boldsymbol{x}_r^+)^\top - \delta \boldsymbol{e}^\top \mathbf{P} \boldsymbol{e} \mathbf{P}^{-1} & -\boldsymbol{x}_r^+ \boldsymbol{\Theta}(\boldsymbol{x})^\top \\ -\big(\boldsymbol{x}_r^+ \boldsymbol{\Theta}(\boldsymbol{x})^\top\big)^\top & 0 \end{pmatrix} - \sum_{k=1}^{\mathrm{N}} \alpha_k \boldsymbol{\Delta}(k). \tag{3.13}$$

Moreover, due to the uncertain coefficients $\boldsymbol{\Xi}$, it might be impossible to achieve $V(\boldsymbol{e}^+) \leq V(\boldsymbol{e})$ for small $\boldsymbol{e}$. To this end, we propose the relaxation

$$V(\boldsymbol{e}^+) \leq \begin{cases} V(\boldsymbol{e}) & \text{if } V(\boldsymbol{e}) \geq \rho \\ \rho & \text{else} \end{cases}$$

for some $\rho > 0$, leading (3.6) to be practically stable.

Finally, to make problem (3.8) effectively solvable via SDP, we propose to compute the inverse $\widehat{\boldsymbol{\Xi}}_2^{-1}$ according to Assumption 3.3.1 and $\widehat{\boldsymbol{M}}_1 = \widehat{\boldsymbol{\Xi}}_2^{-1} \widehat{\boldsymbol{\Xi}}_1$ for the solution of the SINDy regression. Subsequently, we solve the problem

$$\min_{\widetilde{\boldsymbol{M}}_1,\, \widetilde{\boldsymbol{M}}_2,\, \widetilde{\boldsymbol{\Xi}}_2^{-1},\, \alpha_1,\ldots,\alpha_{\mathrm{N}},\, \delta} \quad s \tag{3.14a}$$

$$\text{s.t.} \quad \begin{cases} \begin{pmatrix} sI & \boldsymbol{\Lambda}^\top \\ \boldsymbol{\Lambda} & I \end{pmatrix} \succeq 0 \\ (3.9), \text{ with } (3.10) \text{ and } (3.13) \\ \alpha_1, \ldots, \alpha_{\mathrm{N}} \geq 0 \\ 0 < \delta < 1 \end{cases} \tag{3.14b}$$

instead of (3.8), at each time step, where $\boldsymbol{\Lambda} = \big(\widehat{\boldsymbol{\Xi}}_2^{-1}\, \widehat{\boldsymbol{M}}_1\big) - \big(\widetilde{\boldsymbol{\Xi}}_2^{-1}\, \widetilde{\boldsymbol{M}}_1\big)$. By the Schur complement, the distance $\|\boldsymbol{\Lambda}\|_2^2$ is minimized by (3.14). To ensure that the solution of $\widetilde{\boldsymbol{\Xi}}_2^{-1}$ from (3.14) is invertible, we can add a small full-rank perturbation. Algorithm 3.1 summarizes the whole proposed DD approach.

We conclude this section with the following theorem, which provides the stability guarantee for the closed-loop system.

Theorem 3.4.1 (*Closed-loop stability*)

Under Assumption 3.3.1 and if the SDP (3.14) is solvable with invertible $\widetilde{\boldsymbol{\Xi}}_2^{-1}$ at each time step, then the tracking error dynamics

$$\boldsymbol{e}^+ = -\boldsymbol{K}\boldsymbol{e} + \Big(\boldsymbol{\Xi}_1 - \widetilde{\boldsymbol{\Xi}}_1\Big)\, \mathbf{F}(\boldsymbol{x}) + \Big(\boldsymbol{\Xi}_2 - \widetilde{\boldsymbol{\Xi}}_2\Big)\, \mathbf{G}(\boldsymbol{x}) \widetilde{\boldsymbol{u}}(\boldsymbol{x})$$

is asymptotically stable. □

Proof 3.4.2

By Lemma 3.4.1, the tracking error satisfies $V(\boldsymbol{e}^+) \leq \delta V(\boldsymbol{e})$ for the Lyapunov function $V(\boldsymbol{e}) = \boldsymbol{e}^\top \mathbf{P} \boldsymbol{e}$. Therefore, $V(\boldsymbol{e})$ is strictly decreasing, which implies asymptotic stability with respect to the reference trajectory $\boldsymbol{x}_r$. This also implies $V\big(\boldsymbol{x}[t]\big) \leq \delta^t V\big(\boldsymbol{x}[0]\big)$, and thus $\|\boldsymbol{e}[t]\|_2^2 \leq \frac{\lambda_{\max(\mathbf{P})}}{\lambda_{\min(\mathbf{P})}} \delta^t \|\boldsymbol{e}[0]\|_2^2$. Here, $\lambda_{\max}(\mathbf{P})$ and $\lambda_{\min}(\mathbf{P})$ denote the maximum and minimum eigenvalues of $\mathbf{P}$, respectively, and $\boldsymbol{x}[t], \boldsymbol{e}[t]$ the states and tracking error at time t, respectively. □

Algorithm 3.1 The Online Data-Driven Control Algorithm

Require: Basis functions $\boldsymbol{\Theta}(\boldsymbol{x},\boldsymbol{u})$ satisfying (3.2).
Require: Input-state measurements $\{\boldsymbol{u}(k),\boldsymbol{x}(k)\}_{k=1}^{\mathrm{N}+1}$
satisfying $\boldsymbol{x}(k+1)=\boldsymbol{\Xi}\boldsymbol{\Theta}(\boldsymbol{x}(k),\boldsymbol{u}(k))+\boldsymbol{d}(k)$.
Require: $\epsilon_d>0$ such that $\|\boldsymbol{d}(k)\|_2^2\leq\epsilon_d, k=1,\dots,\mathrm{N}$,
$\boldsymbol{\Delta}(k)\leftarrow$(3.7) for $k=1,\dots,\mathrm{N}$.
$\widehat{\boldsymbol{\Xi}}\leftarrow$ solving (3.4).
Require: Lyapunov function matrix $\mathbf{P}\succ 0$ and $\rho>0$.
Require: Time horizon $\mathrm{T}\in\mathbb{N}\cup\{\infty\}$.
$t\leftarrow 0$
while $t\leq\mathrm{T}$ **do**
$\boldsymbol{x}\leftarrow$ current state measurement
$\boldsymbol{x}_r\leftarrow$ current state reference
$\boldsymbol{x}_r^+\leftarrow$ next state reference
$\boldsymbol{e}\leftarrow\boldsymbol{x}-\boldsymbol{x}_r$
if $V(e)>\rho$ **then**
$\widetilde{\boldsymbol{M}}_1,\widetilde{\boldsymbol{M}}_2,\widetilde{\boldsymbol{\Xi}}_2^{-1}\leftarrow$ solving (3.14)
else
$\widetilde{\boldsymbol{M}}_1,\widetilde{\boldsymbol{M}}_2,\widetilde{\boldsymbol{\Xi}}_2^{-1}\leftarrow$ solving (3.14) with $\boldsymbol{e}^\top\mathbf{P}\boldsymbol{e}=\rho$
end if
$\widetilde{\boldsymbol{\Xi}}_2\leftarrow$ Inverse of $\widetilde{\boldsymbol{\Xi}}_2^{-1}$
$\widetilde{\boldsymbol{\Xi}}_1\leftarrow\widetilde{\boldsymbol{\Xi}}_2\widetilde{\boldsymbol{M}}_1$
$\boldsymbol{K}\leftarrow\widetilde{\boldsymbol{\Xi}}_2\widetilde{\boldsymbol{M}}_2$
$\boldsymbol{u}\leftarrow\big(\widetilde{\boldsymbol{\Xi}}_2\mathbf{G}(\boldsymbol{x})\big)^{-1}\big(-\widetilde{\boldsymbol{\Xi}}_1\mathbf{F}(\boldsymbol{x})-\boldsymbol{K}\boldsymbol{e}+\boldsymbol{x}_{\mathrm{r}}^+\big)$
$t\leftarrow t+1$
end while

3.5 Illustrative Examples

We validate Algorithm 3.1 by two examples. The former involves nonlinearities in $\mathbf{F}(\boldsymbol{x})$ and $\mathbf{G}(\boldsymbol{x})$, while the latter considers a pendulum-like dynamics, often examined in the related literature (see [3]). For both examples, $\mathrm{N}=150$ data samples are generated starting from a randomly-chosen initial condition $\boldsymbol{x}_0$, with a sine-like input signal $\boldsymbol{u}$, and an upper bound $\epsilon_d=10^{-2}$ for the disturbance $\boldsymbol{d}$. The controller performance is evaluated over a time horizon $\mathrm{T}=20$.

All the simulations are conducted in MATLAB®, on a notebook with a 1.70 GHz Intel Core i7-1255U CPU and 16 GB RAM. The optimization problem (3.14) is solved by YALMIP [29] with the SDP solver MOSEK® [30] (v. 10.2).

3.5.1 Example 1

For the first example, the following nonlinear dynamics

$$\begin{cases} x_1^+ = x_1 + 0.1x_2 + u_1 + d_1 \\ x_2^+ = 2\sin(x_1) + 0.9x_2 + \big(0.01+\sin^2(x_1)\big)u_2 + d_2 \end{cases}$$

is parameterized through the state and input basis functions $\mathbf{F}(\boldsymbol{x})=\begin{pmatrix}x_1 & x_2 & \sin(x_1)\end{pmatrix}^\top$ and $\mathbf{G}(\boldsymbol{x})=\begin{pmatrix}1 & 0\\ 0 & 0.01+\sin^2(x_1)\end{pmatrix}$. The true coefficient matrices are ($n_f=3$):

$$\boldsymbol{\Xi}_1=\begin{pmatrix}1 & 0.1 & 0\\ 0 & 0.9 & 2\end{pmatrix},\quad \boldsymbol{\Xi}_2=\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}.$$

Sample data are generated from the initial condition $\boldsymbol{x}_0=\begin{pmatrix}0.4886 & -3.5572\end{pmatrix}^\top$. With a Pareto term $\gamma=0.1$, Fig. 3.1 (a1) shows the open-loop simulation for the SINDy

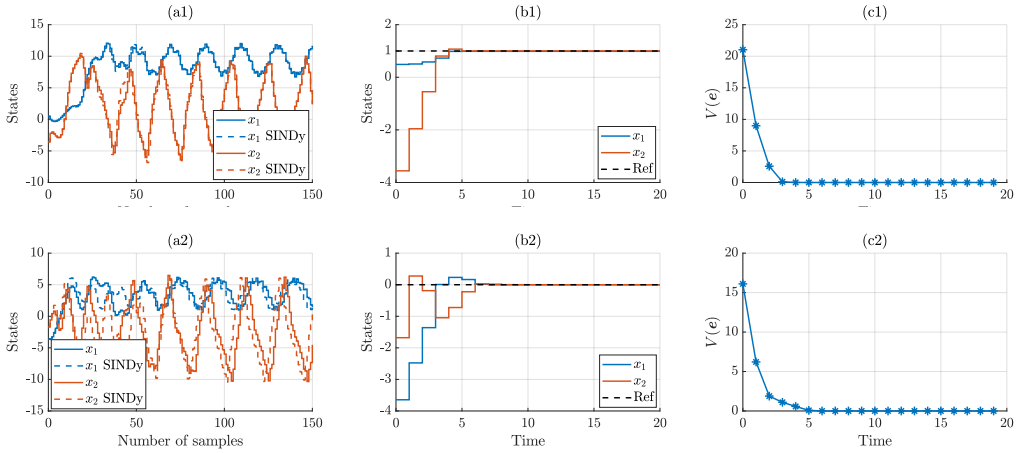

Figure 3.1: Open-loop simulation of true system (corresponding to data samples) and SINDy regression (a1)-(a2), closed-loop system response (b1)-(b2), and Lyapunov function (c1)-(c2) for Examples 1 and 2, respectively.

regression

$$\widehat{\boldsymbol{\Xi}}_1 = \begin{pmatrix} 1 & 0.1 & 0 \\ 0 & 0.8999 & 1.9976 \end{pmatrix}, \quad \widehat{\boldsymbol{\Xi}}_2 = \begin{pmatrix} 1 & 0 \\ 0 & 0.8941 \end{pmatrix},$$

resulting in a percentage mean square error (MSE) around 0.11%. $\mathbf{P} = \mathbf{I}$ and $\rho = 0.3$ are the Lyapunov matrix and the relaxing parameter, respectively. The controlled system is evaluated for a step reference signal. Figures 3.1 (b1)-(c1) indicate respectively the closed-loop step response and the Lyapunov function over time.

3.5.2 Example 2

Example 2 considers the second-order nonlinear dynamics

$$\begin{cases} x_1^+ = x_1 + 0.1x_2 + u_1 + d_1 \\ x_2^+ = 2\sin(x_1) + 0.9x_2 + u_2 + d_2, \end{cases} \tag{3.15}$$

parameterized through the state and input basis functions $\mathbf{F}(\boldsymbol{x}) = \begin{pmatrix} x_1 & x_2 & \sin(x_1) \end{pmatrix}^\top$ and $\mathbf{G}(\boldsymbol{x}) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. The true coefficient matrices are ($n_f = 3$):

$$\boldsymbol{\Xi}_1 = \begin{pmatrix} 1 & 0.1 & 0 \\ 0 & 0.9 & 2 \end{pmatrix}, \quad \boldsymbol{\Xi}_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Sample data are drawn from the initial condition $\boldsymbol{x}_0 = \begin{pmatrix} -3.6452 & -1.6788 \end{pmatrix}^\top$. For a Pareto term $\gamma = 0.001$, Fig. 3.1 (a2) shows the open-loop simulation of the identified dynamics with

$$\widehat{\boldsymbol{\Xi}}_1 = \begin{pmatrix} 1 & 0.1 & 0 \\ 0.0547 & 0.9024 & 1.9075 \end{pmatrix}, \quad \widehat{\boldsymbol{\Xi}}_2 = \begin{pmatrix} 1 & 0 \\ 0.7270 & 0.2887 \end{pmatrix}.$$

Conversely to Example 1, the percentage MSE of the regression is around 10.46%. The positive-definite matrix

$$\mathbf{P} = \begin{pmatrix} 3.7816 & 1.0095 \\ 1.0095 & 1.1980 \end{pmatrix} \tag{3.16}$$

is used in the Lyapunov function constraint. Note that (3.16) is obtained as the solution of a benchmark controller inspired by [22] (cf. Th. 12.4.2), employed to compare the results of our method. Specifically, the control gain $\mathbf{K}$ of the linear state feedback controller $\boldsymbol{u} = \mathbf{K}\boldsymbol{x}$ proposed in [22] is obtained for regulating the Jacobian-linearization of (3.15).

However, it is not able to achieve closed-loop stability, in contrast to the nonlinear control scheme from our method, whose results are shown in Figs. 3.1 (b2)-(c2). More in detail, the former indicates that the control signal computed by Algorithm 3.1 stabilizes the system even with a large regression error, whilst the latter displays the Lyapunov function behaviour.

3.6 Conclusions

In this chapter, we propose a Data-Driven (DD) control method for unknown input-affine nonlinear systems. First we require a library of candidate basis functions to build a Subspace Identification of Nonlinear Dynamics (SINDy) regression offline; second, we solve online a Semidefinite Programming (SDP) problem at each time step. The closed-loop stability is guaranteed by constraining the decay of a Lyapunov function over time within the SDP. The SDP also adjusts the regression estimate by projecting it on a set, where all the systems within it result in a state feedback linearization that robustly ensures the decay of the Lyapunov function for all the systems consistent with the noisy data. The effectiveness of the theoretical results is validated both for stabilization and reference tracking of two different examples, outperforming a benchmark literature-inspired controller.

To address feasibility issues in solving the proposed SDP problem, a theoretical analysis of the conditions that guarantee its solvability could be an important open issue to explore. For non-constant reference trajectories, improvements in this sense could include using a reference governor or relaxing the Lyapunov condition based on trajectory changes. Alternatives might involve the adjustment of the regression matrix upon its previous value (like in adaptive control).

References

[1] Tang, W. and Daoutidis, P., "Data-driven control: Overview and perspectives," in *2022 American Control Conf. (ACC)*, IEEE, 2022, pp. 1048–1064.

[2] Ljung, L., "System identification," in *Signal analysis and prediction*, Springer, 1998, pp. 163–173.

[3] Martin, T., Schön, T. B., and Allgöwer, F., "Guarantees for data-driven control of nonlinear systems using semidefinite programming: A survey," *Annual Reviews in Control*, p. 100 911, 2023.

[4] Soudbakhsh, D., Annaswamy, A. M., Wang, Y., *et al.*, "Data-driven control: Theory and applications," in *2023 American Control Conf. (ACC)*, IEEE, 2023, pp. 1922–1939.

[5] Willems, J. C., Rapisarda, P., Markovsky, I., and De Moor, B. L., "A note on persistency of excitation," *Systems & Control Letters*, vol. 54, no. 4, pp. 325–329, 2005.

[6] Khalil, H. K., *Nonlinear Systems*. Prentice Hall, 2002.

[7] Astolfi, A., "Nonlinear adaptive control," in *Encyclopedia of Systems and Control*, Springer, 2021, pp. 1467–1472.

[8] Bozza, A., Askari, B., Cavone, G., Carli, R., and Dotoli, M., "An Adaptive Model Predictive Control Approach for Position Tracking and Force Control of a Hydraulic Actuator," in *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, 2022, pp. 1029–1034. DOI: 10.1109/CASE49997.2022.9926645.

[9] Novara, C., Fagiano, L., and Milanese, M., "Direct feedback control design for nonlinear systems," *Automatica*, vol. 49, no. 4, pp. 849–860, 2013.

[10] Vandenberghe, L. and Boyd, S., "Semidefinite programming," *SIAM review*, vol. 38, no. 1, pp. 49–95, 1996.

[11] Xie, Y., Berberich, J., and Allgöwer, F., "Data-driven min-max mpc for linear systems," in *2024 American Control Conference (ACC)*, IEEE, 2024, pp. 3184–3189.

[12] Martin, T. and Allgöwer, F., "Data-driven system analysis of nonlinear systems using polynomial approximation," *IEEE Trans. Autom. Control*, 2023.

[13] Verhoek, C., Abbas, H. S., and Tóth, R., "Direct data-driven lpv control of nonlinear systems: An experimental result," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 2263–2268, 2023.

[14] Strässer, R., Schaller, M., Worthmann, K., Berberich, J., and Allgöwer, F., "Koopman-based feedback design with stability guarantees," *IEEE Trans. Autom. Control*, 2024.

[15] De Persis, C., Gadginmath, D., Pasqualetti, F., and Tesi, P., "Data-driven feedback linearization with complete dictionaries," in *2023 62nd IEEE Conf. on Decision and Control (CDC)*, IEEE, 2023, pp. 3037–3042.

[16] Gadginmath, D., Krishnan, V., and Pasqualetti, F., "Data-driven feedback linearization using the koopman generator," *IEEE Trans. Autom. Control*, 2024.

[17] Umlauft, J. and Hirche, S., "Feedback linearization based on gaussian processes with event-triggered online learning," *IEEE Trans. Autom. Control*, vol. 65, no. 10, pp. 4154–4169, 2019.

[18] Dai, T. and Sznaier, M., "Nonlinear data-driven control via state-dependent representations," in *2021 60th IEEE Conf. on Decision and Control (CDC)*, IEEE, 2021, pp. 5765–5770.

[19] Dai, T. and Sznaier, M., "A semi-algebraic optimization approach to data-driven control of continuous-time nonlinear systems," *IEEE Control Syst. Lett.*, vol. 5, no. 2, pp. 487–492, 2020.

[20] Strässer, R., Berberich, J., and Allgöwer, F., "Data-driven control of nonlinear systems: Beyond polynomial dynamics," in *2021 60th IEEE Conf. on Decision and Control (CDC)*, IEEE, 2021, pp. 4344–4351.

[21] Strässer, R., Berberich, J., and Allgöwer, F., "Robust data-driven control for nonlinear systems using the koopman operator," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 2257–2262, 2023.

[22] Guo, M., De Persis, C., and Tesi, P., "Data-driven stabilization of nonlinear systems via taylor's expansion," in *Hybrid and Networked Dynamical Systems: Modeling, Analysis and Control*, Springer, 2024, pp. 273–299.

[23] Martin, T., Schön, T. B., and Allgöwer, F., "Gaussian inference for data-driven state-feedback design of nonlinear systems," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 4796–4803, 2023.

[24] Fasel, U., Kaiser, E., Kutz, J. N., Brunton, B. W., and Brunton, S. L., "Sindy with control: A tutorial," in *2021 60th IEEE Conf. on Decision and Control (CDC)*, IEEE, 2021, pp. 16–21.

[25] Umlauft, J., Pöhler, L., and Hirche, S., "An uncertainty-based control lyapunov approach for control-affine systems modeled by gaussian process," *IEEE Control Syst. Lett.*, vol. 2, no. 3, pp. 483–488, 2018.

[26] Alsalti, M., Berberich, J., Lopez, V. G., Allgöwer, F., and Müller, M. A., "Data-based system analysis and control of flat nonlinear systems," in *2021 60th IEEE Conf. on Decision and Control (CDC)*, IEEE, 2021, pp. 1484–1489.

[27] Slotine, J.-J. E. and Karl Hedrick, J., "Robust input-output feedback linearization," *International Journal of control*, vol. 57, no. 5, pp. 1133–1139, 1993.

[28] Caverly, R. J. and Forbes, J. R., "Lmi properties and applications in systems, stability, and control theory," *arXiv preprint arXiv:1903.08599*, 2019.

[29] Löfberg, J. “Knuth: Computers and typesetting.” (), [Online]. Available: https://yalmip.github.io/. (accessed: July 2024).

[30] Company, M. “Mosek: A tool for solving mathematical optimization problems.” (), [Online]. Available: https://www.mosek.com/. (accessed: July 2024).

Part II:

Data-Driven Adaptive Control and Fault Detection for Process Performance Preservation

Chapter 4

Adaptive MPC for Hydraulic Servo Actuators with Flow Control Valves

Abstract

This chapter presents an *Adaptive Model Predictive Control* (AMPC) for the position tracking and force control of two-chamber *Hydraulic Servo Actuators* (HSAs) based on *flow control valves*. Due to their nonlinear dynamics, the iterative linearization paradigm is employed to approximate the system with a Linear Time-Varying model. Such a representation is used as the internal plant model of the predictive controller involved to effectively make predictions on the system state. The effectiveness of the proposed AMPC architecture is shown through numerical simulations addressing the control of a real HSA in different scenarios. Finally, a comparative analysis on several values of sampling time, prediction and control horizon is carried out for the sake of investigating the effect of the parameters tuning on the performance of the closed-loop control system.

Contents

4.1 Introduction

Hydraulic systems are widely involved in many modern industrial applications. The core components of such systems are represented by hydraulic actuators, which play a crucial role in various industrial applications [1]–[4] due to their high force and torque coupled with a little size-to-power ratio. Since the 1950s, their control has shift from purely mechanical to the so-called servo valves, leading to the development of *Hydraulic Servo Actuators* (HSAs). The basic structure of a HSA includes is shown in Fig. 4.1. It comprises one or more pumps for the hydraulic power transmission, a tank for the oil reservoir, at least a high and a low pressure line, linear pistons and/or rotary motors for the mechanical actuation, servo valves for controlling the flow-rate direction, the quantity, and the pressure delivered from the pump to the actuator chambers, sensors, and other interconnected components (e.g., pipelines). Nevertheless, these features make the HSAs dynamics highly nonlinear, heavily complicating the development of high-performance closed loop controllers, also because of control inputs saturation and valves functioning.

HSAs can use *analogical* (also known as proportional) or *digital* servo valves. The former are based on electrical torque motors able to ensure accurate feedback control, making such systems compliant to the high-efficiency requirements of the most recent applications [5]–[7], but also susceptible to their complex structure and sensitivity [8]. Conversely, the latter are characterized by a switched-based operation (i.e., ON or OFF). They have emerged in the last decade as a promising alternative in various industrial applications [9]–[11], thanks to their benefits in energy saving, fault tolerance, cost-effectiveness, and reliability [12], [13]. Moreover, both analogical and digital servo valves

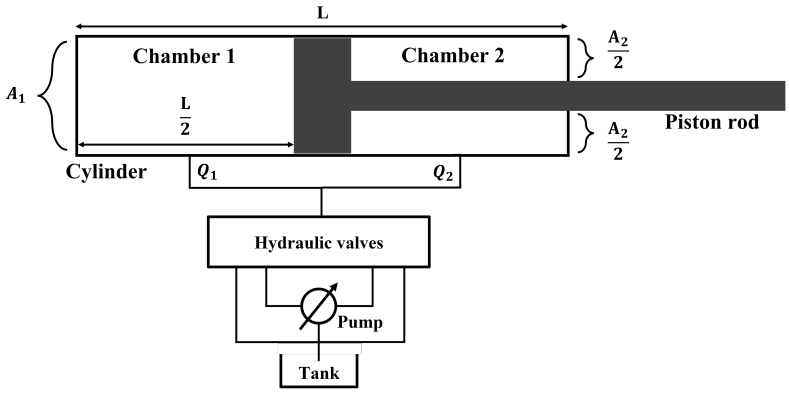

Figure 4.1: Schematic representation of the HSA.

can be distinguished between *flow control* and *pressure control* valves. The first allow regulating the volume of the oil passing through the orifice over a specific time period, while the second permit to maintain a constant value of chambers pressure.

In the related literature, different types of controllers have been proposed to ensure robustness and stability of HSAs. These comprise state feedback control, robust control [14], [15], adaptive control [16], combinations of robust and adaptive control [17], and modern nonlinear control strategies [18]. Traditionally, the most used approach is linear control, including feedback linearization [19] and classic Proportional-Integral-Derivative (PID) controllers [20]. Examples in this regard show that PID controllers are appropriate for reaching the reference signal, but can have limitations in ensuring good tracking performance. Differently, adaptive and *adaptive robust control* aim at dealing with model uncertainties. Within this context, various solutions have been proposed in the literature to control HSAs in [21], [22]. Furthermore, among the various robust control techniques, sliding mode is a well-known variable structure method, used in [23], [24] to minimize the position tracking error of a pneumatic system. A recent advancement in adaptive control for nonlinear systems is *Adaptive Model Predictive Control* (AMPC) [25], [26] that, differently from the previously discussed approaches, allows the multi-variable control in a proactive fashion. Although it is largely appreciated for nonlinear control problems, few works have investigated application of AMPC to HSAs. In [27], for example, a particular type of AMPC is proposed for controlling a HSA used for the regulation of an helicopter torque gearbox.

4.1.1 Chapter Contribution

In this chapter we focus on HSAs based on flow control valves. Specifically, an AMPC approach is proposed for tracking both the position and force of two-chamber HSAs. The actuator is modeled as a Multiple-Input Multiple-Output (MIMO) system, with two flow rates entering the piston chambers as the twofold input, and the piston position and force as the double output. The proposed AMPC is based on a Linear Time-Varying (LTV) Model Predictive Control (MPC) approach able to properly adapt its internal plant model and effectively make predictions on the system state. To this aim, at each iteration of the control algorithm, the plant model is approximated with a Linear Time-Invariant (LTI) system in order to properly update the AMPC controller. In the related literature, position and force are generally individually controlled in HSA. Instead, in this chapter we present an effective control for both, by provisioning the flow rates for both chambers of the actuator.

4.2 Hydraulic Servo Actuator Model

The examined HSA is composed of a cylinder with a single-rod linear piston having one degree of freedom and two chambers, an oil tank, a fixed-displacement pump, and a set of hydraulic valves for the oil provisioning to each chamber (Fig. 4.1).

The HSA model can be described with the following general formulation:

$$\begin{cases} \dot{x}_i(t) = f_i(\boldsymbol{x}(t), \boldsymbol{u}(t)) & \forall i = 1, \dots, n \\ y_j(t) = g_j(\boldsymbol{x}(t), \boldsymbol{u}(t)) & \forall j = 1, \dots, n_p \end{cases} \tag{4.1}$$

where $f_i(\boldsymbol{x}(t), \boldsymbol{u}(t))$ $(\forall i = 1, \dots, n)$ and $g_j(\boldsymbol{x}(t), \boldsymbol{u}(t))$ $(\forall j = 1, \dots, n_p)$ are respectively the state and output functions, $\boldsymbol{x}(t) = (x_1(t), x_2(t), x_3(t), x_4(t))^\top$ is the n-dimensional vector of the state variables (SVs), $\boldsymbol{u}(t) = (Q_1(t), Q_2(t))^\top$ is the n_u-dimensional vector of the input or manipulated variables (MVs), and $\boldsymbol{y}(t) = (y_1(t), y_2(t))^\top$ is the n_p-dimensional vector of the output variables (OVs), with $n = 4$, $n_u = 2$, and $n_p = 2$. Specifically, the state variables x_1, x_2, x_3, and x_4 represent the piston position and velocity, and the pressures in chamber 1 and chamber 2, respectively; the input variables $Q_1(t)$ and $Q_2(t)$ are the flow rate entering chamber 1 and chamber 2, respectively; the output variables $y_1(t)$ and $y_2(t)$ indicate the piston position and force, respectively.

Following [15], [28], the four nonlinear state functions are defined as:

$$\begin{cases} f_1(\boldsymbol{x}(t), \boldsymbol{u}(t)) = x_2(t) & \text{(4.2a)} \\ f_2(\boldsymbol{x}(t), \boldsymbol{u}(t)) = \dfrac{1}{\mathrm{M}}\Big(-\mathrm{b_m} x_2(t) + \mathrm{A}_1 x_3(t) - \mathrm{A}_2 x_4(t)\Big) & \text{(4.2b)} \\ f_3(\boldsymbol{x}(t), \boldsymbol{u}(t)) = \dfrac{\mathrm{E}}{\mathrm{A}_1 x_1(t)}\Big(-\mathrm{A}_1 x_2(t) + Q_1(t)\Big) & \text{(4.2c)} \\ f_4(\boldsymbol{x}(t), \boldsymbol{u}(t)) = \dfrac{\mathrm{E}}{\mathrm{V_{tot}} - \mathrm{A}_1 x_1(t)}\Big(\mathrm{A}_2 x_2(t) + Q_2(t)\Big). & \text{(4.2d)} \end{cases}$$

The mechanical parameters considered in the model above are the total length L of the cylinder, the mass M of the piston, the two cross-sectional areas A_1 and A_2 of chamber 1 and 2, respectively, and the total volume of the cylinder $V_{tot} = \mathrm{A}_1 \frac{L}{2} + \mathrm{A}_2 \frac{L}{2}$. Conversely, the hydraulic parameters are the viscous coefficient $\mathrm{b_m}$ and the bulk modulus of the fluid compressibility E. For simplicity, no additional load force and friction are considered in the model.

Regarding the output equations, the position is directly represented by x_1, while the force is modeled as the derivative of the piston velocity –represented by x_2– with respect to time:

$$F = \mathrm{M}\, \dot{x}_2(t). \tag{4.3}$$

Hence, by substituting (4.2b) in (4.1) for $i = 2$ and, in turn, in (4.3), the following two linear output equations are obtained:

$$\begin{aligned} g_1(\boldsymbol{x}(t), \boldsymbol{u}(t)) &= x_1(t) && \text{(4.4a)} \\ g_2(\boldsymbol{x}(t), \boldsymbol{u}(t)) &= -\mathrm{b_m} x_2(t) + \mathrm{A}_1 x_3(t) - \mathrm{A}_2 x_4(t). && \text{(4.4b)} \end{aligned}$$

Finally, by substituting ((4.2a))-((4.2d)) and ((4.4a))-((4.4b)) in ((4.1)), the complete nonlinear model of the HSA can be expressed as follows:

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = \frac{1}{\mathrm{M}}\Big(-\mathrm{b_m} x_2(t) + \mathrm{A}_1 x_3(t) - \mathrm{A}_2 x_4(t)\Big) \\ \dot{x}_3(t) = \frac{\mathrm{E}}{\mathrm{A}_1 x_1(t)}\Big(-\mathrm{A}_1 x_2(t) + Q_1(t)\Big) \\ \dot{x}_4(t) = \frac{\mathrm{E}}{V_{tot} - \mathrm{A}_1 x_1(t)}\Big(\mathrm{A}_2 x_2(t) + Q_2(t)\Big) \\ y_1(t) = x_1(t) \\ y_2(t) = -\mathrm{b_m} x_2(t) + \mathrm{A}_1 x_3(t) - \mathrm{A}_2 x_4(t). \end{cases} \tag{4.5}$$

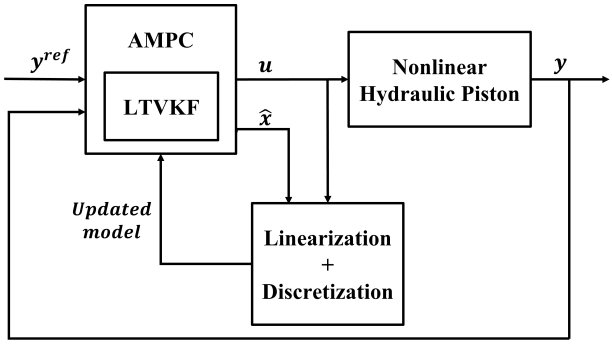

Figure 4.2: Architecture of the proposed AMPC architecture.

4.3 The Proposed Adaptive MPC

For controlling the nonlinear model of the HSA formulated in Section 4.2, the AMPC architecture shown in Fig. 4.2 is proposed. By taking inspiration from [29], at each iteration of the control algorithm the nonlinear model of the HSA is linearized at the current operating conditions and discretized at a given sampling rate. Then, the AMPC-based controller is properly updated, based on a Linear Time-Varying Kalman Filter (LTVKF). In this sense, an MPC able to adapt itself to the plant changes is obtained, thus allowing to make predictions accurately.

4.3.1 System Linearization

After linearization, the nonlinear system defined in (4.5) can be rewritten in accordance with the following linear state-space model structure:

$$\begin{cases} \boldsymbol{\Delta}\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{\Delta}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{\Delta}\boldsymbol{u}(t) \\ \boldsymbol{\Delta}\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{\Delta}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{\Delta}\boldsymbol{u}(t) \end{cases} \tag{4.6}$$

where $\boldsymbol{\Delta}\boldsymbol{x}(t) = \boldsymbol{x}(t) - \overline{\boldsymbol{x}}$, $\boldsymbol{\Delta}\boldsymbol{u}(t) = \boldsymbol{u}(t) - \overline{\boldsymbol{u}}$, and $\boldsymbol{\Delta}\boldsymbol{y}(t) = \boldsymbol{y}(t) - \overline{\boldsymbol{y}}$ indicate the deviations of the state, input, and output vectors from the nominal operating condition defined by the nominal vectors $\overline{\boldsymbol{x}}$, $\overline{\boldsymbol{u}}$, and $\overline{\boldsymbol{y}}$, respectively. The state-space matrices in (4.6) are defined as follows:

$$\begin{aligned} \boldsymbol{A} &= \frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{x}}\bigg|_{(\overline{\boldsymbol{x}},\overline{\boldsymbol{u}})} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{\mathrm{b_m}}{\mathrm{M}} & \frac{\mathrm{A_1}}{\mathrm{M}} & -\frac{\mathrm{A_2}}{\mathrm{M}} \\ -\frac{\mathrm{E}(\overline{Q}_1 - \mathrm{A_1}\overline{x}_2)}{\mathrm{A_1}\overline{x}_1^2} & -\frac{\mathrm{E}}{\overline{x}_1} & 0 & 0 \\ \frac{\mathrm{A_1E}(\overline{Q}_2 + \mathrm{A_2}\overline{x}_2)}{(V_{tot} - \mathrm{A_1}\overline{x}_1)^2} & -\frac{\mathrm{E}}{\overline{x}_1} & 0 & 0 \end{pmatrix}, \\ \boldsymbol{B} &= \frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{u}}\bigg|_{(\overline{\boldsymbol{x}},\overline{\boldsymbol{u}})} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ \frac{\mathrm{E}}{\mathrm{A_1}\overline{x}_1} & 0 \\ 0 & \frac{\mathrm{E}}{V_{tot} - \mathrm{A_1}\overline{x}_1} \end{pmatrix}, \end{aligned} \tag{4.7}$$

$$\boldsymbol{C} = \frac{\partial \boldsymbol{g}(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{x}}\bigg|_{(\overline{\boldsymbol{x}},\overline{\boldsymbol{u}})} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -\mathrm{b_m} & \mathrm{A_1} & -\mathrm{A_2} \end{pmatrix}, \qquad \boldsymbol{D} = \frac{\partial \boldsymbol{g}(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{u}}\bigg|_{(\overline{\boldsymbol{x}},\overline{\boldsymbol{u}})} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

where $\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}) = (f_1(\boldsymbol{x},\boldsymbol{u}), \ldots, f_n(\boldsymbol{x},\boldsymbol{u}))^\top$ and $\boldsymbol{g}(\boldsymbol{x},\boldsymbol{u}) = (g_1(\boldsymbol{x},\boldsymbol{u}), \ldots, g_{n_p}(\boldsymbol{x},\boldsymbol{u}))^\top$ are the state and the output functions defined in (4.2a)-(4.2d) and (4.4a)-(4.4b), respectively.

Since the AMPC-based controller uses the LTVKF as state observer to update its controller states, the plant should be completely observable. Hence, the rank of the Kalman observability matrix $\boldsymbol{K}_{\mathrm{obs}} = [\boldsymbol{C}, \boldsymbol{CA}, \boldsymbol{CA}^2, \boldsymbol{CA}^3]^\top$ is evaluated. However, it holds that $\mathrm{rank}(\boldsymbol{K}_{\mathrm{obs}}) = 3 < n$, thus implying that the linearized system ((4.6))-((4.7))

is not completely observable. This requires its reduction to its minimal form through the complete Kalman decomposition procedure, which leads to the following equivalent system representation:

$$\begin{cases} \boldsymbol{\Delta}\dot{\boldsymbol{x}}_{\mathrm{m}}(t) = \boldsymbol{A}_{\mathrm{m}}\boldsymbol{\Delta}\boldsymbol{x}_{\mathrm{m}}(t) + \boldsymbol{B}_{\mathrm{m}}\boldsymbol{\Delta}\boldsymbol{u}(t) \\ \boldsymbol{\Delta}\boldsymbol{y}(t) = \boldsymbol{C}_{\mathrm{m}}\boldsymbol{\Delta}\boldsymbol{x}_{\mathrm{m}}(t) + \boldsymbol{D}_{\mathrm{m}}\boldsymbol{\Delta}\boldsymbol{u}(t) \end{cases} \tag{4.8}$$

where $\boldsymbol{x}_{\mathrm{m}} = (x_{\mathrm{m}_1}, x_{\mathrm{m}_2}, x_{\mathrm{m}_3})^\top$ is the state vector of the minimal form with $\mathrm{m} = 3$ components. The new operating point is thus described by the couple $(\overline{\boldsymbol{x}}_{\mathrm{m}}, \overline{\boldsymbol{u}})$ and accordingly the new deviation from the nominal operating state is $\boldsymbol{\Delta}\boldsymbol{x}_{\mathrm{m}}(t) = \boldsymbol{x}_{\mathrm{m}}(t) - \overline{\boldsymbol{x}}_{\mathrm{m}}$. Finally, the matrices of the minimal form in (4.8) are defined as:

$$\boldsymbol{A}_{\mathrm{m}} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & -\frac{\mathrm{b_m}}{\mathrm{M}} & \frac{\mathrm{A_1}}{\mathrm{M}} + \frac{\mathrm{A_2^2}}{\mathrm{MA_1}} \\ -\frac{\mathrm{A_2E}(\overline{Q}_2 + \mathrm{A_2}\overline{x}_{\mathrm{m}_2})}{(V_{tot} - \mathrm{A_1}\overline{x}_{\mathrm{m}_1})^2} - \frac{\mathrm{E}(\overline{Q}_1 - \mathrm{A_1}\overline{x}_{\mathrm{m}_2})}{\mathrm{A_1}\overline{x}_{\mathrm{m}_1}^2} & \frac{\mathrm{A_2E}}{\mathrm{A_1}\overline{x}_{\mathrm{m}_1}} - \frac{\mathrm{E}}{\overline{x}_{\mathrm{m}_1}} & 0 \end{pmatrix},$$

$$\boldsymbol{B}_{\mathrm{m}} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ \frac{\mathrm{E}}{\mathrm{A_1}\overline{x}_{\mathrm{m}_1}} & -\frac{\mathrm{A_2E}}{\mathrm{A_1}(V_{tot} - \mathrm{A_1}\overline{x}_{m_1})} \end{pmatrix}, \quad \boldsymbol{C}_{\mathrm{m}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \mathrm{b_m} & \mathrm{A_1} + \frac{\mathrm{A_2^2}}{\mathrm{A_1}} \end{pmatrix}, \quad \boldsymbol{D}_{\mathrm{m}} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}. \tag{4.9}$$

4.3.2 Adaptive MPC Designing

At each time step k the linearized continuous-time system ((4.8))-((4.9)) is first discretized with a sampling time T_s to provide a proper discrete-time state-space model for its prediction over the whole prediction horizon, thus resulting in:

$$\begin{cases} \boldsymbol{\Delta}\boldsymbol{x}_{\mathrm{m}}(h) = \boldsymbol{A}_{\mathrm{k}}\boldsymbol{\Delta}\boldsymbol{x}_{\mathrm{m}}(h-1) + \boldsymbol{B}_{\mathrm{k}}\boldsymbol{\Delta}\boldsymbol{u}(h-1) \\ \boldsymbol{\Delta}\boldsymbol{y}(h) = \boldsymbol{C}_{\mathrm{k}}\boldsymbol{\Delta}\boldsymbol{x}_{\mathrm{m}}(h) + \boldsymbol{D}_{\mathrm{k}}\boldsymbol{\Delta}\boldsymbol{u}(h) \end{cases} \tag{4.10}$$

where $\boldsymbol{\Delta}\boldsymbol{x}_{\mathrm{m}}(h-1)$ and $\boldsymbol{\Delta}\boldsymbol{x}_{\mathrm{m}}(h)$ are respectively the state deviation and the incremental state deviation from the nominal state, $\boldsymbol{\Delta}\boldsymbol{y}(h)$ is the output incremental deviation from the nominal output, and $\boldsymbol{\Delta}\boldsymbol{u}(h-1)$ is the input deviation from the nominal input:

$$\boldsymbol{\Delta}\boldsymbol{x}_{\mathrm{m}}(h) = \boldsymbol{x}_{\mathrm{m}}(h) - \overline{\boldsymbol{x}}_{\mathrm{m}} \tag{4.11a}$$

$$\boldsymbol{\Delta}\boldsymbol{y}(h) = \boldsymbol{y}(h) - \overline{\boldsymbol{y}} \tag{4.11b}$$

$$\boldsymbol{\Delta}\boldsymbol{u}(h) = \boldsymbol{u}(h) - \overline{\boldsymbol{u}}. \tag{4.11c}$$

The discrete state matrix $\boldsymbol{A}_{\mathrm{k}}$ and the input matrix $\boldsymbol{B}_{\mathrm{k}}$ are obtained from $\boldsymbol{A}_{\mathrm{m}}$ and $\boldsymbol{B}_{\mathrm{m}}$ by the Euler method as follows:

$$\boldsymbol{A}_{\mathrm{k}} = e^{\boldsymbol{A}_{\mathrm{m}}T_s} \qquad \boldsymbol{B}_{\mathrm{k}} = \int_{(k-1)T_s}^{kT_s} \left(e^{\boldsymbol{A}_{\mathrm{m}}(kT_s - \tau)}\boldsymbol{B}_{\mathrm{m}}\right) d\tau \tag{4.12}$$

while the discrete output matrix $\boldsymbol{C}_{\mathrm{k}}$ and the feedforward matrix $\boldsymbol{D}_{\mathrm{k}}$ are equal to the continuous case, i.e., $\boldsymbol{C}_{\mathrm{k}} = \boldsymbol{C}_{\mathrm{m}},\ \boldsymbol{D}_{\mathrm{k}} = \boldsymbol{D}_{\mathrm{m}}$.

Second, as for the LTVKF, both the gain $\boldsymbol{L}_{\mathrm{k}}$ (i.e., the solution of the discrete Riccati equation) and $\boldsymbol{M}_{\mathrm{k}}$ (i.e., the innovation gain used to update the state by the Kalman filter) matrices are adjusted over time to maintain consistency with the updated plant model, as follows:

$$\begin{aligned} \boldsymbol{L}_{\mathrm{k}} &= (\boldsymbol{A}_{\mathrm{k}}\boldsymbol{P}_{\mathrm{k|k\text{-}1}}\boldsymbol{C}_{\mathrm{k}}^\top + \boldsymbol{N})(\boldsymbol{C}_{\mathrm{k}}\boldsymbol{P}_{\mathrm{k|k\text{-}1}}\boldsymbol{C}_{\mathrm{k}}^\top + \boldsymbol{R})^{-1} \\ \boldsymbol{M}_{\mathrm{k}} &= \boldsymbol{P}_{\mathrm{k|k\text{-}1}}\boldsymbol{C}_{\mathrm{k}}^\top(\boldsymbol{C}_{\mathrm{k}}\boldsymbol{P}_{\mathrm{k|k\text{-}1}}\boldsymbol{C}_{\mathrm{k}}^\top + \boldsymbol{R})^{-1} \end{aligned} \tag{4.13}$$

where $\boldsymbol{Q}$, $\boldsymbol{R}$, and $\boldsymbol{N}$ are constant covariance matrices and $\boldsymbol{P}_{k|k-1}$ is the state estimate error covariance matrix at time k based on information available at time $k-1$ and updated as follows:

$$\boldsymbol{P}_{\mathrm{k+1|k}} = \boldsymbol{A}_{\mathrm{k}}\boldsymbol{P}_{\mathrm{k|k\text{-}1}}\boldsymbol{A}_{\mathrm{k}}^\top - (\boldsymbol{A}_{\mathrm{k}}\boldsymbol{P}_{\mathrm{k|k\text{-}1}}\boldsymbol{C}_{\mathrm{k}}^\top + \boldsymbol{N})L_{\mathrm{k}}^\top + \boldsymbol{Q}. \tag{4.14}$$

Using (4.13)-(4.14), the state estimation vector $\boldsymbol{\Delta\hat{x}}_{\mathrm{m}}(k|k)$ is updated based on the last state estimation $\boldsymbol{\Delta\hat{x}}_{\mathrm{m}}(k|k-1)$. For the sake of simplicity, in the sequel we adopt the following lightened notation: $\boldsymbol{\Delta\hat{x}}_{\mathrm{m}}(k) \triangleq \boldsymbol{\Delta\hat{x}}_{\mathrm{m}}(k|k)$.

Third, based on the updated nominal operating conditions and the estimated state vector, the linearized discrete-time model of the HSA defined in ((4.10))-((4.12)) is rewritten as follows:

$$\begin{cases} \boldsymbol{\Delta\hat{x}}_{\mathrm{m}}(h) = \boldsymbol{A}_{\mathrm{k}}\boldsymbol{\Delta\hat{x}}_{\mathrm{m}}(h-1) + \boldsymbol{B}_{\mathrm{k}}\boldsymbol{\Delta u}(h-1) \\ \boldsymbol{\Delta y}(h) = \boldsymbol{C}_{\mathrm{k}}\boldsymbol{\Delta\hat{x}}_{\mathrm{m}}(h) + \boldsymbol{D}_{\mathrm{k}}\boldsymbol{\Delta u}(h) \end{cases}. \tag{4.15}$$

where $\boldsymbol{\Delta\hat{x}}_{\mathrm{m}}(h-1)$ and $\boldsymbol{\Delta\hat{x}}_{\mathrm{m}}(h)$ are respectively the estimated state deviation and the incremental estimated state deviation from the nominal state $\hat{\boldsymbol{x}}_{\mathrm{m}}(h)$:

$$\boldsymbol{\Delta\hat{x}}_{\mathrm{m}}(h) = \hat{\boldsymbol{x}}_{\mathrm{m}}(h) - \overline{\boldsymbol{x}}_{\mathrm{m}}. \tag{4.16}$$

Using the new plant model defined in (4.15)-(4.16), the AMPC approach computes both the MVs and the OVs over the control and the prediction horizon, collecting them in $\boldsymbol{U}(k) := \big(\boldsymbol{u}(k+1)^\top, \ldots, \boldsymbol{u}(k+H_u)^\top\big)^\top$, $\boldsymbol{Y}(k) := \big(\boldsymbol{y}(k+1)^\top, \ldots, \boldsymbol{y}(k+H_p)^\top\big)^\top$, and $\hat{\boldsymbol{x}}_{\mathrm{m}}(k) := \big(\hat{\boldsymbol{x}}_{\mathrm{m}}(k+1)^\top, \ldots, \hat{\boldsymbol{x}}_{\mathrm{m}}(k+H_p)^\top\big)^\top$ related to the given control horizon $\{k+1, \ldots, k+H_u\}$ and prediction horizon $\{k+1, \ldots, k+H_p\}$, respectively. The aim of the AMPC is to reduce the error between the desired and the measured output trajectories and to limit the control effort $\boldsymbol{U}(k)$ by minimizing the following objective function:

$$J\big(\boldsymbol{U}(k), \boldsymbol{Y}(k)\big) = J_{\boldsymbol{Y}}\big(\boldsymbol{U}(k), \boldsymbol{Y}(k)\big) + J_{\boldsymbol{\Delta U}}\big(\boldsymbol{U}(k), \boldsymbol{Y}(k)\big). \tag{4.17}$$

The terms $J_{\boldsymbol{Y}}\big(\boldsymbol{U}(k), \boldsymbol{Y}(k)\big)$ and $J_{\boldsymbol{\Delta U}}\big(\boldsymbol{U}(k), \boldsymbol{Y}(k)\big)$ refer respectively to the output reference tracking and the move suppression of the MVs, and are defined as

$$J_{\boldsymbol{Y}}\big(\boldsymbol{U}(k), \boldsymbol{Y}(k)\big) = \sum_{h=k+1}^{k+H_p} \left(\sum_{j=1}^{n_p} w_j^{\boldsymbol{Y}} \Big(y_j(h) - y_j^{\mathrm{ref}}(h) \Big)^2 \right)$$

$$J_{\boldsymbol{\Delta U}}\big(\boldsymbol{U}(k), \boldsymbol{Y}(k)\big) = \sum_{h=k+1}^{k+H_u} \left(\sum_{l=1}^{n_u} w_l^{\boldsymbol{\Delta U}} \Big(u_l(h) - u_l(h-1) \Big)^2 \right).$$

In both terms, the parameters $w_j^{\boldsymbol{Y}}$ and $w_l^{\boldsymbol{\Delta U}}$ are suitable weighting factors. Moreover, since the piston position is subject to physical limitations, the following constraints are also defined:

$$\mathrm{y}_1^{\mathrm{lb}} < y_1(h) < \mathrm{y}_1^{\mathrm{ub}}, \quad \forall h = k+1, \ldots, k+H_p \tag{4.18}$$

where $\mathrm{y}_1^{\mathrm{lb}}$ and $\mathrm{y}_1^{\mathrm{ub}}$ denote the lower and upper bounds for the piston position, respectively. Hence, the following optimization problem is solved by the AMPC-based regulator:

$$\min_{\boldsymbol{U}(k),\, \boldsymbol{Y}(k),\, \hat{\boldsymbol{x}}_{\mathrm{m}}(k)} \quad (4.17) \tag{4.19a}$$

$$\text{s.t.} \begin{cases} \text{eqs.}(4.11\text{b}) - (4.11\text{c}), (4.15) - (4.16), & \forall h = k+1, \ldots, k+H_p \\ \text{eqs.}(4.18). \end{cases} \tag{4.19b}$$

Summing up, at each time step k, the AMPC controller obtains all measurements $\boldsymbol{Y}(k)$, computes the required MVs $\boldsymbol{U}(k)$ by solving online the optimization problem (4.19), and applies to the plant only the first sample of the obtained control strategy.

4.4 Case Study

For the sake of showing the effectiveness of the proposed methodology, numerical results of a real HSA are shown. The mechanical parameters, gathered in Table 4.1, are derived from the literature case study presented in [30]. The AMPC-based control system is implemented in MATLAB® and the optimization problem (4.19) is solved via a QP solver. All numerical experiments are conducted on a laptop with a 2.5 GHz Intel Core i5-8250U CPU and 8 GB RAM.

Table 4.1: Hydraulic Actuator parameters.

Parameter name	Symbol	Value	Unit
Cylinder length	L	500	mm
Piston rod mass	M	500	Kg
Oil bulk modulus	E	1.2 e+3	-
Piston cross sectional area	A_1	2.2 e+3	mm^2
Rod cross sectional area	A_2	$\frac{3}{4}A_1$	mm^2
Total volume of cylinder	V_{tot}	$A_1\frac{L}{2}+A_2\frac{L}{2}$	Lit
Oil density	ρ	870	Kg/M^3
Supply (pump) pressure	P_S	300 e+6	Pa
Reference (tank) pressure	P_E	1	Pa
Viscous coefficient	b_m	50	N s/m
Valve discharge coefficient	C_d	0.62	-
Valve cross sectional area	A_0	$\frac{A_1}{10}$	mm^2

Table 4.2: MPC parameters.

Parameter name	Symbol	Value	Unit
Sampling time	T_s	100	ms
OVs weights	$w_{1,2}^{\boldsymbol{Y}}$	7.38	-
MVs weights	$w_{1,2}^{\boldsymbol{\Delta U}}$	0	-
Piston position lower bound	y_1^{lb}	0	mm
Piston position upper bound	y_1^{ub}	L	mm

4.4.1 Experimental Setup

The rest position is considered for the initial condition, as shown in Fig. 4.1, resulting in the initial state $\boldsymbol{x}_m(0) = (250, 0, 0)^\top$, initial input $\boldsymbol{u}(0) = \left(A_0C_d\sqrt{\frac{2}{\rho}P_S}, A_0C_d\sqrt{\frac{2}{\rho}P_S}\right)^\top$, and initial output $\boldsymbol{y}(0) = (250, 0)^\top$ vectors [30]. By following the steps defined in Section 4.3.2, an AMPC-based controller is designed, whose parameters are listed in Table 4.2.

The performance of the AMPC-based control system is tested on two different scenarios (i.e., A and B), with a simulation time of 10 s. In the first scenario, the HSA works as the core component of an hydraulic pressing machine used for metal forming processes [4], where a high force to a metal sheet to be formed has to be applied. Conversely, in the second scenario the HSA acts as the driving part for the motors of an industrial manipulator. In this case, it has to follow a given reference trajectory with high precision [31]. Thus, two typical reference signals are considered for both cases, with a maximum amplitude of 300 mm and 1500 kN respectively for the position and force.

4.4.2 Results and Discussion

In both scenarios the closed-loop control system response is tested for various prediction (i.e., H_p) and control horizon (i.e., H_u), while the sampling time (i.e., T_s) is set to 100 ms.

The position and force tracking errors are depicted respectively in Figs. 4.3 (a1)-(a2) for Scenario A and in Figs. 4.4 (a1)-(a2) for Scenario B. These quantities are evaluated with $H_p = 10$ and H_u=$\{5, 6, 7, 8, 9\}$. In both cases the higher the control horizon, the lower the error. This also results in higher variations of Q_1 and Q_2 to better guarantee the tracking capability on the position and the force of the piston, as shown in Figs. 4.3 (a3)-(a4) and Figs. 4.4 (a3)-(a4) for Scenario A and B, respectively. Further, since the lowest tracking error is obtained for $H_u = 9$, we evaluate it by considering $H_p = \{10, 15, 20\}$. The output tracking error signals are shown in Figs. 4.3 (b1)-(b2) and Figs. 4.4 (b1)-(b2) for Scenario A and B, respectively. In this case, it is possible to notice that the shorter

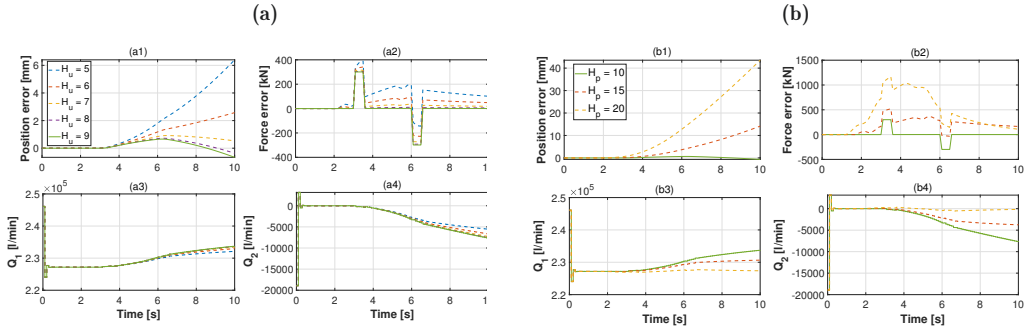

Figure 4.3: Scenario A - Comparison between outputs error (i.e., position and force) and control inputs (i.e., flow rates) for a fixed prediction $H_p = 10$ and different control horizon ($H_u = \{5, 6, 7, 8, 9\}$) (a), and for a fixed control $H_u = 9$ and different prediction horizon ($H_p = \{10, 15, 20\}$) (b).

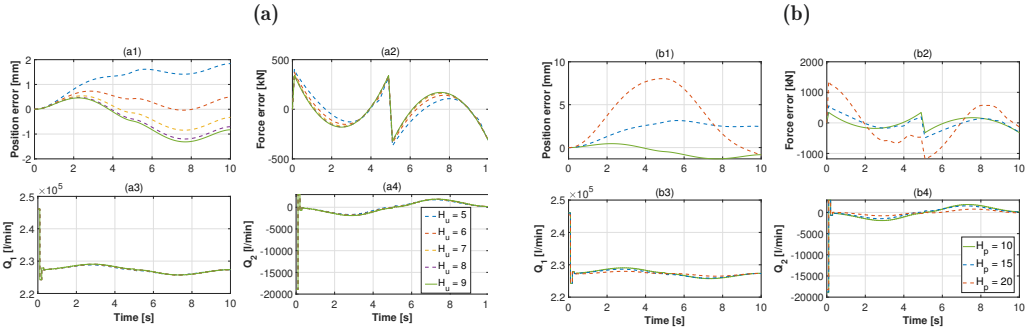

Figure 4.4: Scenario B - Comparison between output errors (i.e., position and force) and control inputs (i.e., flow rates) for a fixed prediction $H_p = 10$ and different control horizon ($H_u = \{5, 6, 7, 8, 9\}$) (a), and for a fixed control $H_u = 9$ and different prediction horizon ($H_p = 10, 15, 20$) (b).

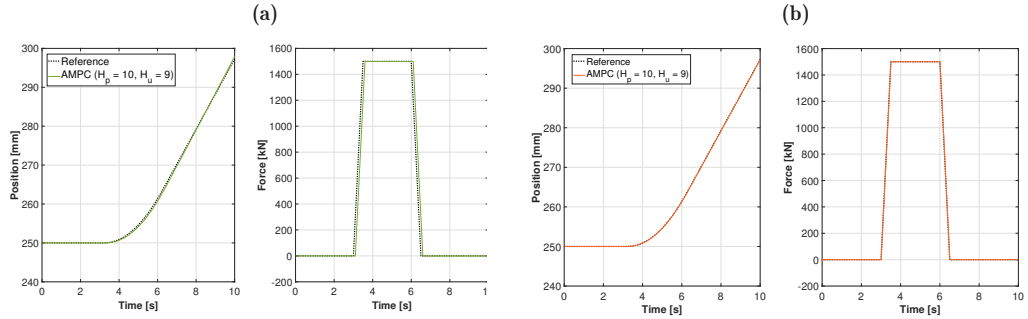

Figure 4.5: Scenario A - Output closed-loop response vs reference signal, obtained for $T_s = 100$ms (a) and $T_s = 10$ms (b) when $H_p = 10, H_u = 9$.

the prediction horizon, the lower the position and force tracking error. Also in this case, the error is lower when higher variation of Q_1 and Q_2 are computed, as arises from Figs. 4.3 (b3)-(b4) and Figs. 4.4 (b3)-(b4) for Scenario A and B, respectively. More in general, it can be inferred that any change in flow rate is directly affected by the value of H_p and H_u. The shorter the prediction horizon interval, the more accurate the prediction made by the controller. Furthermore, it emerges that better performance of the AMPC can be obtained by imposing H_u and H_p of same dimension.

Table 4.3 shows the minimum, maximum, and Mean Square Error (MSE) values for the output tracking obtained on both scenarios. Despite those values (especially MSE) are not negligible when $T_s = 100$ms, it is worthwhile highlighting that peaks in error are due to presence of a short delay between the reference and the actual trajectories, as shown in

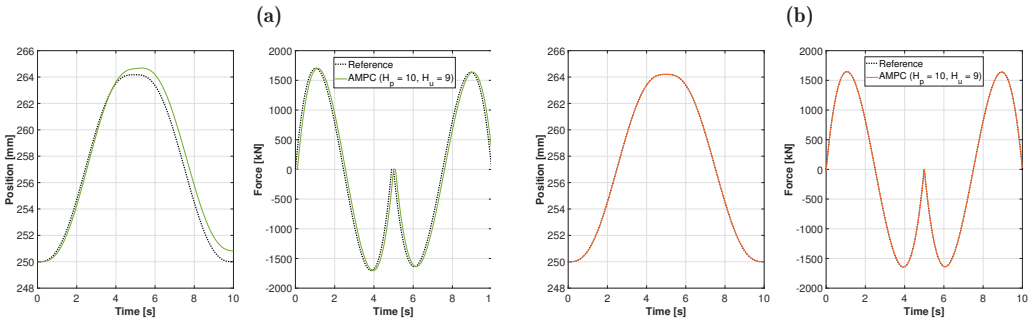

Figure 4.6: Scenario B - Output closed-loop response vs reference signal, obtained for $T_s = 100$ms (a) and $T_s = 10$ms (b) when $H_p = 10, H_u = 9$.

Table 4.3: Output errors.

	$\mathbf{T_s}$ [ms]	Case		min(e)		max(e)		MSE	
		H_p	H_u	**A**	**B**	**A**	**B**	**A**	**B**
Position [mm]	100	10	5	0	0	6.40	1.84	7.14	1.72
		10	6	0	-0.03	2.57	0.73	1.58	0.16
		10	7	0	-0.84	0.91	0.54	0.31	0.23
		10	8	-0.36	-1.19	0.72	0.47	0.12	0.47
		10	9	-0.66	-1.31	0.66	0.45	0.11	0.58
		15	9	0	0	14.15	3.15	33.84	5.42
		20	9	0	-0.88	43.52	8.03	352.64	24.21
	10	10	9	0	-0.06	0.08	0.05	0	0
Force [kN]	100	10	5	-144.38	-367.89	390.27	404.40	19395	23814
		10	6	-227.94	-340.63	341.54	371.86	10566	22889
		10	7	-272.84	-328.10	340.63	353.31	89053	23555
		10	8	-292.33	-322.82	304.57	344.60	8830	24001
		10	9	-298.79	-321.00	300.76	341.49	8894	24150
		15	9	-33.43	-487.00	521.36	562.62	52591	43400
		20	9	-0.16	-1177.6	11697	1369.3	31220	38634
	10	10	9	-30.00	-33.86	30.00	39.97	89.94	233.59

Fig. 4.5 (a) and Fig. 4.6 (a) for Scenario A and B, respectively. Since the delay is equal to one sample T_s, the error decreases as the sampling time is reduced, as can be noticed from the last row of Table 4.3 reporting the results in the case of $T_s = 10$ms. Accordingly, when $T_s = 10$ms the output tracking is more accurate than the case $T_s = 100$ms, as shown in Fig. 4.5 (b) and Fig. 4.6 (b) for scenario A and B, respectively.

4.5 Conclusions

In this chapter, an AMPC architecture is proposed for the position tracking and force control of two-chamber HSAs based on flow control valves. The actuator is represented with a nonlinear MIMO model dynamics. We design an adaptive controller in the sense that is able to change its internal LTV model at each iteration of the control algorithm, exploiting the iterative linearization paradigm. The control system is tested in two different scenarios, and the overall performance is evaluated by means of the minimum, maximum, and mean square error. Simulations also show the effectiveness of the proposed architecture even with a high sampling time, representing a challenging scenario closer to a real manufacturing system.

Extensions regard the integration of the proposed adaptive control approach with a lower-loop control system (within a cascaded closed-loop control architecture) for optimally regulating the valves opening.

References

[1] Alleyne, A. and Hedrick, J. K., "Nonlinear adaptive control of active suspensions," *IEEE transactions on control systems technology*, vol. 3, no. 1, pp. 94–101, 1995.

[2] Liu, Y.-J., Zeng, Q., Liu, L., and Tong, S., "An adaptive neural network controller for active suspension systems with hydraulic actuator," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 12, pp. 5351–5360, 2018.

[3] Yao, B., Bu, F., and Chiu, G.-C., "Nonlinear adaptive robust control of electro-hydraulic servo systems with discontinuous projections," in *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No. 98CH36171)*, IEEE, vol. 2, 1998, pp. 2265–2270.

[4] Bozza, A., Cavone, G., Carli, R., Mazzoccoli, L., and Dotoli, M., "An MPC-based Approach for the Feedback Control of the Cold Sheet Metal Forming Process," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, 2021, pp. 286–291. DOI: 10.1109/CASE49439.2021.9551602.

[5] Hao, Y., Quan, L., Qiao, S., Lianpeng, X., and Wang, X., "Coordinated control and characteristics of an integrated hydraulic–electric hybrid linear drive system," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 2, pp. 1138–1149, 2021.

[6] Deng, W., Zhou, H., Zhou, J., and Yao, J., "Neural network-based adaptive asymptotic prescribed performance tracking control of hydraulic manipulators," *IEEE transactions on systems, man, and cybernetics: systems*, vol. 53, no. 1, pp. 285–295, 2022.

[7] Cavone, G., Bozza, A., Carli, R., and Dotoli, M., "MPC-Based Process Control of Deep Drawing: An Industry 4.0 Case Study in Automotive," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 3, pp. 1586–1598, 2022. DOI: 10.1109/TASE.2022.3177362.

[8] Roemer, D. B., Bech, M. M., Johansen, P., and Pedersen, H. C., "Optimum design of a moving coil actuator for fast-switching valves in digital hydraulic pumps and motors," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 6, pp. 2761–2770, 2015.

[9] Liu, H., Li, Y., Lin, Y., Bao, J., Li, W., and Wang, Y., "The application of the digital controlled hydraulic cylinders group in pendulum wave energy," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 2, pp. 673–682, 2019.

[10] Pedersen, N. H., Johansen, P., and Andersen, T. O., "Feedback control of pulse-density-modulated digital displacement transmission using a continuous approximation," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 5, pp. 2472–2482, 2020.

[11] Shang, Y., Li, R., Wu, S., Liu, X., Wang, Y., and Jiao, Z., "A research of high-precision pressure regulation algorithm based on on/off valves for aircraft braking system," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 8, pp. 7797–7806, 2021.

[12] Tian, H., Li, P. Y., and Van de Ven, J. D., "Valve timing control for a digital displacement hydraulic motor using an angle-domain repetitive controller," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 3, pp. 1306–1315, 2019.

[13] Zhang, Q., Kong, X., Yu, B., Ba, K., Jin, Z., and Kang, Y., "Review and development trend of digital hydraulic technology," *Applied Sciences*, vol. 10, no. 2, p. 579, 2020.

[14] Davison, E. and Ferguson, I., "The design of controllers for the multivariable robust servomechanism problem using parameter optimization methods," *IEEE Transactions on Automatic Control*, vol. 26, no. 1, pp. 93–110, 1981.

[15] Kogler, H., "High dynamic digital control for a hydraulic cylinder drive," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 236, no. 2, pp. 382–394, 2022. DOI: 10.1177/09596518211028089. eprint: https://doi.org/10.1177/09596518211028089. [Online]. Available: https://doi.org/10.1177/09596518211028089.

[16] Lee, J. M., Kim, H. M., Park, S. H., and Kim, J. S., "A position control of electro-hydraulic actuator systems using the adaptive control scheme," in *2009 7th Asian Control Conference*, IEEE, 2009, pp. 21–26.

[17] Yao, B., Bu, F., Reedy, J., and Chiu, G.-C., "Adaptive robust motion control of single-rod hydraulic actuators: Theory and experiments," *IEEE/ASME transactions on mechatronics*, vol. 5, no. 1, pp. 79–91, 2000.

[18] Yao, B., Bu, F., Reedy, J., and Chiu, G.-C., "Adaptive robust motion control of single-rod hydraulic actuators: Theory and experiments," *IEEE/ASME Transactions on Mechatronics*, vol. 5, no. 1, pp. 79–91, 2000. DOI: 10.1109/3516.828592.

[19] Del Re, L. and Isidori, A., "Performance enhancement of nonlinear drives by feedback linearization of linear-bilinear cascade models," *IEEE Transactions on Control Systems Technology*, vol. 3, no. 3, pp. 299–308, 1995.

[20] Ishak, N., Tajjudin, M., Ismail, H., Rahiman, M. H. F., Sam, Y. M., and Adnan, R., "Pid studies on position tracking control of an electro-hydraulic actuator," *International Journal of Control Science and Engineering*, vol. 2, no. 5, pp. 120–126, 2012.

[21] Guan, C. and Pan, S., "Nonlinear adaptive robust control of single-rod electro-hydraulic actuator with unknown nonlinear parameters," *IEEE transactions on control systems technology*, vol. 16, no. 3, pp. 434–445, 2008.

[22] Kaddissi, C., Kenne, J.-P., and Saad, M., "Indirect adaptive control of an electrohydraulic servo system based on nonlinear backstepping," *IEEE/ASME transactions on mechatronics*, vol. 16, no. 6, pp. 1171–1177, 2010.

[23] Hodgson, S., Le, M. Q., Tavakoli, M., and Pham, M. T., "Improved tracking and switching performance of an electro-pneumatic positioning system," *Mechatronics*, vol. 22, no. 1, pp. 1–12, 2012. DOI: https://doi.org/10.1016/j.mechatronics.2011.10.007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957415811001966.

[24] Hodgson, S., Tavakoli, M., Pham, M. T., and Leleve, A., "Nonlinear discontinuous dynamics averaging and pwm-based sliding control of solenoid-valve pneumatic actuators," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 2, pp. 876–888, 2015. DOI: 10.1109/TMECH.2014.2326601.

[25] Dutta, L. and Das, D. K., "A new adaptive explicit nonlinear model predictive control design for a nonlinear mimo system: An application to twin rotor mimo system," *International Journal of Control Automation and Systems*, 2021.

[26] Bujarbaruah, M., Zhang, X., Rosolia, U., and Borrelli, F., "Adaptive mpc for iterative tasks," in *2018 IEEE Conference on Decision and Control (CDC)*, 2018, pp. 6322–6327. DOI: 10.1109/CDC.2018.8618694.

[27] Parvaresh, A. and Mardani, M., "Model predictive control of a hydraulic actuator in torque applying system of a mechanically closed-loop test rig for the helicopter gearbox," *Aviation*, vol. 23, no. 4, pp. 143–153, 2019.

[28] Haas, R. and Lukachev, E., "Optimal feed-forward control of a digital hydraulic drive," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 231, no. 2, pp. 94–106, 2017.

[29] Franco, A. and Santos, V., "Short-term path planning with multiple moving obstacle avoidance based on adaptive mpc," in *2019 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 2019, pp. 1–7. DOI: 10.1109/ICARSC.2019.8733653.

[30] Huang, J., An, H., Yang, Y., Wu, C., Wei, Q., and Ma, H., "Model predictive trajectory tracking control of electro-hydraulic actuator in legged robot with multi-scale online estimator," *IEEE Access*, vol. 8, pp. 95 918–95 933, 2020. DOI: 10.1109/ACCESS.2020.2995701.

[31] Proia, S., Carli, R., Cavone, G., and Dotoli, M., "Control techniques for safe, ergonomic, and efficient human-robot collaboration in the digital industry: A survey," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 3, pp. 1798–1819, 2022. DOI: 10.1109/TASE.2021.3131011.

Chapter 5

Model Reference Adaptive Control for Multi-Chamber Hydraulic Servo Actuators with Pressure Control Valves

V

Abstract

Digital *multi-chamber Hydraulic Servo Actuators* (HSAs) guarantee good energy efficiency while delivering high force with a compact size-to-power ratio, even with multiple loads. However, the complexity introduced by the exponential increase in force combinations with the number of cylinder chambers, coupled with the hybrid behaviour of digital HSAs, poses significant challenges in modeling and control such systems. This chapter introduces a twofold contribution to this context. Firstly, we propose a novel multiple-input single-output *Average Equivalent Discrete-Time Model* based on a Pulse Width Modulation (PWM) strategy inspired to power electronic converters. This allows defining a control-oriented model, by establishing a unique relationship between the duty-cycle of N different PWM signals associated with the digital valves of the HSA and the force delivered by the piston rod. Secondly, we present a two-level adaptive control architecture for the force tracking and pressure control of HSAs that use pressure control valves. It combines a decoupled proportional-integral controller with a decentralized *Model Reference Adaptive Control* providing a streamlined and practical alternative to existing literature-based methods. The approach is applied to a four-chamber digital HSA and compared with a Model Predictive Control -based cascaded control-loop. Albeit our method exhibits a comparable force peak error, it significantly improve the energy efficiency by streamlining the control architecture and reducing the computational time. In addition, the adaptive strategy effectively manages mechanical coupling and treats hydraulic parameters as uncertainties, demonstrating its practical effectiveness.

Contents

5.1 Introduction

In the previous chapter, traditional Hydraulic Servo Actuators (HSAs), namely, hydraulic actuators based on servo valves are presented. Specifically, the chapter aimed at presenting a satisfactory adaptive Model Predictive Control (MPC) strategy for controlling HSAs that use flow control valves. The control system proposed there guarantees a good tracking on rod position and force by provisioning the flow-rate entering in (or exiting from) each piston chamber. Notwithstanding, the contribution lacks methods for controlling the valves opening.

It has been discussed that two types of servo valves exist: *analogical* (also known as proportional) and *digital*. The former are based on electrical torque motors for ensuring accurate pressure feedback control, but make challenging the control due to their complex structure, sensitivity, and high costs. Instead, the latter are characterized by a switched-based operation (i.e., *ON or OFF*) and represent as a promising alternative in various industrial applications [1]–[3], thanks to their benefits in energy saving, fault tolerance, cost-effectiveness, and reliability of industrial systems based on HSAs [4], [5].

One of the first architectures implemented for digital HSAs is based on the *Digital Flow Control Unit* (DFCU). The idea behind is to use several parallel connected ON-OFF valves –which can have either the same or different sizes– to simulate a variable orifice, resulting in a cumulative opening area variation of the flow-rate path. However, the higher control resolution in DFCUs heavily depends on the number of valves, which, in turn, has to be kept as low as possible for not increasing costs and installation spaces [6]. Conversely, one of the most recent advancement and viable solution consists in architectures based on the usage of independent ON-OFF valves –in place of a unique DFCU. In this case, the behaviour of the proportional valve is reproduced by the fast switching frequency of each single digital valve. Moreover, fewer valves are needed, resulting in a low-cost realization of the whole digital HSA [7]. Additionally, one of the latest frontiers in digital HSAs lies in multi-chamber actuators, i.e., cylinders with more than two chambers. These additional chambers serve for regulating the effective cylinder area, thereby modifying the overall force resolution provided by the rod [8]. Specifically, the force resolution is influenced by the number of chambers (denoted as N), as well the number of the available pressure lines (denoted as n), resulting in n^N different force modes. Moreover, utilizing *multi-chamber HSAs* in digital pressure-controlled valve system can enhance the overall energy efficiency by adapting the required pressures and flow-rates, resulting in minimizing throttling losses, especially in case of multiple loads [9]. However, while increasing combinations leads to improved performances, it may also introduce more components, adding complexity to the HSAs control [10].

Therefore, controlling multi-chamber HSAs based on fast-switching pressure-controlled servo valves is not a simple task. In this chapter, we present a twofold result in this sense. On the one hand, we provide a novel Multiple-Input Single-Output (MISO) *Average Equivalent Discrete-time Model* (AEDM) defined upon a *Pulse Width Modulation* (PWM) strategy, inspired to power electronic converters. Specifically, the N different pressure signals (one per each HSA's chamber) act as the multiple inputs, while the force provided by the piston rod serves as the single output in the model. On the other hand, we suggest a two-level cascaded closed-loop control architecture based on an outer Decoupled Proportional-Integral (DPI) controller for the force tracking, and on an inner nonlinear discrete-time Model Reference Adaptive Control (MRAC) (see Appendix A.2) system for the pressure control in each chamber.

5.2 Related Literature and Chapter Positioning

In the related literature, model-based solutions represent the most common way to control HSAs. However, the hybrid nature of digital HSAs, due to the continuous-time hydraulic dynamics and the discrete-time behavior of ON-OFF valves, poses challenges in the modeling process. This often leads to the implementation of partial models, usually neglecting the interaction between the continuous-time and the discrete-time dynamics. For instance, in [11] dynamic equations connecting chamber pressure changes and valve command signals are not integrated, complicating the controller design and introducing stability issues. Subsequent works, such as [12], introduce novel modeling approaches but still separate chamber pressure and valves dynamics, leading to sub-optimal control policies. In the recent chapter by Wang, *et al.* [13], an interesting model defines an equivalent spool displacement for a DFCU-based HSA, but still it treats servo valves and chamber pressure dynamics independently. Overall, addressing these model limitations is crucial for advancing effective control strategies for digital HSAs.

5.2.1 Modulation Techniques of Digital Servo Valves

From an architectural perspective, the control of digital HSAs requires the design of a high-level control law actuated by servo valves. To this aim, the voltage signal commanding the valves opening is properly modulated to apply the high-level control policy at the low-level. Here, a concise overview of the modulation techniques for digital HSAs is provided to present the related limitations.

Upon the above-described frameworks (i.e., DFCU vs ON-OFF -based digital HSAs), some authors categorize the modulation techniques concerning the flow-rate regulation type, resulting in a distinction among *spatial* and *temporal* digital flow-rate HSAs [14], [15]. The former can be *binary-coded* (i.e., higher resolution with fewer valves but prone to pressure fluctuations), *Fibonacci-coded* (i.e., balancing valves and pressure fluctuations), and *equal-coded* (i.e., high redundancy with minimal pressure fluctuations) [13], and usually employ Pulse Number Modulation (PNM) and Pulse Code Modulation (PCM) to regulate DFCUs [16]. Conversely, the latter offer simpler, but less control-oriented architectures, which generally use Pulse Frequency Modulation (PFM) or PWM -based signals to control each ON-OFF valve [7].

With PWM, which involves modulating the width of the pulse signal used to control the opening of each digital valve by keeping a fixed frequency, the flow-rate is proportionally delivered to the digital valve's opening time. Conversely, PFM-based techniques modulate the pulse frequency by keeping a fixed pulse width, resulting in other effective solutions for controlling single ON-OFF servo valves [17]. On the one hand, PFM promotes the diversification of pulses generation in terms of frequency, resulting in enhancing energy efficiency; on the other hand, it requires high working frequencies, potentially exceeding the most recent switching device limits. Thus, an alternative can be arranging N phase-shifted parallel valves [17]. Moreover, in PFM a shorter duration for pulses allows precise actuator positioning (but requires high switching frequency), instead longer pulses offer higher flow-rates (but less precise positioning due to the low frequency) [18]. To this aim, DFCU-based architectures, where the valve modulation is a result of a combined action between PNM, PCM, and PFM approaches, can provide an alternative. For instance, in [19], a PFM-based control enhances the controllability of an equal-coded DFCU compared to PNM. On the contrary, pure PWM-based techniques have been traditionally employed mostly for pneumatic systems (see, e.g., [20]–[22]). Only since the last decade PWM has been utilized for hydraulic systems, due to its higher efficiency and cost reduction with respect to PFM. The basic characteristics of a PWM hydraulic pressure-regulating circuit built on an accumulator, an oil pipe, and a one-way valve are studied in [23]. The results exhibit a good trade-off between the flow-rate resolution and the balancing of the chambers charging and discharging, under different working pressures. In [24] the authors propose the position control of a HSA driven by two PWM-based ON-OF valves. Moreover, in [25] a successful improvement of the stability is achieved in a hydraulic pressure-boost system, bringing the pressure from 50 up to 116 bar by modulating the valve opening with a PWM-based signal. More recently, a self-correcting PWM-based control strategy is introduced in [26]. Compared to the traditional PWM methods, it minimizes the valves opening and closing times, and energy efficiency, by leveraging the feedback of the electrical current to generate duty-cycles. Further, in [27], a PWM-based control scheme is integrated with PNM to increase the flow-rate resolution However, it results in excessive pressure fluctuations due to higher switching frequencies. In addition, authors in [28] present a PWM control strategy using real-time state observation. Despite pressure delay time, coil shell temperature rise, and energy efficiency are reduced compared to three-voltage control, it involves the current monitoring –usually hard to implement. Lastly, a pressure controller based on a differential PWM scheme is suggested in [7] to regulate the chambers pressure in a digital HSA. Albeit such systems may require high (but constant) working switching frequencies, where the pulse width varies over time (leading to severe pressure fluctuations and wear) [29], PWM represents a viable modulation solution for providing well-defined control systems for digital HSAs, since the pressure is almost proportional to the duty-cycle of the PWM signal [14], [25].

5.2.2 Control Techniques for Digital Multi-Chamber HSAs

Linear-based control approaches represent a straightforward and practical solution for controlling the force and/or position of digital HSAs. For instance, a Proportional-Derivative with a velocity feed-forward displacement feedback control system is proposed in [30] for high-speed ON-OFF valves. This approach guarantees high requirements for the cleanliness of the transmission medium, short service life, low reliability, and high manufacturing costs. Moreover, in [31], [32], Linjama *et al.* suggest a Proportional-Integral-Derivative -based multi-level cascaded architecture for controlling position, velocity, and force using fast-switching valves. However, they simplify the control at a higher level by using a linearized system for the inner-loop force controller. To address errors, they implement the so-called *throttling control*, leading to significant energy losses. In [33], the same authors focus on a force controller aiming at minimizing errors by selecting the control signal based on a pressure estimator and a *search space* look-up table. While achieving better tracking results with slower digital valves (avoiding throttling control), energy efficiency is not addressed. Other linear-based control approaches in the literature, such as optimal control theory in [34] and a receding horizon Linear-Quadratic (LQ) tracking controller in [35], offer good tracking results. However, simulations reveal that these techniques can be more time-consuming then others. Particularly, in the latter work, the authors develop an optimal valve profile tracking control for a digital HSA using an event-by-event model of the valve opening duration. In addition, it is worth noting that all the aforementioned works specifically focus on two-chamber HSAs.

On the contrary, multi-chamber HSAs are traditionally controlled using proportional four-way valves. However, this approach is not suitable for digital hydraulics due to the use of parallel-connected ON-OFF valves needed for emulating the traditional servo valve behavior. In this context, the concept of *secondary control*, introduced by Linjama *et al.* in [36], has emerged as one of the initial approaches to control digital multi-chamber HSAs. It consists of controlling the piston rod displacement, resulting in the control of the overall hydraulic actuator force (or torque). In their study, each chamber is connected to the supply or venting line via a single ON-OFF valve, incrementally enhancing the piston force controllability. A similar secondary control approach for a four-chamber HSA is also proposed in [37], utilizing three pressure lines and three ON-OFF valves per chamber to generate 81 force steps. While secondary control reduces energy losses significantly, it faces challenges in ensuring precise control at lower velocities. An alternative to the secondary control is presented in [38], where the authors propose a *Model Predictive Control* (MPC)-based cascaded architecture for a four-chamber HSA. This approach offers a force and pressure control, combining the benefits of previous works. The three-level control architecture includes a force index controller for the valve combination selection, an MPC for both the tracking and energy efficiency, and a valve flow allocator for the low-level pressure control. Although this architecture improves tracking performance and reduces energy losses compared to secondary control, challenges arise from system switching and the need for throttling control for switch compensation in the force index controller. Additionally, the MPC formulation lacks consideration of nonlinear dynamics, while the flow allocator employs a static threshold-based control.

Moreover, since the 1990s, HSAs research has emphasized *adaptive* and *robust control* to address nonlinearities and uncertainties for analogical hydraulic systems [11], [39]–[41]. Adaptive control is crucial for handling plant changes during operation that conventional tools may not handle [42], making these strategies well-suited for HSAs. Conversely, only few examples are given for digital HSAs in the related literature, to the best of our knowledge. For instance, in [43], an adaptive predictive controller is explored for the tracking control in a digital HSA. The system parameters are identified using a recursive least squares method, and the controller linearizes the system based on a reference model. Further, high-precision pressure control systems based on sliding mode are proposed in [44]–[46] for automotive breaking systems to overcome the modeling error and uncertainties. Instead, in [47], an adaptive mechanism using a Kalman filter estimates the valves pressure variability, exploiting them as a feed-forward action in the

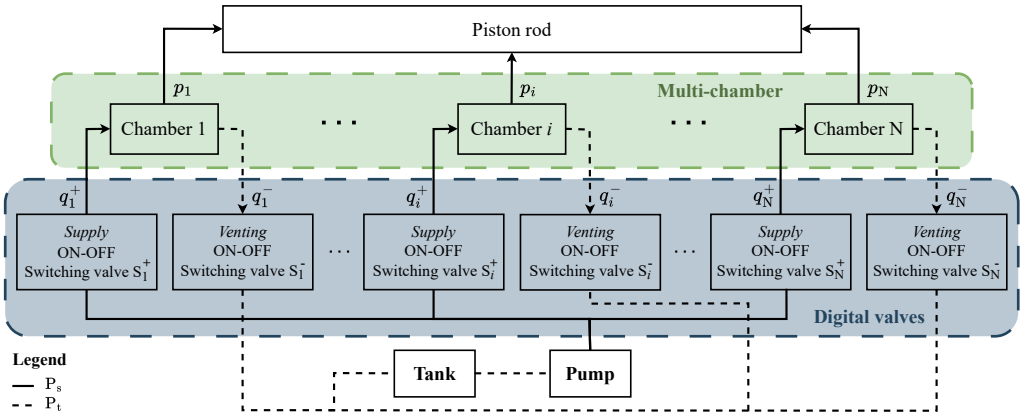

Figure 5.1: Architecture of the digital multi-chamber HSA under study. The cyan and green boxes respectively indicate the digital valves configuration and the HSA's multi-chamber.

LQ control, leading to an improved tracking accuracy than the test bench control scheme. In addition, [48] proposes an adaptive MPC for controlling the rod force and position in a two-chamber HSA. Although achieving good tracking results, the flow-rate based control poses significant implementation challenges.

5.2.3 Chapter Contribution

Differently from the discussed literature, we tackle the twofold problem of defining a proper and efficient modulation technique for the command signal of the switching valves –on the one hand– and, of designing an adaptive control for digital multi-chamber HSAs –on the other hand–, that adapts to the hybrid nature of such systems.

Regarding the first problem, we have already developed an AEDM –inspired to power electronic converters– in [49], proposing a compact and control-oriented solution for a two-chamber HSA. In fact, averaging-based methods represent a typical approach to equivalently describe the continuous-time dynamics of PWM-controlled servo actuators [50]. Nonetheless, despite these modelling techniques are widely used within pneumatic systems thanks to their advantages in rendering the original discontinuous system –eventually non-affine in the input– into an equivalent one –in the control canonical form– [51], [52], very few examples can be found in the related literature concerning HSAs, to the best of our knowledge. As for the control system design, we suggest a two-level cascaded closed-loop control architecture built on a high-level DPI-based and a low-level nonlinear discrete-time MRAC-based controller for respectively tracking the force and regulating the chambers pressure. Despite MRAC is widely used in various applications due to its efficacy in handling plants with time-varying parameters [53], its application to HSAs remains unexplored, to the best of our knowledge. More specifically, the two specific contributions are reported hereafter.

- A novel PWM approach aimed at extending the validity of the AEDM presented in [49] is defined. The aided value of the new modeling approach results in its validity also for four-chamber HSAs, as well as the fact that it introduces the mechanical behaviour of the rod in the dynamics (besides the pure hydraulic one). It results in a unique Multiple-Input Single-Output (MISO) relation between the duty-cycle governing each ON-OFF valve (as the multiple input), and the overall force provided by the piston rod (as the single output), where the velocity and the chamber pressures act as state variables. Moreover, the AEDM suggested in this chapter allows avoiding the hybrid behavior of HSAs, resulting in a discrete-time dynamical representation that simplifies the control formulation and reduces energy losses during valve switching.

- The higher control level uses the DPI-based controller to assess the force tracking, providing the pressure values to the lower level, where a decentralized MRAC-based controller handles them as reference signals. The outer control loop features a linear control law, while the inner adaptive control deals with the HSA nonlinearities and uncertainties in a decentralized manner.

For properly designing the suggested adaptive control architecture, the overall AEDM is split in two parts. The former represents the MISO mechanical system, with the N pressure signals and the rod force as the multiple inputs and the single output, respectively. The latter expresses the Multiple-Input Multiple-Output (MIMO) hydraulic dynamics, where the N different duty-cycle and the relative pressure signals act as the multiple inputs and the multiple outputs, respectively. Finally, the proposed MRAC scheme treats the coupling factors (i.e., rod velocity) as known disturbances, while considers all the hydraulic parameters as uncertainties, namely without requiring their explicit knowledge.

5.3 The Traditional Hybrid Model for Digital Multi-Chamber HSAs

Figure 5.1 shows the complete architecture of the considered digital multi-chamber HSA, comprising a single-rod linear cylinder with N chambers (grouped in a green dashed box) each equipped with a pair of digital valves (grouped in a grey dashed box), an oil tank, and a fixed-displacement pump. The chamber i is characterized by an internal pressure p_i, which serves as mechanical input for the piston rod. The two valves associated to a generic i-th chamber are denoted by S_i^+ and S_i^-, respectively serving for the oil supply and venting. Each valve allows a unique oil direction: either from S_i^+ to the chamber i, or from the latter to S_i^-. The first supplies the flow-rate $q_i^+(t)$ into the chamber, while the latter discharges it with the flow-rate $q_i^-(t)$. The whole HSA is equipped with high $\mathrm{P_s}$ and low $\mathrm{P_t}$ pressure lines, depicted respectively with solid and dashed lines, which allow the oil circulation.

The dynamics of digital multi-chamber HSAs is generally modeled as follows [34], [38], [48]):

$$\begin{cases} \dot{x}(t) = v(t) & \text{(5.1a)} \\ \dot{v}(t) = \dfrac{1}{\mathrm{M}}\Big(-\mathrm{b_m} v(t) - \displaystyle\sum_{i\,=\,1}^{\mathrm{N}} (-1)^i \mathrm{A}_{c_i} p_i(t)\Big) & \text{(5.1b)} \\ \dot{p}_i(t) = f_i\big(\boldsymbol{x}(t)\big) + g_i\big(\boldsymbol{x}(t)\big)\boldsymbol{s}_i(t), \quad \forall i = 1, \ldots, \mathrm{N} & \text{(5.1c)} \\ y(t) = -\mathrm{b_m} v(t) - \displaystyle\sum_{i\,=\,1}^{\mathrm{N}} (-1)^i \mathrm{A}_{c_i} p_i(t) & \text{(5.1d)} \end{cases}$$

where $\boldsymbol{x}(t) = \big(x(t), v(t), p_1(t), \ldots, p_{\mathrm{N}}(t)\big)^\top$ is the state variables vector including N+2 elements, respectively, the rod position, the rod velocity, and the pressure of the N chambers; $y(t)$ is the output variable, i.e., the piston rod force, whereas $\boldsymbol{s}_i(t) = \big(s_i^+(t), s_i^-(t)\big)$ is the i-th input variable vector, i.e., the ON-OFF signal for the valves, defined as follows [54]:

$$s_i^+(t) = \begin{cases} 1 & \text{if } \mathrm{S}_i^+ \text{ is open} \\ 0 & \text{if } \mathrm{S}_i^+ \text{ is closed} \end{cases} \tag{5.2a}$$

$$s_i^-(t) = \begin{cases} 1 & \text{if } \mathrm{S}_i^- \text{ is open} \\ 0 & \text{if } \mathrm{S}_i^- \text{ is closed} \end{cases} \tag{5.2b}$$

$\forall i = 1, \ldots, \mathrm{N}$. Upon the switching-based command signal defined in (5.2), all the possible switching input vectors are gathered in the following set [55]:

$$\mathcal{S}_i = \Big\{ \boldsymbol{s}_{i_m}(t) = \Big(s_i^+(t), s_i^-(t)\Big) : m = 1, \ldots, 4 \Big\}, \tag{5.3}$$

$\forall i = 1, \ldots, \mathrm{N}$, where m denotes the specific operating mode associated with the input vector. The elements of $\mathcal{S}_i$ represent all the possible opening and closing combinations for the corresponding pair of valves i. In model (5.1), eqs. (5.1a)-(5.1b), (5.1d) express the mechanical dynamics, while (5.1c) models the nonlinear input-affine pressure dynamics, where the nonlinear state and input functions are respectively defined as follows [34], [38], [48]:

$$f_i\big(\boldsymbol{x}(t)\big) = \frac{\mathrm{E}}{V_i(t)}\Big((-1)^i \mathrm{A}_{c_i} v(t)\Big), \quad \forall i = 1, \ldots, \mathrm{N} \tag{5.4a}$$

$$g_i\big(\boldsymbol{x}(t)\big) = \frac{\mathrm{E}}{V_i(t)}\Big(q_i^+(t) + q_i^-(t)\Big), \quad \forall i = 1, \ldots, \mathrm{N}. \tag{5.4b}$$

In model (5.1)-(5.4), the piston rod mass M, the i-th chamber cross-sectional area A_{c_i}, the viscous coefficient $\mathrm{b_m}$, and the fluid compressibility bulk modulus E are the time-invariant parameters. Conversely, the i-th chamber volume $V_i(t)$ is the unique time-varying parameter, defined as follows [38], [56]:

$$V_i(t) = \left(\frac{\mathrm{L}}{2} - (-1)^i x(t)\right)\mathrm{A}_{c_i}, \quad \forall i = 1, \ldots, \mathrm{N} \tag{5.5}$$

where L is the overall length of the rod. The supply $q_i^+(t)$ and venting $q_i^-(t)$ flow-rates contributions in (5.4b) can be respectively expressed as functions of $p_i(t)$, by using the Continuity Equation (CE) of flow-rate through an orifice [57]:

$$q_i^+(t) = \Phi_i \Delta p_i^+(t) \tag{5.6a}$$

$$q_i^-(t) = \Phi_i \Delta p_i^-(t) \tag{5.6b}$$

with:

$$\Phi_i = \mathrm{A}_{s_i} \mathrm{C}_{\mathrm{d},i} \sqrt{\frac{2}{\rho}} \tag{5.7a}$$

$$\Delta p_i^+(t) = \sqrt{\left|\mathrm{P_s} - p_i(t)\right|} \tag{5.7b}$$

$$\Delta p_i^-(t) = -\sqrt{\left|p_i(t) - \mathrm{P_t}\right|} \tag{5.7c}$$

$\forall i = 1, \ldots, \mathrm{N}$, where A_{s_i} and $\mathrm{C}_{\mathrm{d},i}$ indicate respectively the cross-sectional area and discharge coefficient of the valve i; ρ is the oil density, while $\Delta p_i^+(t)$ and $\Delta p_i^-(t)$ are the supply and venting pressure drops, respectively.

In summary, the hybrid model traditionally used for digital multi-chamber HSAs is comprehensively expressed by eqs. (5.1), (5.4)-(5.7) for the nonlinear continuous-time hydro-mechanical dynamics, and by eq. (5.2) for the discrete -event dynamics. Moreover, for each of the 2^{N} different force modes, a different dynamical model is defined, due to the specific control input combinations. Consequently, the formulation of the related control problem is non-trivial.

5.4 The Proposed Modelling Approach for Digital Multi-Chamber HSAs

In this section, we first introduce the PWM strategy for properly commanding the digital HSA. Then, we describe the AEDM formulation for providing a solution to reduce the overall number of the HSA force modes from 2^{N} to N.

Exploiting the electro-hydraulic analogy, we establish an equivalence between our digital multi-chamber HSA and the single-phase DC/AC inverter (Fig. 5.2). While Fig. 5.2 (a) depicts the electronic half-bridge circuit traditionally used for modeling single-phase DC/AC inverters, Fig. 5.2 (b) illustrates its equivalent hydraulic version proposed here for describing the behavior of each i-th chamber. As for the inverter, the

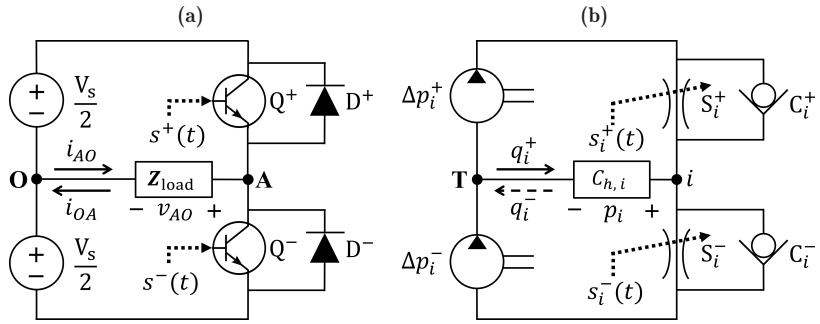

Figure 5.2: Electronic circuit of the half-bridge single-phase DC/AC inverter [58] (a) and equivalent hydraulic half-bridge circuit of the i-th ON-OFF couple of valves [49] (b).

transistors Q^+ and Q^- represent a pair of digital switches, respectively activated by the PWM-based command signals $s^+(t)$ and $s^-(t)$. As well, the diodes D^+ and D^- permit the current flow only in a specific direction, generating a switching supply voltage $v_{AO}(t)$ across the load for the DC/AC conversion (for further details, see Appendix A.3).

A similar behavior can be applied to each i-th HSA chamber and its digital valves couple. The chamber can be treated as a hydraulic load, characterized by the equivalent hydraulic capacitance $C_{h,i}$. The supply S_i^+ and S_i^- and venting valves, properly driven by their respective PWM-based control signals $s_i^+(t)$ and $s_i^-(t)$, act as digital switches. These, thanks to their respected check valves C_i^+ and C_i^-, allow the oil flow only in a unique specific direction. In this way, upon all the possible switched-based command signals defined in (5.3), the corresponding valve can deliver either a positive or a negative flow-rate according to eq. (5.6). As a result, the pressure dynamics of the i-th chamber switches as follows:

$$\dot{p}_{im}(t) = f_i\big(\boldsymbol{x}(t)\big) + g_i\big(\boldsymbol{x}(t)\big)\boldsymbol{s}_{i_m}(t), \quad \forall \boldsymbol{s}_{i_m}(t) \in \mathcal{S}_i,$$

which is valid $\forall i = 1, \ldots, \mathrm{N}$.

5.4.1 PWM Strategy

The PWM strategy proposed here is based on the following definition of *duty-cycle*.

Definition 5.4.1 (*Duty-cycle*)

The duty-cycle associated with the supply S_i^+ *and venting* S_i^- *valve is respectively defined as*

$$d_{i,k}^+ := \frac{t_{ON,i,k}^+}{\mathrm{T}} \quad \text{and} \quad d_{i,k}^- := \frac{t_{ON,i,k}^-}{\mathrm{T}}$$

$\forall i = 1, \ldots, \mathrm{N}$, *where* $\mathrm{T} > 0$ *is the constant sampling period used to define the discrete sampling time* $t_k = k\mathrm{T}$, $\forall k \in \mathbb{N}$. □

The values of $d_{i,k}^+$ and $d_{i,k}^-$ respectively indicate the portion of time during which $s_i^+(t)$ and $s_i^-(t)$ reach the maximum value (i.e., 1) within T, at each sampling time t_k. Therefore, getting inspired to power electronics, we suggest the following PWM strategy.

Proposition 5.4.1 (*PWM strategy*)

For each duty-cycle $d_{i,k}^+$ *associated with the supply valve* S_i^+, *the PWM-based signal for commanding the* i*-th couple of switching valves at time* t *is defined as*

$$s_i^+(t) = \begin{cases} 1 & \text{for } t_k \le t \le t_k + d_{i,k}^+ \mathrm{T} \\ 0 & \text{for } t_k + d_{i,k}^+ \mathrm{T} < t < t_k + \mathrm{T} \end{cases} \tag{5.8a}$$

$$s_i^-(t) = \begin{cases} 0 & \text{for } t_k \le t \le t_k + d_{i,k}^+ \mathrm{T} \\ 1 & \text{for } t_k + d_{i,k}^+ \mathrm{T} < t < t_k + \mathrm{T} \end{cases} \tag{5.8b}$$

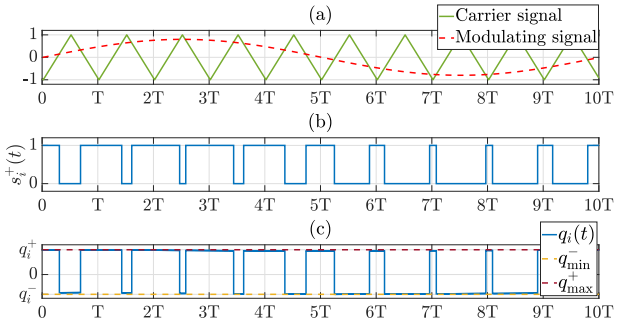

Figure 5.3: PWM approach with $m = 0.8, R = 10$: triangular carrier and sinusoidal modulating signals (a), i-th PWM-based signal $s_i^+(t)$ (b), and resulting flow-rate $q_i(t)$ entering the corresponding HSA chamber (c).

$\forall i = 1, \ldots, \mathrm{N}.$ □

Proof 5.4.1

For each duty-cycle $d^+_{i,k}$ and $d^-_{i,k}$ respectively associated with the supply and venting valves, a general switching-based command signal $\boldsymbol{s}_i(t)$ can be modulated through the PWM, as follows:

$$s_i^+(t) = \begin{cases} 1 & \text{for } t_k \le t \le t_k + d^+_{i,k}\mathrm{T} \\ 0 & \text{for } t_k + d^+_{i,k}\mathrm{T} < t < t_k + \mathrm{T} \end{cases}$$

$$s_i^-(t) = \begin{cases} 1 & \text{for } t_k \le t \le t_k + d^-_{i,k}\mathrm{T} \\ 0 & \text{for } t_k + d^-_{i,k}\mathrm{T} < t < t_k + \mathrm{T} \end{cases}$$

$\forall i = 1, \ldots, \mathrm{N}$, *resulting in the following set of all the possible input vectors, at time t:*

$$\mathcal{S}_i = \left\{ \boldsymbol{s}_{i_m}(t) = \left(s_i^+(t), s_i^-(t) \right) : m = 1, \ldots, 4 \right\}$$

$\forall i = 1, \ldots, \mathrm{N}$, *where m denotes the specific operating mode associated with the input vector. Within each PWM period, the control input $\boldsymbol{s}_{i_m}(t)$ can switch between these m modes. Hence, we can define the duty-cycle associated with the mode m (or modal duty-cycle) $\boldsymbol{d}_{i,k_m}$ as the fraction of time T where a given mode is active, and arrange all these modal duty-cycles in the following matrix [55]:*

$$\mathbf{D}_{i,k} = \left(\boldsymbol{d}^\top_{i,k_1}, \ldots, \boldsymbol{d}^\top_{i,k_4} \right)^\top, \quad \forall i = 1, \ldots, \mathrm{N}$$

where $\boldsymbol{d}_{i,k_m} = \left(d^+_{i,k_m},\ d^-_{i,k_m} \right)$, $\forall m = 1, \ldots, 4$ represents the duty-cycle vector associated with the corresponding mode m, for the i-th chamber.

To make the modulation physically consistent, the duration of each active mode m must be exactly equal to T (i.e., the same of the whole PWM period), to this aim we impose [55]:

$$\left\| \mathbf{D}_{i,k} \right\|_1 = 1, \quad \text{for } t_k \le t \le t_k + \mathrm{T}, \quad \forall m = 1, \ldots, 4.$$

Moreover, without loss of generality, only two admissible modes (i.e., for pressurizing and for venting the chamber i) can be considered for the corresponding switching valves [50]. This results in the following constraint for each modal duty-cycle $\boldsymbol{d}_{i,k_m}$:

$$\boldsymbol{d}_{i,k_m} = \left(d^+_{i,k}\ ,\ 1 - d^+_{i,k} \right) \quad \text{for } m = 1, 2 \tag{5.9}$$

$\forall i = 1, \ldots, \mathrm{N}$.

Constraint (5.9) *guarantees that, within the PWM time interval T, the supply valve is open and closed respectively for a time* $d^+_{i,k}$ *(resulting in the pressurized mode) and a time* $1-d^+_{i,k}$ *(resulting in the venting mode). Accordingly, the venting valve is open and closed respectively for a time* $1-d^+_{i,k}$ *(resulting in the venting mode) and a time* $d^+_{i,k}$ *(resulting in the pressurized mode).* □

The PWM-based strategy defined in Proposition. 5.4.1 ensures that each i-th chamber can be either pressurized or vented, at the same time t. As a consequence, the corresponding pair of valves can deliver either a positive or a negative flow-rate, as described below:

$$\begin{cases} q_i^+(t) = \Phi_i \Delta p_i^+(t), & \text{for } \; t_k \le t \le t_k + d^+\mathrm{T} \\ q_i^-(t) = \Phi_i \Delta p_i^-(t), & \text{for } \; t_k + d^+\mathrm{T} < t < t_k + \mathrm{T} \end{cases}. \tag{5.10}$$

Figure 5.3 illustrates the proposed PWM strategy, where a triangular-wave carrier and a sinusoidal-wave modulating signals are used to generate the switching command $s_i^+(t)$ (note that, $s_i^-(t) = 1 - s_i^+(t)$). The carrier signal has a unit width and frequency $f_s = 1/\mathrm{T}$, while the modulating signal has frequency $f_m \ll f_s$ and amplitude m. The pulses of the generated PWM-based signal are centered in $k\mathrm{T}$. Additionally, we define the PWM frequency ratio as $R = f_s/f_m$.

5.4.2 AEDM Formulation

The flow-rates entering the chamber i in accordance with eq. (5.10) impact the pressure during the time interval $d^+_{i,k}\mathrm{T}$ and $\mathrm{T} - d^+_{i,k}$, as respectively indicated in (5.6)-(5.7). Nonetheless, hydraulic pressure transient times typically exceed the PWM sampling period T, thus we can assume that the chamber pressure $p_i(t)$ remains constant for t_k [56], approximating it with its mean value $\overline{p}_{i,k}$:

$$p_i(t) \approx \overline{p}_{i,k} = \frac{1}{\mathrm{T}} \int_{t_k}^{t_k+\mathrm{T}} p_i(t)\, dt, \quad t_k \le t \le t_k + \mathrm{T} \tag{5.11}$$

$\forall i = 1, \ldots, \mathrm{N}$. This leads to focusing on substantial changes in the mean pressure value at each t_k. As a result, the supply and venting pressure drops can be approximated as:

$$\Delta p_i^+(t) \approx \Delta p_{i,k}^+ = \sqrt{\left|\mathrm{P_s} - p_{i,k}\right|}, \quad t_k \le t \le t_k + d^+_{i,k}\mathrm{T} \tag{5.12a}$$

$$\Delta p_i^-(t) \approx \Delta p_{i,k}^- = -\sqrt{\left|p_{i,k} - \mathrm{P_t}\right|}, \; t_k + d^+_{i,k}\mathrm{T} < t < t_k + \mathrm{T} \tag{5.12b}$$

$\forall i = 1, \ldots, \mathrm{N}$, while the flow-rates become:

$$q_{i,k}^+ \approx \Phi_i \Delta p_{i,k}^+, \quad t_k \le t \le t_k + d^+_{i,k}\mathrm{T} \tag{5.13a}$$

$$q_{i,k}^- \approx \Phi_i \Delta p_{i,k}^-, \quad t_k + d^+_{i,k}\mathrm{T} < t < t_k + \mathrm{T} \tag{5.13b}$$

$\forall i = 1, \ldots, \mathrm{N}$. Similarly to the single-phase DC/AC inverter, the proposed PWM-based formulation allows to approximate also the flow-rate signal with its mean value, within the discrete period t_k. Therefore, rather than relying on (5.13), we can directly consider the mean value of the total flow-rate entering the i-th chamber during each discrete sampling time t_k, expressed as follows:

$$\overline{q}_{i,k} = \frac{1}{\mathrm{T}} \left(\int_{t_k}^{t_k + d^+_{i,k}\mathrm{T}} q_{i,k}^+ \, dt + \int_{t_k + d^+_{i,k}\mathrm{T}}^{t_k+\mathrm{T}} q_{i,k}^- \, dt \right) \tag{5.14}$$

$\forall i = 1, \ldots, \mathrm{N}$. Substituting (5.12) in (5.13), and the latter in (5.14), the flow-rate can be rewritten as a function of duty-cycle $d^+_{i,k}$:

$$\begin{aligned}\overline{q}_{i,k} &= \Phi_i\left(\sqrt{|\mathrm{P_s}-\overline{p}_{i,k}|}d^+_{i,k} - \sqrt{|\overline{p}_{i,k}-\mathrm{P_t}|}\left(1-d^+_{i,k}\right)\right)\\ &= \Phi_i\left(\sqrt{|\mathrm{P_s}-\overline{p}_{i,k}|} + \sqrt{|\overline{p}_{i,k}-\mathrm{P_t}|}\right)d^+_{i,k} - \Phi_i\sqrt{|\overline{p}_{i,k}-\mathrm{P_t}|}\end{aligned} \tag{5.15}$$

$\forall i = 1,\ldots,\mathrm{N}$. Thus, applying the CE to the equivalent hydraulic half-bridge circuit results in the following equivalent Kirchhoff's Current Law (EKCL) at node i for governing its corresponding pressure dynamics [57]:

$$\dot{p}_i(t) = \tfrac{1}{C_{h,i}(t)}\Big(q_i(t) - \dot{V}_i(t)\Big), \quad \forall i = 1,\ldots,\mathrm{N} \tag{5.16}$$

where $V_i(t), C_{h,i}(t)$, and $p_i(t)$ denote respectively the volume, the equivalent hydraulic capacitance, and the pressure of the HSA's chamber i, whereas $q_i(t)$ the provisioned flow-rate. Thus, substituting (5.5) and (5.18) in (5.16), the EKCL becomes:

$$\dot{p}_i(t) = \frac{\mathrm{E}}{V_i(t)}\left((-1)^i \mathrm{A}_{c_i} v(t) + q_i(t)\right), \quad \forall i = 1,\ldots,\mathrm{N}. \tag{5.17}$$

It is easy to see that (5.17) is identical to (5.1c), showing that each chamber i can be effectively considered as an equivalent hydraulic capacitance load.

Remark 5.4.1

Note that, within the electro-hydraulic analogy framework, the time derivative of the i-th chamber volume $\dot{V}_i(t)$ exactly represents a flow-rate, whereas the equivalent hydraulic capacitance can be defined as:

$$C_{h,i}(t) = \tfrac{V_i(t)}{\mathrm{E}}, \quad \forall i = 1,\ldots,\mathrm{N}. \tag{5.18}$$

From the above equation, it appears evident that also $C_{h,i}(t)\dot{p}_i(t)$ represents a flow-rate (i.e., expressed in m^3/s). □

Then, the mechanical and pressure dynamical equations are discretized with the Taylor-Lie approximation method [59], which enables the representation of an nonlinear input-affine dynamics as a polynomial expansion. However, the PWM-based flow-rate could not be discretized using a fixed sampling time (because of its discontinuity within T), resulting in violating the Zero-Order Hold (ZOH) condition. Instead, if we use its mean value (which adhere the ZOH) over the sampling period T, the duty-cycle can be used as the constant input, resulting in the following formulation:

$$\begin{cases}\overline{v}_{k+1} = \overline{v}_k + \dfrac{1}{\mathrm{M}}\Big(-\mathrm{b_m}\overline{v}_k - \displaystyle\sum_{i\,=\,1}^{\mathrm{N}}(-1)^i \mathrm{A}_{c_i}\overline{p}_{i,k}\Big)\mathrm{T} & \text{(5.19a)}\\ \overline{y}_{k+1} = \overline{y}_k - \Big(\mathrm{b_m}\overline{v}_k - \displaystyle\sum_{i=1}^{\mathrm{N}}(-1)^i \mathrm{A}_{c_i}\overline{p}_{i,k}\Big)\mathrm{T} & \text{(5.19b)}\end{cases}$$

$$\overline{p}_{i,k+1} = a_{i,k}\ f_{i,d}(\overline{p}_{i,k}) + b_{i,k}\ g_{i,d}(\overline{p}_{i,k})d^+_{i,k} + \omega_{i,k} \tag{5.19c}$$

with:

$$a_{i,k} = b_{i,k} = \frac{\mathrm{E}\Phi_i}{\overline{V}_{i,k}} \tag{5.20a}$$

$$f_{i,d}(\overline{p}_{i,k}) = -\sqrt{|\overline{p}_{i,k}-\mathrm{P_t}|} \tag{5.20b}$$

$$g_{i,d}(\overline{p}_{i,k}) = \sqrt{|\mathrm{P_s}-\overline{p}_{i,k}|} + \sqrt{|\overline{p}_{i,k}-\mathrm{P_t}|} \tag{5.20c}$$

$$\omega_{i,k} = (-1)^i \mathrm{A}_{c_i}\overline{v}_k \tag{5.20d}$$

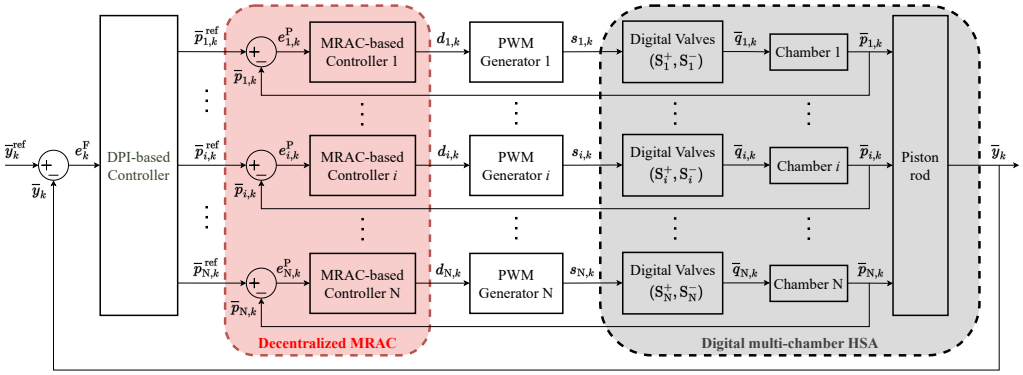

Figure 5.4: Block diagram of the proposed two-level adaptive control architecture. The plant, representing the whole configuration of the digital multi-chamber HSA, is highlighted in the gray box, while the red box underlines the decentralized MRAC-based controller implemented at the low-level.

where eqs. (5.19c) and (5.20b)-(5.20d) are valid $\forall i = 1, \ldots, \mathrm{N}$. Eqs. (5.19a)-(5.19b) model the pure mechanical MISO behaviour of the HSA, having the mean value of pressure $\overline{p}_{i,k}$, $\forall i = 1, \ldots, \mathrm{N}$, as multiple input and the mean value of piston rod force $\overline{y}_{k+1}$ as single output. Conversely, eq. (5.19c) represents the i-th SISO model for the pure hydraulic behaviour of the corresponding chamber, having the duty-cycle associated with the supply valve $d^{+}_{i,k}$ as single input and the mean value of pressure $\overline{p}_{i,k}$ as single output, $\forall i = 1, \ldots, \mathrm{N}$.

The terms $a_{i,k}$ and $b_{i,k}$ represent unknown time-varying parameters, $f_{i,d}(\cdot)$ and $g_{i,d}(\cdot)$ respectively stand for the nonlinear state and input functions, while Φ_i is a constant parameter defined in (5.7a). The discrete variable representing the mean value of rod velocity (i.e., $\overline{v}_k$) acts as a state variable for the mechanical model, whereas is included in $\omega_{i,k}$ as a known disturbance for the hydraulic dynamics. Parameters such as mass piston (M), total rod length (L), chamber area (A_{c_i}), valve cross-sectional area (A_{s_i}), valve discharge coefficient ($\mathrm{C}_{\mathrm{d},i}$), viscous coefficient ($\mathrm{b}_{\mathrm{m}}$), oil density ($\rho$), fluid compressibility (E), supply pressure P_{s}, and venting pressure P_{t} are constant. On the contrary, $a_{i,k}$ and $b_{i,k}$ (which depend on the i-th chamber volume $\overline{V}_{i,k}$) and $\omega_{i,k}$ (which depends on the rod velocity) change over time.

Remark 5.4.2

Besides pressure $p_i(t)$, also the i-th chamber volume $V_i(t)$ and the piston rod velocity $v(t)$ exhibit slow variations within the time period T*, thus:*

$$V_i(t) \approx \overline{V}_{i,k} = \frac{1}{\mathrm{T}} \int_{t_k}^{t_k+\mathrm{T}} V_i(t)\, dt \quad \text{for } t_k \leq t \leq t_k + \mathrm{T}$$

$$v(t) \approx \overline{v}_k = \frac{1}{\mathrm{T}} \int_{t_k}^{t_k+\mathrm{T}} v(t)\, dt \qquad \text{for } t_k \leq t \leq t_k + \mathrm{T}.$$

□

5.5 The Proposed Adaptive Control Architecture

The complete two-level adaptive control framework proposed for the digital multi-chamber HSA modeled before, is shown in Fig. 5.4. The entire architecture of the digital HSA described in Section 5.3 is represented in the the gray box, while the MRAC-based controller is displayed in the red box. At each sampling time t_k, the high-level employs a DPI-based controller to decouple the force error into N different components –one per each chamber– and evaluate the pressure signals as control variables. Then, the low-level uses a decentralized MRAC-based controller to accurately track the pressure signals computed

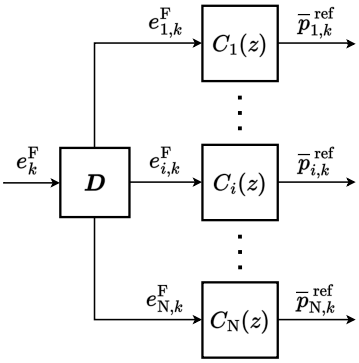

Figure 5.5: Block diagram of the outer DPI-based controller.

by the high-level, by evaluating the appropriate duty-cycle for each valve, ensuring the proper flow-rate to the chambers.

5.5.1 The DPI-based Force Controller

The block diagram of the outer DPI closed-loop system is shown in Fig. 5.5. Firstly, through the decoupling column-vector $\boldsymbol{D} = (D_1, D_2, \ldots, D_{\mathrm{N}})^\top$, the force error is split into N different contributions, one per each chamber i. In the following, we define the decoupling factors of $\boldsymbol{D}$ as geometrically proportional to each chamber contribution:

$$D_i = \begin{cases} \mathrm{A}_{c_i} \Big/ \sum\limits_{\substack{i\,=\,1 \\ i\,=\,\mathrm{odd}}}^{\mathrm{N}} \mathrm{A}_{c_i} \\ \\ -\mathrm{A}_{c_i} \Big/ \sum\limits_{\substack{i\,=\,2 \\ i\,=\,\mathrm{even}}}^{\mathrm{N}} \mathrm{A}_{c_i} \end{cases} \tag{5.21}$$

where $\sum_{i=1,i=\mathrm{odd}}^{\mathrm{N}} \mathrm{A}_{c_i}$ and $\sum_{i=2,i=\mathrm{even}}^{\mathrm{N}} \mathrm{A}_{c_i}$ are respectively the total cross-sectional areas of the whole *odd* and *even* chambers (i.e., *chamber 1* is considered as *odd*, while *chamber 2* as *even*, and so on). Then, N different PI-based controllers are designed –one per each i-th decoupled single-input single-output (SISO) subsystem obtained from the overall MISO mechanical dynamics. We define the N SISO subsystems as the elements of the following row-vector transfer matrix:

$$\boldsymbol{W}(s) = \Big(G_1(s), \ldots, G_i(s), \ldots, G_{\mathrm{N}}(s)\Big)$$

where the i-th element is the transfer function of the related SISO dynamics obtained from (5.1b), (5.1d), defined as follows:

$$G_i(s) = \frac{Y_i(s)}{P_i(s)} = -(-1)^i \frac{\mathrm{A}_{c_i} s}{\frac{\mathrm{b_m}}{\mathrm{M}} + s}, \quad \forall i = 1, \ldots, \mathrm{N}. \tag{5.22}$$

Each transfer function in (5.22) represents the relation between the i-th chamber pressure and the contribution of force provided by the relative HSA's chamber.

Hence, the following PI-based controller $C_i(s)$ is obtained for each plant in (5.22):

$$C_i(s) = G_i(s)^{-1}\Big(\frac{\mathrm{K_{P}}_i s + \mathrm{K_{I}}_i}{s}\Big), \quad \forall i = 1, \ldots, \mathrm{N} \tag{5.23}$$

where $G_i(s)^{-1}$ aims at removing the internal mechanical dynamics of the corresponding chamber i; $\mathrm{K_{p}}_i$ and $\mathrm{K_{I}}_i$ are the proportional and the integral PI constants, respectively. Thus, the closed-loop transfer function becomes as follows:

$$G_{O_i}(s) = \frac{\mathrm{K_{P}}_i s + \mathrm{K_{I}}_i}{(\mathrm{K_{P}}_i + 1)s + \mathrm{K_{I}}_i}, \quad \forall i = 1, \ldots, \mathrm{N}.$$

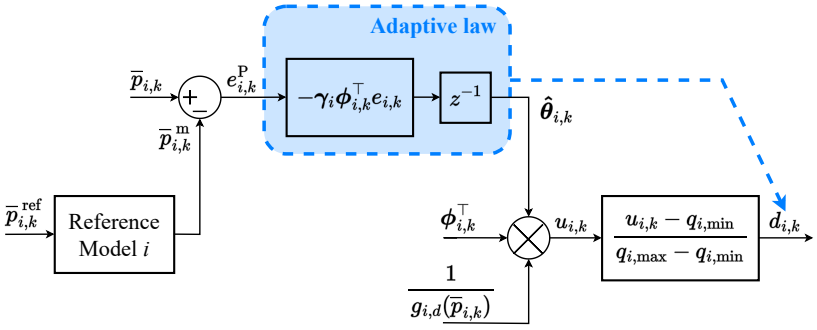

Figure 5.6: Block diagram of the i-th inner decentralized MRAC-based controller. The blue box highlights the adaptation mechanism.

We can get conditions to find the proper values of the parameters $\mathrm{K_{P}}_i$ and $\mathrm{K_{I}}_i$ from the frequency response of the closed-loop transfer function $G_{O_i}(s)$. Specifically, in order to have a correct tracking of the force reference signal, we impose the following constraints:

$$\begin{cases} |G_{O_i}(s)|\Big|_{s=j\omega} = 0.9 \\ arg\big(G_{O_i}(s)\big)\Big|_{s=j\omega} = -\pi/18 \end{cases} \qquad \forall i = 1, \ldots, \mathrm{N}. \tag{5.24}$$

Thus, the following equations are obtained:

$$\mathrm{K_{P}}_i = \frac{9\big(190\sqrt{\tan(\frac{\pi}{18})^2+1} - 1629\tan(\frac{\pi}{18})^2+171\big)}{32761\tan(\frac{\pi}{18})^2 + 361} \tag{5.25a}$$

$$\mathrm{K_{I}}_i = \frac{90\omega\tan(\frac{\pi}{18})\big(181\sqrt{\tan(\frac{\pi}{18})^2+1} + 180\big)}{32761\tan(\frac{\pi}{18})^2 + 361} \tag{5.25b}$$

$\forall i = 1, \ldots, \mathrm{N}$, where ω is a design parameter representing the main frequency of the force reference signal y^{ref}_k.

Finally, (5.23) is discretized with the Forward Euler method (i.e., substituting $s = \frac{z-1}{\mathrm{T}}$), resulting in the following discrete-time PI-based controller:

$$C_i(z) = \mathrm{K_{C}}_i \frac{z^2 + \big(\frac{\mathrm{K_{I}}_i \mathrm{T}(\mathrm{b_m T}-\mathrm{M})}{\mathrm{M}}\big)z + \frac{(\mathrm{b_m T}-1)(\mathrm{K_{I}}_i \mathrm{T}-\mathrm{K_{P}}_i)}{\mathrm{K_{P}}_i} - 1}{z^2 - 2z + 1} \tag{5.26}$$

$\forall i = 1, \ldots, \mathrm{N}$, where $\mathrm{K_{C}}_i = \frac{\mathrm{K_{P}}_i}{-(-1)^i \mathrm{A_{c}}_i}$ represents the gain of the controller i, while the proportional $\mathrm{K_{P}}_i$ and the integral $\mathrm{K_{I}}_i$ parameters are the solutions of the system (5.25a)-(5.25b), respectively.

5.5.2 The Decentralized MRAC-Based Pressure Controller

MRAC uses a *reference model* to generate the desired plant output trajectory. The tracking error, reflecting deviation from this trajectory, is minimized over time by adjusting the feedback control law via an online controller parameter estimation [60]. *Direct* adaptive schemes make direct adjustments to controller parameters, while *indirect* schemes estimate and modify model parameters before adjusting the controller [53]. The proposed decentralized architecture for the inner control-loop is shown in Fig. 5.6, where N different direct MRAC-based controllers –one per each chamber– are designed for ensuring the accurate pressure tracking for the whole MIMO systems. The proposed architecture simplifies the multi-variable control problem by treating the rod velocity as a known disturbance and the hydraulic parameters as unknown variables.

For the sake of describing the controller designing, the following assumptions are given.

Assumption 5.5.1 (*A1*)

(cf. [60], Chapter 6) The nonlinear discrete-time state function $f_{i,d}(\overline{p}_{i,k})$ is bounded for bounded $\overline{p}_{i,k}$, and the input function $g_{i,d}(\overline{p}_{i,k}) > c > 0 \ \forall \ \overline{p}_{i,k} \in \mathbb{R}$, for some constant $c \in \mathbb{R}_*^+$, $\forall k \in \mathbb{N}$ and $\forall i = 1, \ldots, \mathrm{N}$. □

From (5.20b)-(5.20c), we can verify that $-\sqrt{|\mathrm{P_s} - \mathrm{P_t}|} \leq f_{i,d}(\overline{p}_{i,k}) \leq 0$ for $\mathrm{P_t} \leq \overline{p}_{i,k} \leq \mathrm{P_s}$, and $g_{i,d}(\overline{p}_{i,k})$ is always strictly positive for $c = \sqrt{|\mathrm{P_s} - \mathrm{P_t}|} > 0$, $\forall k \in \mathbb{N}$ and $\forall i = 1, \ldots, \mathrm{N}$.

Assumption 5.5.2 (*A2*)

(cf. [61]) The unknown model parameters $a_{i,k}$ and $b_{i,k}$ (with $b_{i,k} > 0$) are slowly time-varying, i.e., $\| a_{i,k+1} - a_{i,k} \| \leq \Delta a_{i_{\max}}$ and $\| b_{i,k+1} - b_{i,k} \| \leq \Delta b_{i_{\max}}$, where $\Delta a_{i_{\max}} \in \mathbb{R}^+$, and $\Delta b_{i_{\max}} \in \mathbb{R}^+$ are known, $\forall k \in \mathbb{N}$ and $\forall i = 1, \ldots, \mathrm{N}$. □

Assumption 5.5.3 (*A3*)

(cf. [61]) The disturbance $\omega_{i,k}$ is persistently bounded and slowly time-varying, respectively meaning that $\| \omega_{i,k} \| \leq \omega_{i_{\max}}$ and $\| \omega_{i,k+1} - \omega_{i,k} \| \leq \Delta\omega_{i_{\max}}$, where $\omega_{i_{\max}} \in \mathbb{R}^+$ and $\Delta\omega_{i_{\max}} \in \mathbb{R}^+$ are unknown, $\forall k \in \mathbb{N}$ and $\forall i = 1, \ldots, \mathrm{N}$. □

We can see from (5.20a) that $b_{i,k} > 0$. As well, both $b_{i,k}$, and the variations of the unknown parameters $a_{i,k}$ and $b_{i,k}$ are bounded by the known and real positive quantities equal to $\Delta a_{i_{\max}} = \Delta b_{i_{\max}} = \mathrm{A}_{c_i}\mathrm{L}$, $\forall k \in \mathbb{N}$ and $\forall i = 1, \ldots, \mathrm{N}$. Further, the disturbance $\omega_{i,k}$ and its variation over $t_{k+1} - t_k$ are respectively bounded by some unknown (but real and positive) quantities $\omega_{i_{\max}} > 0$ and $\Delta\omega_{i_{\max}} > 0$, depending on the maximum allowable velocity $\overline{v}_k$. Overall, these conditions reflect a real-world scenario.

The objective of each MRAC-based controller i is to allow the closed-loop controlled system (5.19c)-(5.20) to follow the discrete-time reference model dynamics below:

$$\overline{p}_{i,k+1}^{\,\mathrm{m}} = -a_i^{\mathrm{m}} \overline{p}_{i,k}^{\,\mathrm{m}} + b_i^{\mathrm{m}} p_{i,k}^{\mathrm{ref}}, \quad \forall i = 1, \ldots, \mathrm{N} \tag{5.27}$$

where $\overline{p}_{i,k}^{\,\mathrm{m}}$ and $p_{i,k}^{\mathrm{ref}}$ are respectively the reference model pressure (acting as a state variable) and the pressure reference signals for the i-th chamber, at the discrete time t_k; conversely, a_i^{m} and b_i^{m} represent the model reference state and input functions, respectively. The proposed control law is:

$$u_{i,k} = \frac{1}{g_{i,d}(\overline{p}_{i,k})} \boldsymbol{\phi}_{i,k}^{\top} \hat{\boldsymbol{\theta}}_{i,k}, \quad \forall i = 1, \ldots, \mathrm{N} \tag{5.28}$$

where $\boldsymbol{\phi}_{i,k}$ and $\hat{\boldsymbol{\theta}}_{i,k}$ represent respectively the regression and the estimated parameters vectors of the unknown controller gains at time t_k, equal to:

$$\boldsymbol{\phi}_{i,k} = \left(f_{i,d}(\overline{p}_{i,k}),\ \overline{p}_{i,k},\ p_{i,k}^{\mathrm{ref}},\ \omega_{i,k} \right)^{\top} \tag{5.29a}$$

$$\hat{\boldsymbol{\theta}}_{i,k} = \left(\hat{\theta}_{i_1,k},\ \hat{\theta}_{i_2,k},\ \hat{\theta}_{i_3,k},\ \hat{\theta}_{i_4,k} \right)^{\top} \tag{5.29b}$$

$\forall i = 1, \ldots, \mathrm{N}$. Since the unknown time-varying parameters $a_{i,k}$ and $b_{i,k}$ defined in (5.20a) are slowly time-varying (for A5.5.2), we can rename a_i and b_i, respectively. Thus, the vector of the true (unknown) gains of the controller can be defined as follows:

$$\boldsymbol{\theta^*}_i = \left(\theta_{i_1}^*,\ \theta_{i_2}^*,\ \theta_{i_3}^*,\ \theta_{i_4}^* \right)^{\top} \tag{5.30}$$

with:

$$\theta_{i_1}^* = -\frac{a_i}{b_i}, \quad \theta_{i_2}^* = -\frac{a_i^{\mathrm{m}}}{b_i}, \quad \theta_{i_3}^* = \frac{b_i^{\mathrm{m}}}{b_i}, \quad \theta_{i_4}^* = -\frac{1}{b_i}$$

$\forall i = 1, \ldots, \mathrm{N}$. From the equations above, we can obtain the following expressions:

$$a_i = -\theta_{i_1}^* b_i, \quad b_i \theta_{i_2}^* = a_i^{\mathrm{m}}, \quad b_i \theta_{i_3}^* = b_i^{\mathrm{m}}, \quad b_i \theta_{i_4}^* = -1 \tag{5.31}$$

$\forall i = 1, \ldots, \mathrm{N}$. To estimate the controller gains, the Lyapunov-based approach is proposed [60].

First, the pressure dynamics (5.19c) is rewritten in terms of the unknown controller gains vector $\boldsymbol{\theta^*}_i$. To this aim, substituting (5.31) in (5.19c), and then, adding and subtracting the term $b_i \boldsymbol{\phi}_{i,k}^{\top} \boldsymbol{\theta^*}_i$, we obtain:

$$\overline{p}_{i,k+1} = -\, a_i^{\mathrm{m}} \overline{p}_{i,k} + b_i^{\mathrm{m}} p_{i,k}^{\mathrm{ref}} + b_i \Big(\boldsymbol{\phi}_{i,k}^{\top} \boldsymbol{\theta^*}_i + g_{i,d}(\overline{p}_{i,k}) d_{i,k}^{+} \Big)$$

$\forall i = 1, \ldots, \mathrm{N}$.

Second, substituting the control law (5.28) in the above equation, the following closed-loop dynamics is achieved for the pressure:

$$\overline{p}_{i,k+1} = -a_i^{\mathrm{m}} \overline{p}_{i,k} + b_i^{\mathrm{m}} p_{i,k}^{\mathrm{ref}} + b_i \boldsymbol{\phi}_{i,k}^{\top} \tilde{\boldsymbol{\theta}}_{i,k} \tag{5.32}$$

$\forall i = 1, \ldots, \mathrm{N}$, where $\tilde{\boldsymbol{\theta}}_{i,k}$ represents the error vector of the estimated controller parameters at time t_k, defined as follows:

$$\tilde{\boldsymbol{\theta}}_{i,k} = \Big(\tilde{\theta}_{i_1,k},\ \tilde{\theta}_{i_2,k},\ \tilde{\theta}_{i_3,k},\ \tilde{\theta}_{i_4,k} \Big)^{\top} \tag{5.33}$$

with:

$$\begin{array}{ll} \tilde{\theta}_{i_1,k} = \hat{\theta}_{i_1,k} - \theta_{i_1}^*, & \tilde{\theta}_{i_2,k} = \hat{\theta}_{i_2,k} - \theta_{i_2}^*, \\ \tilde{\theta}_{i_3,k} = \hat{\theta}_{i_3,k} - \theta_{i_3}^*, & \tilde{\theta}_{i_4,k} = \hat{\theta}_{i_4,k} - \theta_{i_4}^*. \end{array} \tag{5.34}$$

$\forall i = 1, \ldots, \mathrm{N}$. Subtracting each term of eq. (5.27) to eq. (5.32), we obtain the following pressure tracking error dynamics:

$$e_{i,k+1}^{\mathrm{P}} = -a_i^{\mathrm{m}} e_{i,k}^{\mathrm{P}} + b_i \boldsymbol{\phi}_{i,k}^{\top} \tilde{\boldsymbol{\theta}}_{i,k}, \quad \forall i = 1, \ldots, \mathrm{N} \tag{5.35}$$

where $e_{i,k}^{\mathrm{P}} = \overline{p}_{i,k} - \overline{p}_{i,k}^*$ and represents the tracking error on the pressure at the discrete time t_k.

Third, in order to bring (5.35) to zero, the following discrete-time adaptive law for the estimated control parameters vector is defined:

$$\hat{\boldsymbol{\theta}}_{i,k+1} = \hat{\boldsymbol{\theta}}_{i,k} - \boldsymbol{\gamma}_i \boldsymbol{\phi}_{i,k}^{\top} e_{i,k}^{\mathrm{P}} \mathrm{T}, \quad \forall i = 1, \ldots, \mathrm{N} \tag{5.36}$$

where $\boldsymbol{\gamma}_i$ represents the adaptive gains vector, defined as follows:

$$\boldsymbol{\gamma}_i = \Big(\gamma_{i_1}, \gamma_{i_2}, \gamma_{i_3}, \gamma_{i_4} \Big)^{\top}, \quad \text{with } \gamma_{i_l} > 0$$

$\forall i = 1, \ldots, \mathrm{N}, \forall l = 1, \ldots, 4$, with $l \in \mathbb{N}$.

Finally, by means of the corresponding PWM Generator block, the i-th control law (5.28), updated through (5.36), is normalized within the interval $[0, 1]$ to achieve the duty-cycle $d_{i,k}$ related to the corresponding chamber i, as follows:

$$d_{i,k}^{+} = \frac{u_{i,k} - q_{i,\min}}{q_{i,\max} - q_{i,\min}}, \quad \forall i = 1, \ldots, \mathrm{N} \tag{5.37}$$

where $q_{i,\min}$ and $q_{i,\max}$ are respectively the minimum and the maximum allowable flow-rates for the i-th chamber.

5.6 Case Study

For the sake of demonstrating the effectiveness of the proposed approach, this section conducts numerical experiments on a digital four-chamber HSA with eight ON-OFF valves. The HSA serves as the primary component of a real 30-ton class excavator presented in [38], whose mechanical and hydraulic parameters are summarized in Table 5.1. With the aim of further evaluating the proposed methodology, we present a comparison of the

Table 5.1: Parameters of the digital multi-chamber HSA [38].

Parameter name	Symbol	Value	Unit
Number of chambers	N	4	-
Cylinder length	L	1.4	m
Piston rod mass	M	500	Kg
Chamber 1 cross sectional area	A_{c_1}	$1.54\ 10^{-2}$	m^2
Chamber 2 cross sectional area	A_{c_2}	$0.77\ 10^{-2}$	m^2
Chamber 3 cross sectional area	A_{c_3}	$0.38\ 10^{-2}$	m^2
Chamber 4 cross sectional area	A_{c_4}	$0.19\ 10^{-2}$	m^2
Lumped flow coefficient for $S_1^{\pm}$	Φ_1	$6.29\ 10^{-6}$	$\frac{m^3}{s\sqrt{Pa}}$
Lumped flow coefficient for $S_2^{\pm}$	Φ_2	$3.14\ 10^{-6}$	$\frac{m^3}{s\sqrt{Pa}}$
Lumped flow coefficient for $S_3^{\pm}$	Φ_3	$1.56\ 10^{-6}$	$\frac{m^3}{s\sqrt{Pa}}$
Lumped flow coefficient for $S_4^{\pm}$	Φ_4	$0.78\ 10^{-6}$	$\frac{m^3}{s\sqrt{Pa}}$
Valves switching frequency	f_s	1	KHz
Oil bulk modulus	E	$1.5\ \ 10^9$	Pa
Oil density	ρ	880	Kg/m^3
Viscous coefficient	b_m	0.195	Pa·s
Supply (pump) line pressure	P_s	$350\ 10^5$	Pa
Venting (tank) line pressure	P_t	$20\ 10^5$	Pa

Table 5.2: Setup Parameters used for the AEDM Validation and the Closed-loop Simulation.

	Parameter	Value
AEDM Validation	f_s	6283.2 rad/s
	f_m	6.2832 rad/s
	m_1, m_2, m_3, m_4	0.8, 0.9, 0.5, 0.6
	R	1000
Closed-loop Simulation	ω_0	3.14 rad/s
	ω_f	9.42 rad/s
	ω_s	1.5 rad/s
	$\boldsymbol{D}$	$(0.8, -0.8, 0.2, -0.2)^\top$
	ω	15 rad/s
	$K_{P1}, K_{P2}, K_{P3}, K_{P4}$	2.0438
	$K_{I1}, K_{I2}, K_{I3}, K_{I4}$	62.7710
	$\gamma_1, \gamma_2, \gamma_3, \gamma_4$	$(0.1, 0.1, 0.1, 0.1)^\top$

obtained results with [38], one of the most recent and innovative chapter in this field, which employs a cascaded control architecture for digital multi-chamber HSAs. The performance of the two techniques are evaluated by means of two significant indices, i.e., the maximum peak force error and energy efficiency.

All the simulations are conducted in MATLAB/Simulink and run on a laptop with a 1.70 GHz Intel Core i7-1255U CPU and 16 GB RAM.

5.6.1 Simulation Setup

Two simulations are conducted: an open-loop simulation for assessing the AEDM reliability and a closed-loop simulation for testing the two-level adaptive control system. Table 5.2 provides all setup parameters for both cases.

For the AEDM validation, the considered HSA is modeled through eqs. (5.19)-(5.20). Then, four PWM-based commands are generated with a triangular-wave carrier and modulating sinusoidal-wave signals for each chamber i. The high frequency ratio ($R = 1000$) is determined by carrier (f_s) and modulating signals (f_m), while m_1, m_2,

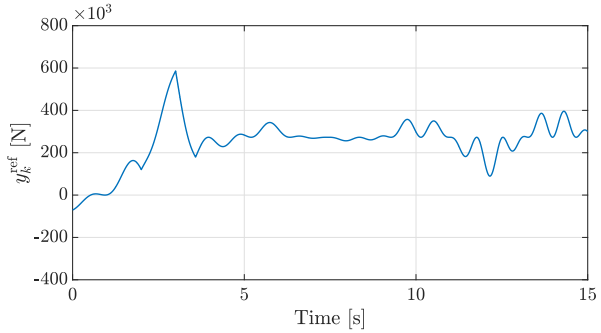

Figure 5.7: Chirp-based force reference signal used for the closed-loop simulation.

Table 5.3: AEDM Validation - Relative Percentage Errors.

			Error [%]		
			min(e)	max(e)	MSE
Variable	Flow-rate	q_1	$2.9702\ 10^{-6}$	0.0800	0.5535
		q_2	$3.4259\ 10^{-6}$	0.0800	0.2735
		q_3	$8.5032\ 10^{-8}$	0.0784	0.1157
		q_4	$2.0124\ 10^{-6}$	0.0814	0.0604
	Pressure	p_1	$3.8866\ 10^{-6}$	0.1223	0.0175
		p_2	$2.6798\ 10^{-5}$	0.1699	0.0293
		p_3	$1.1551\ 10^{-6}$	0.0668	0.0042
		p_4	$2.2807\ 10^{-6}$	0.1084	0.0096
	Velocity	v	$1.2606\ 10^{-9}$	$1.9045\ 10^{-5}$	$1.1258\ 10^{-14}$
	Force	y	$1.9472\ 10^{-6}$	0.0811	0.0294

m_3, and m_4 represent the modulating signal amplitudes. Conversely, in the closed-loop simulation, the PWM-based command signals result from the cascaded control law defined in Section 5.5, implemented with a sampling time T = 0.001 s (i.e., equal to $1/f_s$). Then, four PI-based controllers, characterized by proportional and integral gains $\mathrm{K_{P}}_i$ and $\mathrm{K_{I}}_i$ ($\forall i = 1, 2, 3, 4$), are decoupled through the $\boldsymbol{D}$ vector for the force reference signal depicted in Fig. 5.7. The signal is chirp-based, obtained by modulating a sinusoidal signal having a frequency that linearly increases from ω_0 to ω_f, with a constant frequency ω_s. The DPI designing parameter ω is set to 15 rad/s , resulting from the signals modulation. Finally, an MRAC-based controller with $\boldsymbol{\gamma}_i$ as adaptive gains vector is designed for each i-th chamber.

5.6.2 AEDM Validation

Table 5.3 summarizes the relative minimum (min(e)), maximum (max(e)), and Mean Square Error (MSE) values for the flow-rate, pressure, velocity, and piston rod force across all chambers at a frequency ratio of $R = 1000$. The values, indicative of the proposed model accuracy, consistently exhibit maximum and mean square errors below 1% for all variables.

Furthermore, in Fig. 5.8 the effectiveness of the proposed AEDM is detailed shown over four time intervals around 8 s. We can note that the duty-cycle associated to chamber 1, 2, 3, and 4 remain nearly constant within each sampling period, as shown in in figures (a1), (a2), (a3), and (a4), respectively. Instead, the corresponding PWM-based signals are displayed in figures (b1), (b2), (b3), and (b4) for all the chambers. Finally, from figures (c1), (c2), (c3), and (d4), we can see how the discrete-time flow-rate and pressure signals accurately represent their respective mean values over T, $\forall t_k$.

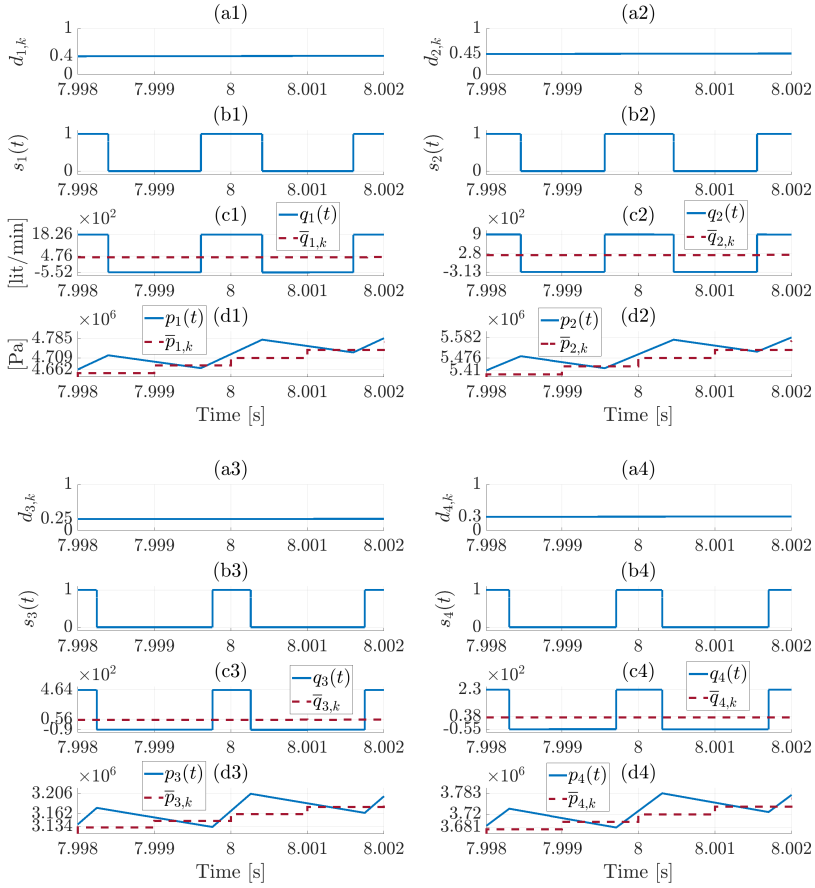

Figure 5.8: AEDM Validation - Detailed behavior around 8 s. Duty-cycle associated to the valves of chamber 1 (a1), chamber 2 (a2), chamber 3 (a3), and chamber 4 (a4), and corresponding PWM-based signals (b1), (b2), (b3), and (b4), respectively; comparison between the (actual) switching flow-rate and pressure against their respective discrete-time mean values for chamber 1 (c1)-(d1), chamber 2 (c2)-(d2), chamber 3 (c3)-(d4), and chamber 4 (c4)-(d4).

5.6.3 Results and Discussion

Figure 5.9 shows the inner MRAC-based control signals. The first and second rows respectively display the control input evaluated with eq. (5.28) and the corresponding duty-cycle computed with eq. (5.37), for each chamber. Specifically, the behavior of the control law and the duty-cycle signals for chamber 1, 2, 3, and 4 is respectively depicted in figures (a1)-(b1), (a2)-(b2), (a3)-(b3), and (a4)-(b4), over the entire simulation time. Moreover, a zoom around 8 s is illustrated in Fig. 5.10. In particular, figures (a1), (a2), (a3), and (a4) respectively show the changes in the control input between two consecutive time steps for chamber 1, 2, 3, and 4, while figures (b1), (b2), (b3), and (b4) show their corresponding obtained duty-cycles. It is worth noting that changes in duty-cycle are minimal in this specific case. Further, the pulse width of the command signals, associated with their respective duty-cycles, for chamber 1, 2, 3, and 4 are depicted in figures (c1), (c2), (c3), and (c4). Conversely, the tracking performances of the inner control-loop are displayed in Fig. 5.11, showing a good pressure tracking with a zero relative percentage error after reaching the settling time of 1 s. Specifically, the pressure tracking in chamber 1, 2, 3, and 4 is respectively depicted in figures (a1), (a2), (a3), and (a4), while their relative percentage error is shown in figures (b1), (b2), (b3), and (b4), respectively.

Fig. 5.12 illustrates the force tracking performance. More in detail, figure (a) compares the overall behavior of the evaluated force signal against the reference, while figure (b) shows that the relative percentage error lies within $[-2, 2]\%$ after the settling time (i.e., 1 s) is reached. Also, the decoupled force errors are detailed in Fig. 5.13, highlighting the

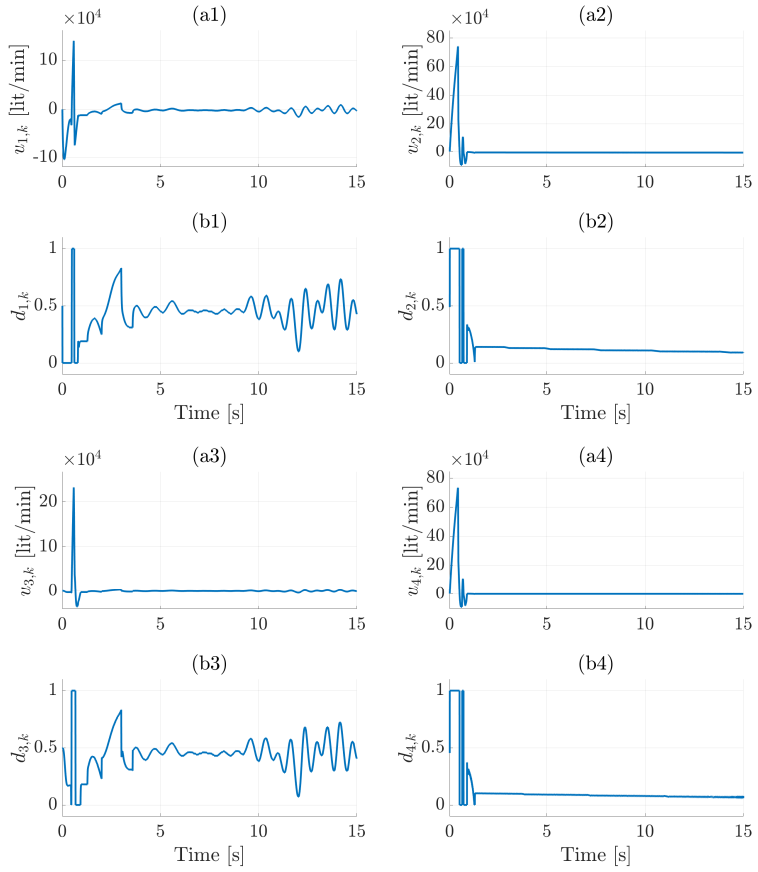

Figure 5.9: Closed-loop simulation - Overall behavior over the whole simulation time (15 s). Control signal provided by each MRAC-based controllers to chamber 1 (a1), chamber 2 (a2), chamber 3 (a3), and chamber 4 (a4) and corresponding duty-cycle respectively generated for each PWM-based couple of valves for chamber 1 (b1), chamber 2 (b2), chamber 3 (b3), and chamber 4 (b4).

Table 5.4: Closed-loop simulation - Relative Percentage Errors.

			Error [%]		
			$\min(e)$	$\max(e)$	MSE
Variable	Pressure	e_1	$1.4092\ 10^{-4}$	5.2069	2.7255
		e_2	1.0317	3.9808	3.3092
		e_3	$1.9823\ 10^{-5}$	7.1595	2.3912
		e_4	0.5002	3.0186	0.8601
	Force	e_1^{F}	$3.7415\ 10^{-5}$	2.2613	0.5404
		e_2^{F}	$7.4829\ 10^{-5}$	4.5226	2.2221
		e_3^{F}	$3.7415\ 10^{-5}$	2.2613	0.5555
		e_4^{F}	$7.4829\ 10^{-5}$	4.5226	2.2221
		e^{F}	$1.9053\ 10^{-5}$	1.4530	0.2124

contributions of each decoupled controller $C_1(z)$, $C_2(z)$, $C_3(z)$, and $C_4(z)$ in figures (a), (b), (c), and (d), respectively. All the tracking error results are summarized in Tab. 5.4. Notably, the maximum force error e^{F} is limited to 1.45% (and to 0.21% for the MSE), while the MSE for all pressure signals remains below 3.3%, showcasing robust performance in this challenging scenario.

Furthermore, the implemented PWM strategy enhances the regulation of the valves switching with respect to the reference approach. By modulating the pulse width, we can adjust the time interval for each valve to be opened or closed through the duty-

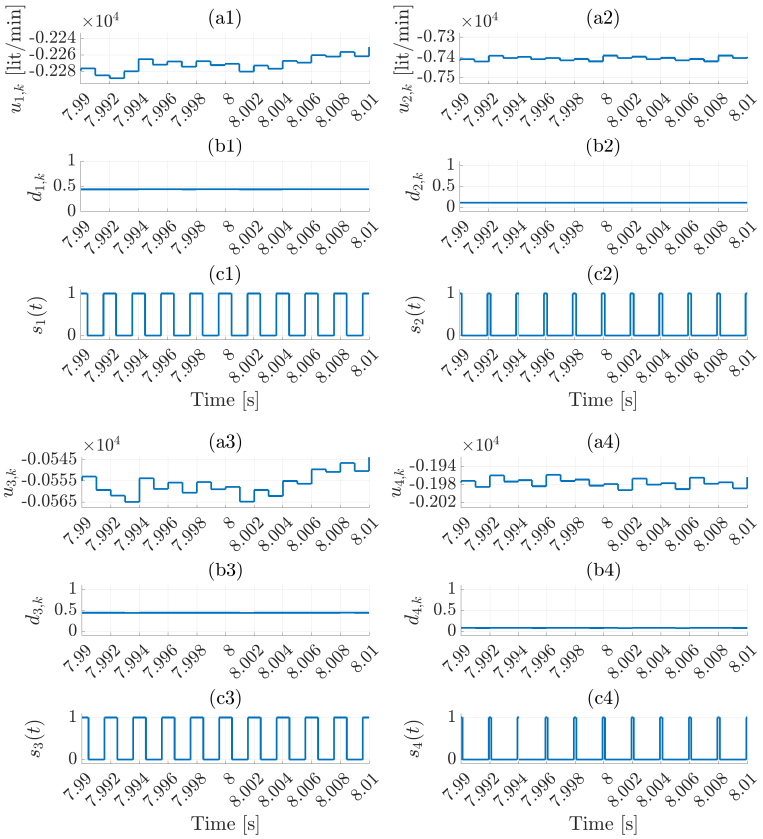

Figure 5.10: Closed-loop simulation - Detailed behaviour around 8 s. Control signal provided by each MRAC-based controllers to chamber 1 (a1), chamber 2 (a2), chamber 3 (a3), and chamber 4 (a4), corresponding duty-cycle generated for each couple of valves for chamber 1 (b1), chamber 2 (b2), chamber 3 (b3), and chamber 4 (b4), and PWM-based command signal respectively obtained for chamber 1 (c1), chamber 2 (c2), chamber 3 (c3), and chamber 4 (c4).

cycle, resulting in improving the overall HSA energy efficiency. Specifically, the average percentage energy efficiency over an entire work cycle can be defined as follows:

$$\eta = \frac{\mathrm{W}_{\mathrm{mech}}}{\mathrm{W}_{\mathrm{mech}} + \mathrm{W}_{\mathrm{loss}}} \cdot 100$$

in which $\mathrm{W}_{\mathrm{mech}}$ and $\mathrm{W}_{\mathrm{loss}}$ are respectively the average values of the mechanical energy and hydraulic energy losses (due to the valves switching), defined as in [38]:

$$\mathrm{W}_{\mathrm{mech}} = \frac{1}{p} \sum_{k=1}^{p} \left| \overline{v}_k \ \overline{y}_k \right|$$

$$\mathrm{W}_{\mathrm{loss}} = \sum_{i=1}^{\mathrm{N}} \left(\frac{1}{p} \sum_{k=1}^{p} \left| (\Delta p_{i,k}^{+} - \Delta p_{i,k}^{-}) \ \overline{q}_{i,k} \right| \right)$$

where p is the overall number of all the collected samples during an entire whole cycle. As a result, the average percentage of energy efficiency achieved during a complete simulation work cycle is 78.60 %, demonstrating a significant improvement with respect to the reference approach ($\simeq$ 71 %).

Finally, for the sake of completeness, the behaviour of the estimated parameters vector $\hat{\boldsymbol{\theta}}_{i,k}$ over the whole simulation time is illustrated in Fig. 5.14, for each chamber. Specifically, in figures (a), (b), (c), and (d) we can appreciate how the estimates of the parameters vector for chamber 1, 2, 3, and 4 respectively change during the adaptation mechanism, until to stabilizing at specific constant values.

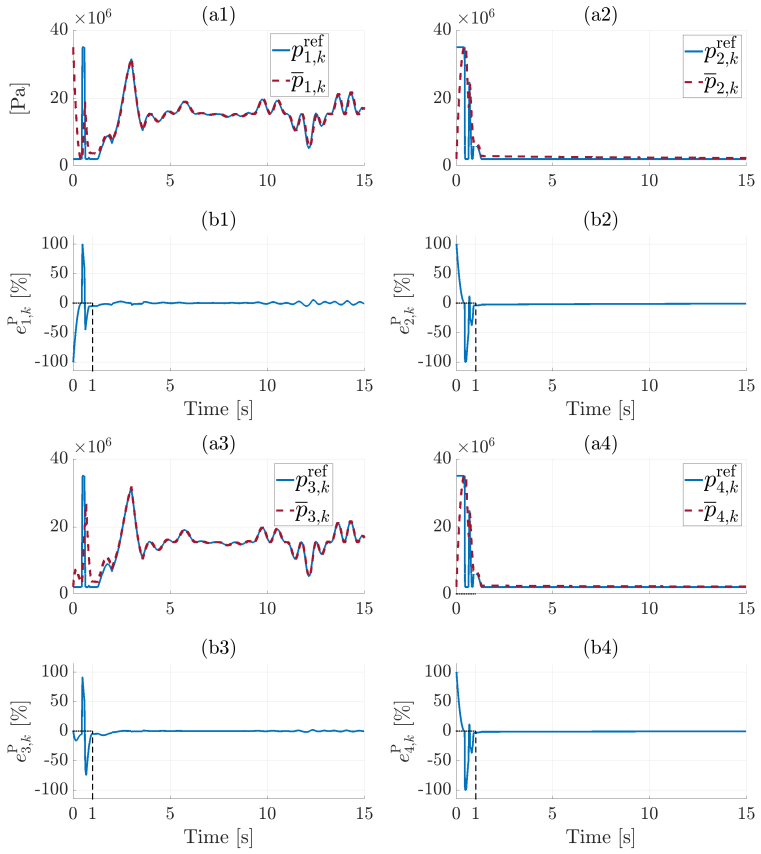

Figure 5.11: Closed-loop simulation - Pressure tracking for chamber 1 (a1), chamber 2 (a2), chamber 3 (a3), and chamber 4 (a4) and corresponding relative percentage error respectively obtained for chamber 1 (b1), chamber 2 (b2), chamber 3 (b3), and chamber 4 (b4) over the whole simulation time.

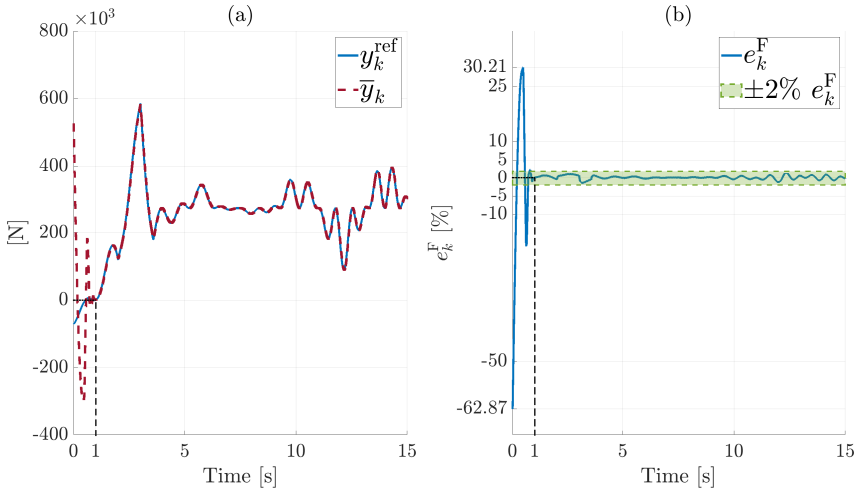

Figure 5.12: Closed-loop simulation - Force tracking (a) and corresponding relative percentage error (b) over the whole simulation time.

In summary, compared to [38], our approach improves energy efficiency by 7.6%, albeit the force peak error goes beyond by 0.45%. Notably, our adaptive-based control architecture significantly streamlines the cascaded control-loop proposed by the benchmark, reducing the computational time required for evaluating the control law. Specifically, in contrast to the reference chapter, that needs to compute an optimization problem (i.e., MPC formulation) for obtaining the control action, our method relies on a faster algebraic

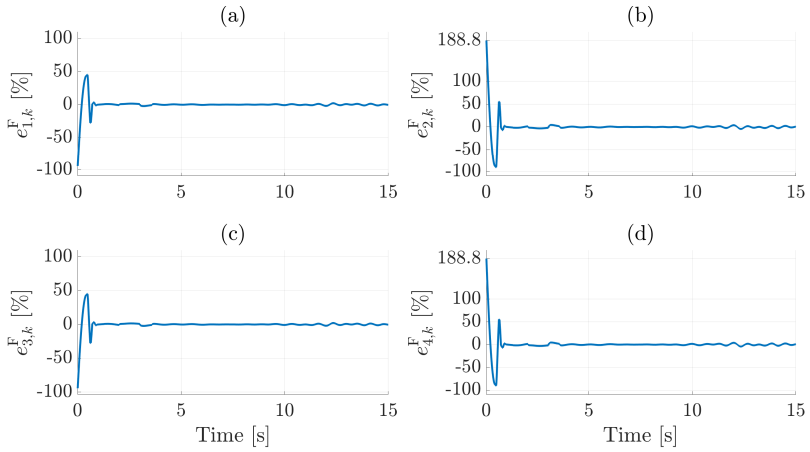

Figure 5.13: Closed-loop simulation - Decoupled force tracking error for chamber 1 (a), chamber 2 (b), chamber 3 (c), and chamber 4 (d) over the whole simulation time.

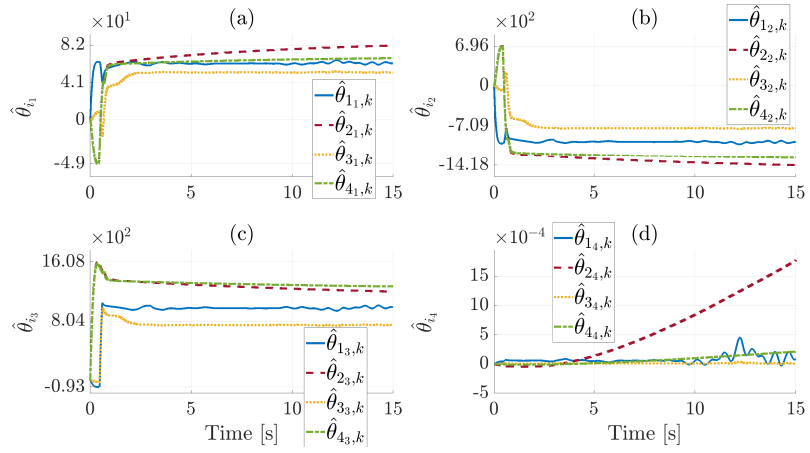

Figure 5.14: Closed-loop simulation - MRAC estimated parameters $\hat{\theta}_{i_1}$ (a), $\hat{\theta}_{i_2}$ (b), $\hat{\theta}_{i_3}$ (c), and $\hat{\theta}_{i_4}$ (d), $\forall i = 1, 2, 3, 4$, over the whole simulation time.

computation.

5.7 Conclusions

In this chapter, we design a MRAC, based on a PWM strategy inspired to power electronics, for digital multi-chamber HSAs that use pressure control valves. Specifically, the contribution is twofold. First, we propose a novel AEDM, resulting in a MISO dynamical model relating the N duty-cycle signals (one per each couple of ON-OFF valves) with the average value of piston rod force. Then, a two-level closed-loop architecture is designed for controlling both the force and chambers pressure. The high-level closed-loop employs a DPI-based controller, whereas the low-level uses a decentralized nonlinear MRAC. The outer loop employs a linear control law, while the inner loop accounts for all the model nonlinearities and uncertainties by treating the rod velocity as a known disturbance and the hydraulic parameters as uncertainties. Numerical simulations on a four-chamber HSA with eight ON-OFF valves show that our adaptive control architecture, coupled with the PWM-based AEDM, allows an effective tracking for both force and pressure. Moreover, compared to a literature-based control approach, our method demonstrates similar tracking performances with enhanced energy efficiency, presenting a promising and more viable alternative.

Future work may extend the current multi-chamber approach to a multi-pressure

architecture. This extension would require adapting the AEDM accordingly. For the inner control loop, the nonlinear discrete-time MRAC formalization could incorporate the time-varying case. Additionally, exploring methods for improving the controller robustness (e.g., leakage, parameter projection, dead zone) represents another interesting path for further research.

References

[1] Liu, H., Li, Y., Lin, Y., Bao, J., Li, W., and Wang, Y., "The application of the digital controlled hydraulic cylinders group in pendulum wave energy," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 2, pp. 673–682, 2019.

[2] Pedersen, N. H., Johansen, P., and Andersen, T. O., "Feedback control of pulse-density-modulated digital displacement transmission using a continuous approximation," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 5, pp. 2472–2482, 2020.

[3] Shang, Y., Li, R., Wu, S., Liu, X., Wang, Y., and Jiao, Z., "A research of high-precision pressure regulation algorithm based on on/off valves for aircraft braking system," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 8, pp. 7797–7806, 2021.

[4] Tian, H., Li, P. Y., and Van de Ven, J. D., "Valve timing control for a digital displacement hydraulic motor using an angle-domain repetitive controller," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 3, pp. 1306–1315, 2019.

[5] Zhang, Q., Kong, X., Yu, B., Ba, K., Jin, Z., and Kang, Y., "Review and development trend of digital hydraulic technology," *Applied Sciences*, vol. 10, no. 2, p. 579, 2020.

[6] Huova, M. and Plöckinger, A., "Improving resolution of digital hydraulic valve system by utilizing fast switching valves," 2010.

[7] Gao, Q., "Nonlinear adaptive control with asymmetric pressure difference compensation of a hydraulic pressure servo system using two high speed on/off valves," *Machines*, vol. 10, no. 1, p. 66, 2022.

[8] Niemi-Pynttäri, O., Linjama, M., Laamanen, A., and Huhtala, K., "Parallel pump-controlled multi-chamber cylinder," in *Fluid Power Systems Technology*, American Society of Mechanical Engineers, vol. 45974, 2014, V001T01A014.

[9] Raduenz, H., Ericson, L., Heybroek, K., De Negri, V. J., and Krus, P., "Improving the efficiency of valve-controlled systems by using multi-chamber actuators," in *Scandinavian International Conference on Fluid Power*, 2021, pp. 224–236.

[10] Donkov, V. H., Andersen, T., Linjama, M., and Ebbesen, M., "Digital hydraulic technology for linear actuation: A state of the art review," *International Journal of Fluid Power*, pp. 263–304, 2020.

[11] Yao, B., Bu, F., Reedy, J., and Chiu, G.-C., "Adaptive robust motion control of single-rod hydraulic actuators: Theory and experiments," *IEEE/ASME transactions on mechatronics*, vol. 5, no. 1, pp. 79–91, 2000.

[12] Stauch, C. and Rudolph, J., "Control-oriented modelling and development of a model-based switching algorithm for a digital hydraulic independent metering cylinder drive," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 231, no. 2, pp. 66–81, 2017.

[13] Wang, P., Cheng, Y., Linjama, M., Yao, J., and Shan, D., "A novel equivalent continuous metering control with a uniform switching strategy for digital valve system," *IEEE/ASME Transactions on Mechatronics*, 2023.

[14] Wang, F., Gu, L., and Chen, Y., "A continuously variable hydraulic pressure converter based on high-speed on–off valves," *Mechatronics*, vol. 21, no. 8, pp. 1298–1308, 2011.

[15] Gao, Q., Wang, J., Zhu, Y., Wang, J., and Wang, J., "Research status and prospects of control strategies for high speed on/off valves," *Processes*, vol. 11, no. 1, p. 160, 2023.

[16] Wang, H., Chen, Z., Huang, J., Quan, L., and Zhao, B., "Development of high-speed on–off valves and their applications," *Chinese Journal of Mechanical Engineering*, vol. 35, no. 1, p. 67, 2022.

[17] Gradl, C. and Scheidl, R., "A basic study on the response dynamics of pulse-frequency controlled digital hydraulic drives," in *Fluid Power Systems Technology*, American Society of Mechanical Engineers, vol. 56086, 2013, V001T01A022.

[18] Schepers, I., Weiler, D., and Weber, J., "Comparison and evaluation of digital control methods for on/off valves," in *Proceedings of the Fifth Workshop on Digital Fluid Power*, 2012, pp. 103–122.

[19] Paloniitty, M., Linjama, M., and Huhtala, K., "Equal coded digital hydraulic valve system–improving tracking control with pulse frequency modulation," *Procedia engineering*, vol. 106, pp. 83–91, 2015.

[20] Ferraresi, C., "A new pcm-pwm combined technique for pneumatic flow-regulating valves," in *Proceedings of the Joint Hungarian–British International Mechatronics Conference*, 1994, pp. 385–390.

[21] Van Varseveld, R. B. and Bone, G. M., "Accurate position control of a pneumatic actuator using on/off solenoid valves," *IEEE/ASME Transactions on mechatronics*, vol. 2, no. 3, pp. 195–204, 1997.

[22] Belforte, G., Mauro, S., and Mattiazzo, G., "A method for increasing the dynamic performance of pneumatic servosystems with digital valves," *Mechatronics*, vol. 14, no. 10, pp. 1105–1120, 2004.

[23] Scheidl, R., Gradl, C., Kogler, H., Foschum, P., and Plöckinger, A., "Investigation of a switch-off time variation problem of a fast switching valve," in *Fluid Power Systems Technology*, American Society of Mechanical Engineers, vol. 45974, 2014, V001T01A033.

[24] Adeli, M. and Kakahaji, H., "Modeling and position sliding mode control of a hydraulic actuators using on/off valve with pwm technique," in *Electrodynamic and Mechatronic Systems*, IEEE, 2011, pp. 59–64.

[25] Wang, F., Gu, L., and Chen, Y., "A hydraulic pressure-boost system based on high-speed on–off valves," *IEEE/ASME transactions on mechatronics*, vol. 18, no. 2, pp. 733–743, 2012.

[26] Zhong, Q., Zhang, B., Yang, H.-Y., Ma, J.-E., and Fung, R.-F., "Performance analysis of a high-speed on/off valve based on an intelligent pulse-width modulation control," *Advances in Mechanical Engineering*, vol. 9, no. 11, p. 1 687 814 017 733 247, 2017.

[27] Paloniitty, M. and Linjama, M., "High-linear digital hydraulic valve control by an equal coded valve system and novel switching schemes," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 232, no. 3, pp. 258–269, 2018.

[28] Gao, Q., Zhu, Y., Luo, Z., and Bruno, N., "Investigation on adaptive pulse width modulation control for high speed on/off valve," *Journal of Mechanical Science and Technology*, vol. 34, pp. 1711–1722, 2020.

[29] Zhang, B., Zhong, Q., Ma, J.-e., *et al.*, "Self-correcting pwm control for dynamic performance preservation in high speed on/off valve," *Mechatronics*, vol. 55, pp. 141–150, 2018.

[30] Liu, Z.-h., Gao, Q.-h., Deng, G., Niu, H.-l., and Li, J.-y., "The position control of hydraulic cylinder based on high-speed on-off valve," *International Journal of Modelling, Identification and Control*, vol. 22, no. 1, pp. 54–67, 2014.

[31] Linjama, M., Huova, M., and Huhtala, K., "Model-based force and position tracking control of an asymmetric cylinder with a digital hydraulic valve," *International Journal of Fluid Power*, vol. 17, no. 3, pp. 163–172, 2016.

[32] Linjama, A. P. M., Huova, I. M., Karhu, M. S. O., and Huhtala, K., "High-performance digital hydraulic tracking control of a mobile boom mockup," 2016.

[33] Linjama, M. and Huova, M., "Model-based force and position tracking control of a multi-pressure hydraulic cylinder," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 232, no. 3, pp. 324–335, 2018.

[34] Kogler, H., "High dynamic digital control for a hydraulic cylinder drive," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 236, no. 2, pp. 382–394, 2022. DOI: 10.1177/09596518211028089. eprint: https://doi.org/10.1177/09596518211028089. [Online]. Available: https://doi.org/10.1177/09596518211028089.

[35] Li, H., Huang, Y., Zhu, G. G., and Lou, Z. D., "Profile tracking for an electro-hydraulic variable valve actuator using receding horizon lqt," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 1, pp. 338–349, 2019.

[36] Linjama, M., Vihtanen, H.-P., Sipola, A., and Vilenius, M., "Secondary controlled multi-chamber hydraulic cylinder," in *The 11th Scandinavian International Conference on Fluid Power SICFP'09, Linköping, Sweden, June 2-4 2009*, 2009, p. 15.

[37] Dell'Amico, A., Carlsson, M., Norlin, E., and Sethson, M., "Investigation of a digital hydraulic actuation system on an excavator arm," in *13th Scandinavian International Conference on Fluid Power, June 3-5, 2013, Linköping, Sweden*, Linköping University Electronic Press, 2013, pp. 505–511.

[38] Heybroek, K. and Sjöberg, J., "Model predictive control of a hydraulic multichamber actuator: A feasibility study," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 3, pp. 1393–1403, 2018.

[39] Alleyne, A. and Hedrick, J. K., "Nonlinear adaptive control of active suspensions," *IEEE transactions on control systems technology*, vol. 3, no. 1, pp. 94–101, 1995.

[40] Yao, B., Bu, F., and Chiu, G.-C., "Nonlinear adaptive robust control of electro-hydraulic servo systems with discontinuous projections," in *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No. 98CH36171)*, IEEE, vol. 2, 1998, pp. 2265–2270.

[41] Guan, C. and Pan, S., "Nonlinear adaptive robust control of single-rod electro-hydraulic actuator with unknown nonlinear parameters," *IEEE transactions on control systems technology*, vol. 16, no. 3, pp. 434–445, 2008.

[42] Narendra, K. S. and Annaswamy, A. M., *Stable adaptive systems*. Courier Corporation, 2012.

[43] Jeronymo, C. E., Yamada, H., and Muto, T., "Adaptive predictive controller for hydraulic systems driven by fast switching on/off solenoid valves," *JSME international journal. Ser. C, Dynamics, control, robotics, design and manufacturing*, vol. 39, no. 3, pp. 522–527, 1996.

[44] Zhao, X., Li, L., Song, J., Li, C., and Gao, X., "Linear control of switching valve in vehicle hydraulic control unit based on sensorless solenoid position estimation," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 7, pp. 4073–4085, 2016.

[45] Lv, C., Wang, H., and Cao, D., "High-precision hydraulic pressure control based on linear pressure-drop modulation in valve critical equilibrium state," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 10, pp. 7984–7993, 2017.

[46] Yang, G., Chen, K., Du, L., Du, J., and Li, B., "Dynamic vacuum pressure tracking control with high-speed on-off valves," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 232, no. 10, pp. 1325–1336, 2018.

[47] Li, H., Huang, Y., Zhu, G., and Lou, Z., "Adaptive lqt valve timing control for an electro-hydraulic variable valve actuator," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 5, pp. 2182–2194, 2018.

[48] Bozza, A., Askari, B., Cavone, G., Carli, R., and Dotoli, M., "An Adaptive Model Predictive Control Approach for Position Tracking and Force Control of a Hydraulic Actuator," in *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, 2022, pp. 1029–1034. DOI: 10.1109/CASE49997.2022.9926645.

[49] Bozza, A., Cavone, G., Carli, R., and Dotoli, M., "A Power Electronic Converters-Inspired Approach for Modeling PWM Switched-Based Nonlinear Hydraulic Servo Actuators," in *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2023, pp. 2477–2482. DOI: 10.1109/SMC53992.2023.10394065.

[50] Shen, X., Zhang, J., Barth, E. J., and Goldfarb, M., "Nonlinear averaging applied to the control of pulse width modulated (pwm) pneumatic systems," in *Proceedings of the 2004 American control conference*, IEEE, vol. 5, 2004, pp. 4444–4448.

[51] Yang, G., Jiang, P., Lei, L., Wu, Y., Du, J., and Li, B., "Adaptive backstepping control of vacuum servo system using high-speed on-off valves," *IEEE Access*, vol. 8, pp. 129 799–129 812, 2020.

[52] Laib, K., Pham, M. T., Lin-Shi, X., and Meghnous, R., "Average modeling and nonlinear observer design for pneumatic actuators with on/off solenoid valves," *Journal of Dynamic Systems, Measurement, and Control*, vol. 144, no. 2, p. 021 006, 2022.

[53] Deng, K., Cong, S., Kong, D., and Shen, H., "Discrete-time direct model reference adaptive control application in a high-precision inertially stabilized platform," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 1, pp. 358–367, 2018.

[54] Komaee, A., "Stabilization of linear systems by pulsewidth modulation of switching actuators," *IEEE Transactions on Automatic Control*, vol. 65, no. 5, pp. 1969–1984, 2019.

[55] Shen, X., Zhang, J., Barth, E. J., and Goldfarb, M., "Nonlinear model-based control of pulse width modulated pneumatic servo systems," 2006.

[56] Kogler, H., "High dynamic digital control for a hydraulic cylinder drive," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 236, no. 2, pp. 382–394, 2022.

[57] Jelali, M. and Kroll, A., *Hydraulic servo-systems: modelling, identification and control*. Springer Science & Business Media, 2002.

[58] Lee, D.-C. and Kim, Y.-S., "Control of single-phase-to-three-phase ac/dc/ac pwm converters for induction motor drives," *IEEE transactions on industrial electronics*, vol. 54, no. 2, pp. 797–804, 2007.

[59] Meena, G. D. and Janardhanan, S., "Taylor–lie formulation based discretization of nonlinear systems," *International Journal of Dynamics and Control*, vol. 6, pp. 459–467, 2018.

[60] Ioannou, P. A. and Sun, J., *Robust adaptive control*. PTR Prentice-Hall Upper Saddle River, NJ, 1996, vol. 1.

[61] Fu, H., Chen, X., Wang, W., and Wu, M., "Mrac for unknown discrete-time nonlinear systems based on supervised neural dynamic programming," *Neurocomputing*, vol. 384, pp. 130–141, 2020.

Chapter 6

Adaptive Constrained Clustering for Real-Time Fault Detection in Industrial Systems

Abstract

Thanks to the pervasive deployment of sensors in Industry 4.0, Data-Driven (DD) methods are recently playing an important role in detecting and diagnosing all potential faults within the plant. In this chapter, a novel *Adaptive Constrained Clustering* (ACC) algorithm is defined to support the real-time fault detection in Industrial Systems (ISs), by clustering the incoming monitoring data into two clusters over time, representing the nominal and non-nominal work conditions, respectively. To this aim, the proposed algorithm relies on a two-stage procedure: *micro-clustering* and *constrained macro-clustering*. The former stage is responsible for grouping the batches of work-cycle data into *micro-clusters*, while the data stream continuously arrives from the data acquisition system. Then, after condensing the micro-clusters into vectors of cluster features, and leveraging on additional knowledge on the nominal and non-nominal working conditions (i.e., constraints on some samples), the second stage aims at offline grouping the micro-clusters features into *macro-clusters*. Experimental results on a real-world industrial case study show that the proposed ACC achieves the same results of offline baseline methods (e.g., Constrained K-means) with a higher responsiveness and processing speed; in comparison to stream baseline methods (e.g., Stream K-means), the proposed approach obtains more accurate and easily interpretable results.

Contents

6.1 Introduction

Due to harsh conditions and severe operations, automatic Industrial Systems (ISs) are prone to failure under unpredictable circumstances. By properly recognizing and addressing possible faults, damages may be reduced, maintenance costs can be minimized, and productivity can be increased [1]. Thus, in the design of automatic systems it is of paramount importance to ensure reliability and dependability [2]. Thereby, process monitoring and fault diagnosis are becoming a fundamental part of modern automation, as outlined in [3]–[5]. As a result, the discipline of Prognostics and Health Management (PHM) is expanding and is essential for implementing predictive maintenance in complex ISs. In this context, Condition Based Monitoring (CBM) is becoming increasingly important in order to improve industrial applications and bring several benefits, such as the reduction of machinery downtime and maintenance costs [6].

Recently, as opposed to model-based techniques [7], Data-Driven (DD) methods have been widely used in ISs to detect possible faults under the realistic hypothesis that

monitoring data is available [8]–[10]. For instance, Bayesian networks and the principal component analysis are respectively used in [11] and [12] as the core component of the fault diagnosis approach. However, within the Machine Learning (ML) context, unsupervised learning (e.g., *clustering*) is one of the most common techniques to interpret and transform raw data into information that can potentially create value in CBM applications. For instance, a clustering analysis successfully supports the identification of the fault severity level in the case of multi-degradation systems where distinguishing among different faults is difficult. Hence, several clustering-based automated techniques are proposed in the related literature to deal with CBM. In [13] the normal operation of a selective laser melting machine – defined as a manufacturing process that uses a metal powder bed and thermal energy supplied by a computer-controlled and focused laser beam to build a work-piece – are compared against three faulty conditions using clustering methods in order to implement a CBM system and enable machine tools for predictive maintenance solutions. In standard clustering algorithms (e.g., K-means) the goal is to divide the original data set into non-intersecting groups such that the distances between the instances of the same group (i.e., intra-clusters) are smaller than the distances with respect to instances of other groups (i.e., inter-clusters). An interesting modification of the traditional K-means is represented by the Constrained K-means, where the learning process is not totally unsupervised. In addition to the original data set, some additional information about the desired clusters are provided: for example, information can be related, but not limited, to must-link and cannot-link constraints.

Nevertheless, all the traditional clustering methods (such as K-means and Constrained K-means) require the availability of the whole data set to build and run the ML models. In the current industrial scenarios, where sensors and Internet of Things devices generate dynamical process data at high speed, producing actionable insights at the right time is challenging [14]. More specifically, when real-time monitoring data arrive over time, conventional techniques used for static data sets are not suitable for online fault detection based on Data Stream (DS) [15]. The goal of DS clustering is to maintain a continuously consistent good clustering of the observed sequence, using a limited amount of memory, time, and information about data. Performing the clustering analysis over DS requires additional challenges, including the need for quick responses and the single-pass constraint over the raw data. Several stream clustering approaches have been proposed in the literature to address the DS clustering problem: they can be mainly categorized into *partitioning-based* (e.g., Leader, Stream K-means, and CluStream), *density-based* (e.g., DenStream), and *grid-based* e.g., DStream) [15]. In particular, the partitioning-based DS clustering techniques are divided, in turn, into adaptive methods (e.g., Leader and Stream K-Means) and Online-Offline clustering methods (e.g., CluStream) [15]. For instance, in [16] a fault-detection system based on data stream prediction is proposed as a DS management system, including different data prediction methods both for short-term and long-term prediction: the achieved results show that the methods using more historical data perform better in long-term prediction, while the methods using the most recent data perform better in short-term prediction. Another peculiarity of DS mining is the so-called *concept drift*, which refers to the unforeseen change in statistical properties of the instances coming from the stream of data over time [17]. Especially when the goal of the clustering activity is to determine a fault detection strategy, understanding how and when the stream of data changes over time is crucial. To this aim, in [18] authors indicate four different concept drift modes: sudden, gradual, incremental, and recurring. A method to detect concept drift in data streams as potential indication for faulty system behaviors is proposed in [19]. Although the results show the effectiveness of the approach, it is worthwhile noting that the concept drift does not always represent a meaningful measure of the degradation of the process under analysis.

6.1.1 Chapter Contribution

In this chapter, a partitioned-based DS clustering approach is used to perform a fault detection strategy in a real-time fashion, aimed at preventing the controller performance

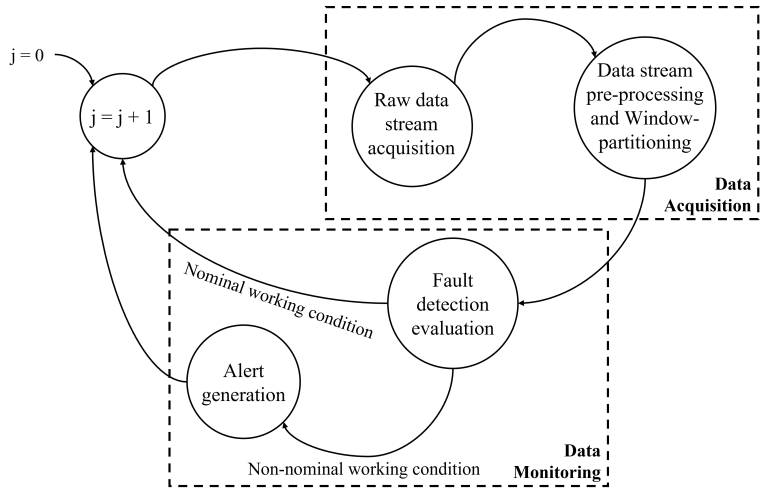

Figure 6.1: Work-flow of the real-time fault detection approach.

deterioration and preserving the overall performance of ISs.

Specifically, the proposed method relies on a cyclical Adaptive Constrained Clustering (ACC), which is composed by two steps: *micro-clustering* and *constrained macro-clustering*. The DS is partitioned over a finite number of batches (also known as windows) of data. In the micro-clustering step, incoming data are grouped into several so-called micro-clusters, which are thus dynamically updated over time. This represents a big challenge in DS mining, usually done by means of common clustering methods (e.g., K-means), where however no control on the final clusters is guaranteed. Moreover, within the DS mining context, challenging issues arise when some constrained-based learning approaches must be applied to clusters which, in turn, are formed over time. To cope with such an issue, additional must-link and cannot-link constraints are thus taken into account during the macro-clustering step in an innovative fashion, while dealing with the vectors of cluster features related to each generated micro-cluster. Then, leveraging the information acquired at the previous stage, the composition of macro-clusters is dynamically updated, so that the incoming monitoring data are grouped into two clusters over the time, representing the nominal and non-nominal work conditions, respectively. Differently from the related literature where unsupervised methods are typically employed in DS mining, the proposed approach combines the use of a large amount of unlabeled data with a limited quantity of additional information. This results in shifting the benefits of the offline constrained clustering to the real-time scenario, with the final aim of discriminating the nominal and non-nominal working conditions of a real industrial machine.

6.2 The Real-Time Fault Detection Framework

Consider an industrial process with a repeated work-cycle, where a number of parameters is periodically monitored through the measurements collected by faultless sensors installed on the machine. It is worthwhile noting that the faultless hypothesis for the sensor results in assuming the retrieved data be accurate and reliable, without any errors or faults. Those parameters are used to perform an analysis on the work conditions of the IS under monitoring. More specifically, the main goal is to define a real-time detection methodology aimed at detecting whether the IS is nominally working or any faulty condition is occurring. The flow-diagram showing the high-level activities performed by the proposed real-time fault detection approach is illustrated in Fig. 6.1.

The overall framework comprises two main stages to be conducted for each iteration j corresponding to a given time window: *Data Acquisition* and *Data Monitoring*. The former is focused on acquiring a continuous stream of raw data from the industrial process, whereas the latter performs the mining activities aimed at extracting information useful

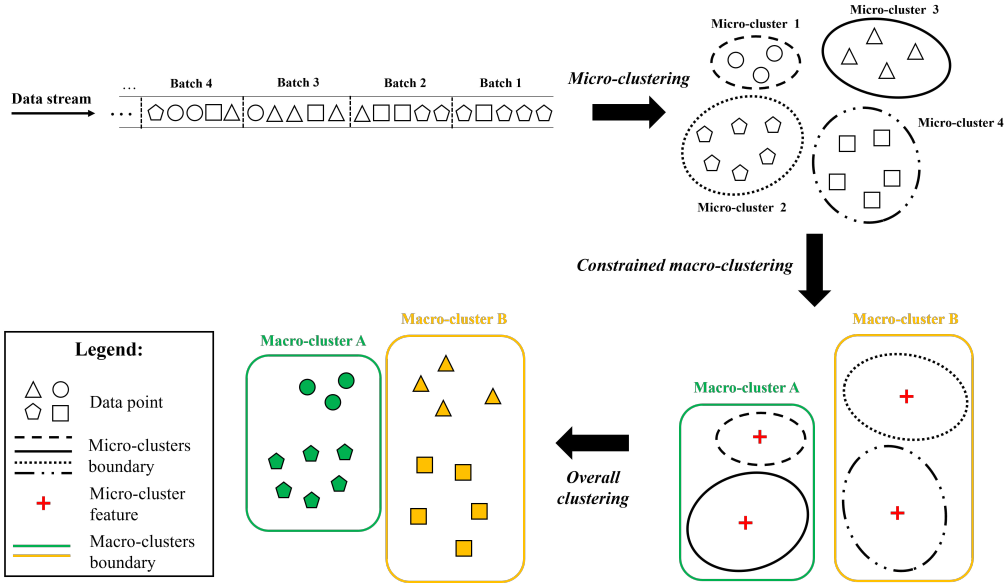

Figure 6.2: Scheme of the proposed ACC algorithm.

for determining the system working state. Hence, during the Data Acquisition step, a continuous DS is retrieved by sensors and, being properly pre-processed and rearranged, data are analyzed during the Data Monitoring step. As long as the process is working in accordance with the nominal behaviour, no further action is taken; conversely, if any faulty condition occurs, an alert is arisen and eventual subsequent PHM actions are immediately triggered.

For the sake of handling the incremental stream of data, the single-pass paradigm is considered, according to which an instance of parameters which has been already processed cannot be processed again (e.g., random access is not allowed) [15], [20], [21]. Suppose that the monitoring information of the work-cycle i is stored in an m-dimensional vector $\vec{X}_i$. Due to memory and computational effort limitation, only a finite group of data extracted from the input stream are analysed at a time. In particular, the stream of data is divided into equal *batches* (or *windows*) containing B data points, in accordance with the so-called landmark window model [15], [17]. As a consequence, the fault detection procedure is iterated by a sampling period $W = B\,T$, where T denotes the cycle time. In the j-th time window only the B instances $\vec{X}_i$ related to the cycle times $i\,T$ belonging to the window $[(j-1)\,W, j\,W]$ are considered, as follows:

$$D_j = \left\{ \vec{X}_i, \ldots, \vec{X}_{i+B-1} \right\}, \qquad \forall j = 1, 2, \ldots \tag{6.1}$$

In this way, it is ensured that two consecutive windows D_j and D_{j+1} $(\forall j = 1, 2, \ldots)$ do not overlap , resulting in allowing real-time data processing. Therefore, a lower memory usage and, at the same time, a faster run-time than traditional static approaches are reached for large data sets.

6.3 The Proposed Adaptive Constrained Clustering

In this section the proposed ACC algorithm to perform the real-time fault detection, in accordance with the problem formulated in Section 6.2, is presented. The basic idea is to cluster the incoming data points into two clusters over time, representing the nominal and non-nominal work conditions, respectively. The cluster of non-nominal conditions groups all the possible faults that may affect the IS dynamics. To this aim, the proposed algorithm relies on a two-stage procedure including the *micro-clustering* and *macro-clustering* steps as shown in Fig. 6.2. The former is responsible of grouping the batches of

Algorithm 6.1 Adaptive Constrained Clustering

Require: Data stream $\mathcal{D}_j, \quad \forall j = 1, 2, \ldots$.

Require: Initial micro-clusters $\{\mathcal{N}_0^1, \ldots, \mathcal{N}_0^{I_0}\} = \emptyset$.

Require: Initialize macro-clusters $\mathcal{M}_0^H = \mathcal{M}_0^F = \emptyset$.

for $j = 1, 2, \ldots$ **do**
$\big(\{\mathcal{N}_j^1, \ldots, \mathcal{N}_j^{I_j}\}, \{\mathrm{CF}_{\mathcal{N}_j^1}, \ldots, \mathrm{CF}_{\mathcal{N}_j^{I_j}}\}\big) = \text{Micro-clustering}\big(\mathcal{D}_j, \{\mathcal{N}_{j-1}^1, \ldots, \mathcal{N}_{j-1}^{I_{j-1}}\}\big)$
$\{\mathcal{M}_j^H, \mathcal{M}_j^F\} = \text{Macro-clustering}\big(\{\mathrm{CF}_{\mathcal{N}_j^1}, \ldots, \mathrm{CF}_{\mathcal{N}_j^{I_j}}\}, \{\mathcal{M}_{j-1}^H, \mathcal{M}_{j-1}^F\}\big)$
end for
return Data clusters $\{\mathcal{M}_j^H, \mathcal{M}_j^F\}, \quad \forall j = 1, 2, \ldots$

Algorithm 6.2 Micro-clustering

Require: Data batch $\mathcal{D}_j$.
Require: Previous micro-clustering $\{\mathcal{N}_{j-1}^1, \ldots, \mathcal{N}_{j-1}^{I_{j-1}}\}$.
Require: Distance threshold $d_{threshold}$.
$I_j \leftarrow I_{j-1}$
$\{\mathcal{N}_j^1, \ldots, \mathcal{N}_j^{I_j}\} \leftarrow \{\mathcal{N}_{j-1}^1, \ldots, \mathcal{N}_{j-1}^{I_{j-1}}\}$
for $\vec{X} \in \mathcal{D}_j$ **do**
$i^* = \min_{i \in 1, \ldots, I_j} \Big(\mathrm{dist}(\vec{X}, \mathcal{N}_j^i)\Big)$
$d^* = \mathrm{dist}(\vec{X}, \mathcal{N}_j^{i^*})$
if $d^* < d_{threshold}$ **then**
$\mathcal{N}_j^{i^*} \leftarrow \mathcal{N}_j^{i^*} \cup \{\vec{X}\}$
else
$I_j \leftarrow I_j + 1$
$\mathcal{N}_j^{I_j} \leftarrow \{\vec{X}\}$
end if
end for
for $\mathcal{N} \in \{\mathcal{N}_j^1, \ldots, \mathcal{N}_j^{I_j}\}$ **do**
Compute $\mathrm{CF}_{\mathcal{N}} = (N_{\mathcal{N}}, \overrightarrow{\mathrm{LS}}_{\mathcal{N}}, \mathrm{SS}_{\mathcal{N}})$ through Eqs. (6.3)-(6.5)
end for
return Updated micro-clustering $\{\mathcal{N}_j^1, \ldots, \mathcal{N}_j^{I_j}\}$.
return Micro-cluster features $\{\mathrm{CF}_{\mathcal{N}_j^1}, \ldots, \mathrm{CF}_{\mathcal{N}_j^{I_j}}\}$.

data into *micro-clusters* while DS continuously arrive from the data acquisition system. Then, after processing each batch of data in the micro-clustering step, the micro-clusters are condensed into vectors of *cluster features*. Finally, leveraging on additional knowledge on the nominal working conditions set (i.e., clustering constraints on some samples), the macro-clustering aims at grouping the micro-clusters features into *macro-clusters*, so that each data point is clustered in accordance with the overall clusters. The procedure is repeatedly executed until the DS process is active.

The overall procedure is formally described in Algorithm 6.1, whilst the micro-clustering and macro-clustering steps – respectively defined in Algorithm 6.2 and 6.3 – are described in detail in the sequel.

6.3.1 Micro-clustering Step

The goal of the primary step is to identify and discover behavioral groups (or clusters) in the DS.

Algorithm 6.3 Macro-clustering

Require: Micro-clustering $\{\mathcal{N}_j^1, \ldots, \mathcal{N}_j^{I_j}\}$.
Require: Micro-cluster features $\{\mathrm{CF}_{\mathcal{N}_j^1}, \ldots, \mathrm{CF}_{\mathcal{N}_j^{I_j}}\}$.
Require: Previous macro-clustering $\{\mathcal{M}_{j-1}^H, \mathcal{M}_{j-1}^F\}$.
Require: Must-link constraints for nominal and non-nominal conditions $\mathcal{H}$ and $\mathcal{F}$
  for $\mathcal{N} \in \{\mathcal{N}_j^1, \ldots, \mathcal{N}_j^{I_j}\}$ **do**
    compute micro-cluster centroid $\vec{X}_{\mathcal{N}}^0$ by Eqs. (6.6)
    $d^H = \mathrm{dist}(\vec{X}_{\mathcal{N}}^0, \mathcal{H})$
    $d^F = \mathrm{dist}(\vec{X}_{\mathcal{N}}^0, \mathcal{F})$
    if $d^H < d^F$ **then**
      $\mathcal{M}_j^H \leftarrow \mathcal{M}_{j-1}^H \cup \{\mathcal{N}\}$
    else
      $\mathcal{M}_j^F \leftarrow \mathcal{M}_{j-1}^F \cup \{\mathcal{N}\}$
    end if
  end for
  return Updated macro-clustering $\{\mathcal{M}_j^H, \mathcal{M}_j^F\}$.

The micro-clustering procedure shown in Algorithm 6.2 is based on the leader clustering, which is a partition-based DS clustering technique [15]. Each batch of data is optimally clustered into micro-clusters using a distance criterion: whenever a new data point $\vec{X}$ arrives from the streams, the cluster i^* having the closest distance d^* is found. If such a distance is below the threshold $d_{threshold}$, the data point $\vec{X}$ is assigned to cluster i^*, otherwise a new cluster (leader) is created, being composed of the single data point $\vec{X}$. Note that no prior information on the number of micro-clusters is needed; however, the performance of the algorithm depends on the correctness of the threshold guess $d_{threshold}$. Indeed, $d_{threshold}$ is used to control the granularity of the micro-clustering and the trade-off between micro-cluster size and number of micro-clusters. A common approach is to set statically such a threshold to low values that capture the local structure of data while ensuring the resulting clusters not to be too small. As an alternative, the Exponentially Weighted Moving Average (EWMA) [22], [23] method can be used to dynamically adjust and fine-tune the $d_{threshold}$ value in real-time, allowing for adaptive and responsive adjustments based on dynamic characteristics and the eventual occurrence of drift in data.

Once the data points in the batch are clustered into micro-clusters, the following features are computed for each micro-cluster. Given a cluster $\mathcal{N}$ composed of data points $\{\vec{X}_i\}$, the vector of corresponding cluster features is defined as the following triple:

$$\mathrm{CF}_{\mathcal{N}} = \left(N_{\mathcal{N}}, \overrightarrow{\mathrm{LS}}_{\mathcal{N}}, \mathrm{SS}_{\mathcal{N}}\right) \tag{6.2}$$

where $N_{\mathcal{N}}$ is the number of points in $\mathcal{N}$, $\overrightarrow{\mathrm{LS}}_{\mathcal{N}}$ is the linear sum of the data points, and $\mathrm{SS}_{\mathcal{N}}$ is the square sum of data points:

$$N_{\mathcal{N}} = |\mathcal{N}| \tag{6.3}$$

$$\overrightarrow{\mathrm{LS}}_{\mathcal{N}} = \sum_{i=1}^{N_{\mathcal{N}}} \vec{X}_i \tag{6.4}$$

$$\mathrm{SS}_{\mathcal{N}} = \sum_{i=1}^{N_{\mathcal{N}}} \vec{X}_i^2 = \sum_{i=1}^{N_{\mathcal{N}}} \left\langle \vec{X}_i, \vec{X}_i \right\rangle. \tag{6.5}$$

6.3.2 Constrained Macro-clustering Step

The goal of the secondary step is to group the micro-clusters obtained by the previous step into macro-clusters based on a given set of constraints, which are additional information

about the data set. As soon as the micro-clustering creates the micro-clusters features over batch data, the macro-clustering step updates the final clusters using the constrained sets. In particular, as formally described in Algorithm 6.3, the observed I micro-clusters are grouped into $K = 2$ predefined groups (i.e., the macro-clusters of nominal and non-nominal conditions, which are denoted at iteration j by $\mathcal{M}_j^H$ and $\mathcal{M}_j^F$, respectively), where an observed micro-cluster is represented by its corresponding cluster feature vector [15]. Indeed, as usually done in constrained clustering [6], the Euclidean distance between each micro-cluster centroid and the must-link constraints sets is calculated in order to update the overall macro-clusters after each batch processing. In particular, knowing the corresponding cluster features, the centroid $\vec{X}_{\mathcal{N}}^0$ is computed in reference to cluster $\mathcal{N}$ as:

$$\vec{X}_{\mathcal{N}}^0 = \frac{\sum_{i=1}^{N_{\mathcal{N}}} \vec{X}_i}{N_{\mathcal{N}}} = \frac{\overrightarrow{LS}_{\mathcal{N}}}{N_{\mathcal{N}}}. \tag{6.6}$$

The constraints sets can be represented by *must-link* (i.e., constraint specifying that two data points must be assigned to the same cluster) and/or *cannot-link* (i.e., constraint specifying that two data points cannot belong to the same cluster) constraints; however, without loss of generality, in Algorithm 6.3 we refer to the must-link set only. Hence, the distance of the must-link samples to micro-clusters is calculated, relying on the information on some samples related to the healthy and faulty behaviors of the considered industrial component (i.e., samples belonging to the macro-cluster of nominal and non-nominal conditions). As a results, the macro-clustering component finds the nominal micro-clusters and merge them in a final cluster of nominal conditions, whilst the remaining micro-clusters are combined in the final cluster of non-nominal conditions.

6.4 Case Study

For the sake of validating the proposed approach, the ACC algorithm is implemented in Python and used to perform the real-time fault detection of a pneumatic system deployed in a real industrial process. In particular, the experiments refer to a robotic work-cell produced by an automotive company leader of the Italian and European market [24]. The whole cell – that is employed to analyze the quality and detect any possible defects of an automotive work-piece performing a 90 seconds work-cycle – consists of an anthropomorphic manipulator and three grippers. Among them, the "rotary gripper" (positioned on the base of the framework) represents the most critical component and, consequently, subject to frequent faults. Hence, it constitutes the case study for the proposed real-time fault detection procedure.

6.4.1 Experimental Setup

The rotary gripper is driven by two pneumatic actuators respectively called "A side" and "B side", whose pneumatic scheme is represented in Fig. 6.3 (a). Each side consists of a linear pneumatic piston and a typical 4-way valve for provisioning the needed air flow-rate into the chambers of the related piston. The gripper is equipped with flow meters, so that during each whole work-cycle both the maximum and mean flow-rate of the air entering the piston chambers are retrieved and collected by an on-board acquisition system. Moreover, the overall working time of each 4-way valve (one for each piston side) is computed as the absolute value of the difference between the time when the actuation command is given to the valve until the time when the related piston rod reaches its final stroke. Therefore, for each work-cycle, six different monitoring features are collected and recorded in the DS, as summarized in Table 6.1. Leveraging on a dimension reduction, only a subset of the monitored features is used in the clustering procedures; namely, as reported in the subsequent analysis on the complete experimental data set, the two most meaningful features correspond to MaxFlowA and MaxFlowB.

The whole experiment is based on a data set of N=600 work-cycles, comprehending 100 nominal and 500 non-nominal operating conditions shuffled in a random fashion. The

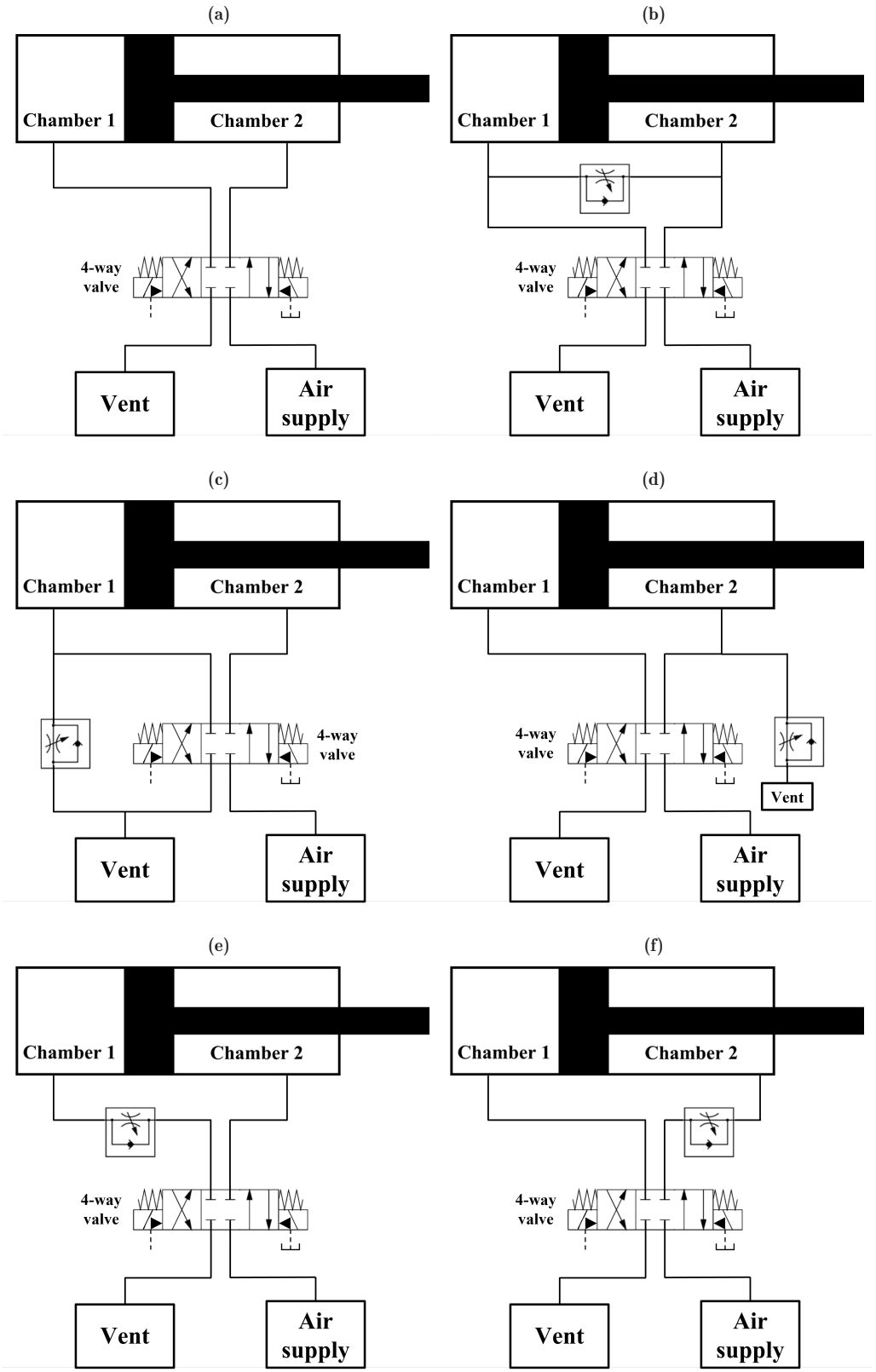

Figure 6.3: Pneumatic schemes of the actuator used for driving the rotary gripper in the case of nominal working condition (a), internal air flow-rate leakage (b), external air flow-rate leakage for the A side (c) and the B side (d), internal occlusion for the A side (e) and the B side (f).

Table 6.1: Data features collected during each work-cycle.

Feature name	Description	Unit
MaxFlowA	Maximum flow-rate on A side piston	l/min
MaxFlowB	Maximum flow-rate on B side piston	l/min
IntegrFlowA	Mean flow-rate on A side piston	l/min
IntegrFlowB	Mean flow-rate on B side piston	l/min
TimeA	Working time of the 4-way valve on A side piston	s
TimeB	Working time of the 4-way valve on B side piston	s

Table 6.2: Description of the fault types addressed by the case study.

Fault type	Number of instances	System state
No fault	100	Nominal
Ext. leakage on side A	100	Non-nominal
Ext. leakage on side B	100	Non-nominal
Int. occlusion on side A	100	Non-nominal
Int. occlusion on side B	100	Non-nominal
Internal leakage	100	Non-nominal

faulty conditions regard the rotary gripper over the complete work-cycles of the robotic cell. Specifically, as described in Table 6.2, five different anomaly conditions are generated. It is worth pointing out that the robotic work-cell under study may also be affected by other less frequent types of faults in addition to the described ones. Nevertheless, only data related to the five considered faults have been provided by the automotive Italian company that produces the work-cell. As indicated in the table, the external leakage of air flow and the internal occlusion are simulated for both pistons of the pneumatic actuator, while the internal leakage is the same for both sides. For the sake of implementing the above-described faulty scenarios, three different configurations of the pneumatic circuit are deployed:

1. Pneumatic circuit generating the internal air flow-rate leakage – An additional valve is inserted between both the outputs of the 4-way valve (Fig. 6.3 (b)).
2. Pneumatic circuit generating the external air flow-rate leakage – Two additional valves are inserted between the air venting source (i.e., flow-rate not in pressure) and chamber 1 of the piston for the A side (Fig. 6.3 (c)) and between the air supply source (i.e., the flow-rate in pressure) and chamber 2 of the piston for the B side (Fig. 6.3 (d)), respectively.
3. Pneumatic circuit generating the internal occlusion – Two additional valves are inserted between the pressure pipeline output of the 4-way valve (i.e., flow-rate not in pressure) and chamber 1 of the piston for the A side (Fig. 6.3 (e)) and between the venting pipeline output of the 4-way valve (i.e., flow-rate not in pressure) and chamber 2 of the piston for the B side (Fig. 6.3 (f)), respectively.

A portion equal to 1% of the dataset is used to define the constrained set on sample working conditions. Specifically, the faulty must-link set $\mathcal{F}$ is formed by picking one sample from each of the five non-nominal categories in Table 6.2, while the healthy must-link set $\mathcal{H}$ is composed of two samples of nominal operating condition. Furthermore, for the sake of evaluating the effectiveness of the proposed approach, the labels are removed

from the entire data set; subsequently, the ACC algorithm is applied to the unlabeled data; finally, the obtained macro-clusters are compared with the true classes (i.e., the *ground truth*), while different clustering methods can be compared through evaluation metrics. The experiment is repeated 100 times by changing both the batch size and the value of the algorithm parameters to show the performance sensitivity.

6.4.2 Performance Evaluation

The performance of the micro-clustering procedure is evaluated using the *Sum of Squared Errors* (SSE) and the *Silhouette Score* (SS), which are internal methods (i.e., they evaluate the clustering performance without any reference). The former measures the sum of squared distances of data points from their cluster centroids:

$$\mathrm{SSQ} = \sum_{k=1}^{I} \sum_{\vec{X}_i \in \mathcal{N}_k} \left\| \vec{X}_i - \vec{X}^0_{\mathcal{N}_k} \right\|^2. \tag{6.7}$$

Conversely, the SS of a data point aims at assessing how well a data point fits into the corresponding micro-cluster compared to others:

$$\mathrm{SS}(\vec{X}_i) = \frac{b(\vec{X}_i) - a(\vec{X}_i)}{\max\{a(\vec{X}_i), b(\vec{X}_i)\}} \tag{6.8}$$

where $a(\vec{X}_i)$ is the average dissimilarity of point $\vec{X}_i$ with all other points in the same cluster, and $b(\vec{X}_i)$ is the lowest average dissimilarity of point $\vec{X}_i$ to any other cluster to which it does not belong. The SS for the overall clustering is the mean of the SS for all data points, as follows:

$$\mathrm{SS} = \sum_{\vec{X}_i \in \{\mathcal{N}_1 \cup \ldots \cup \mathcal{N}_I\}} \frac{\mathrm{SS}(\vec{X}_i)}{N}. \tag{6.9}$$

As for the evaluation of the macro-clustering, the *Purity index* is considered, which is one the most commonly used external methods (i.e., they assess the accuracy of clustering by comparing the results to known ground truth). It is computed as follows:

$$\mathrm{Purity} \ = \frac{1}{N} \sum_{h=1}^{K} \max_{k \in \{H,F\}} \left| \Omega_h \cap \mathcal{M}^k \right| \tag{6.10}$$

where Ω_h (for $h = 1, 2$) denotes the sets of nominal and non-nominal data points with ground truth labels. Note that perfect clustering has a purity index equal to 1, while poor clustering has purity values close to 0.

6.4.3 Results Analysis and Discussion

Figure 6.4 shows the composition of all micro-clusters and two macro-clusters over time obtained in the case of batches with a fixed size equal to $B = 50$ and setting the threshold distance to $d_{threshold} = 1$ (i.e., a maximum size of 100 data points for each microcluster in real-time processing).

It is worthwhile noting two findings: on the one hand, the number and size of micro-clusters (represented in Fig. 6.4 by coloring the corresponding data points) increases over iterations; on the other hand, at each iteration, all data points are grouped into two macro-clusters (represented in Fig. 6.4 by coloring the corresponding micro-clusters), thus allowing to properly distinguish between the nominal and non-nominal working conditions in real-time.

Finally, since the most significant algorithm parameter is the distance threshold $d_{threshold}$, different experiments are analyzed and compared by varying $d_{threshold}$ within the range $[0, 1, 2, 3, 4, 5, 6, 7, 8]$. In particular, for each value of $d_{threshold}$ the number of

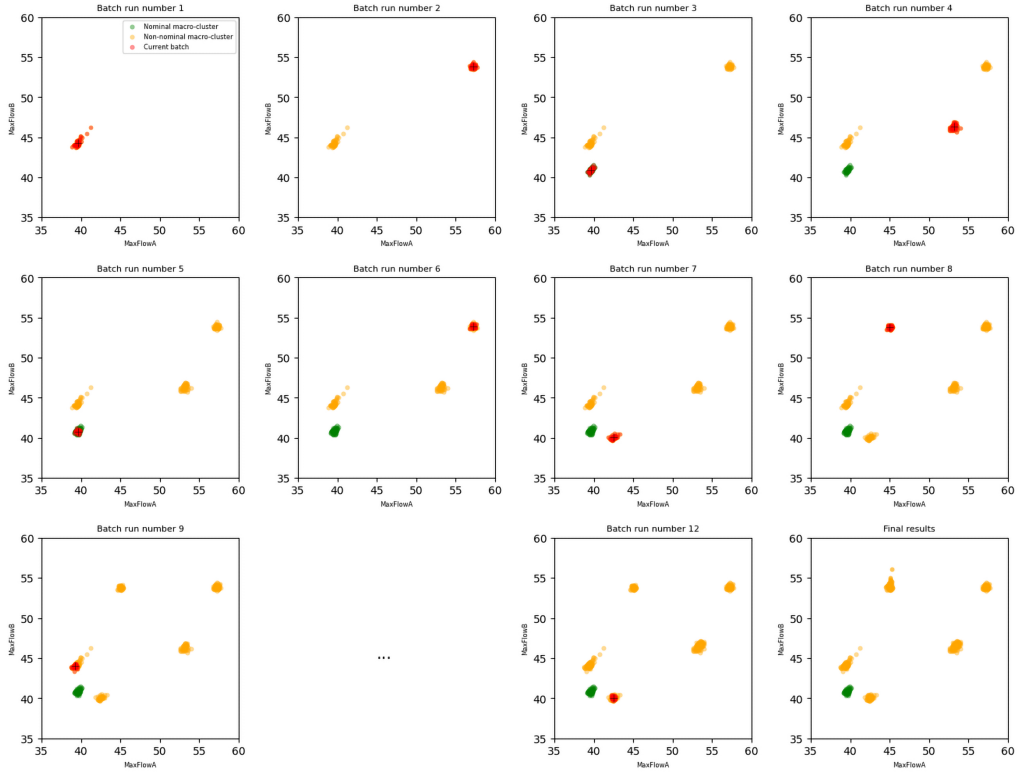

Figure 6.4: Results of the ACC algorithm over batches in the case of batch size $B = 50$. Data points in the current batch are represented by red color, while the green and orange colors denote the micro-clusters belonging to the nominal and non-nominal macro-clusters, respectively.

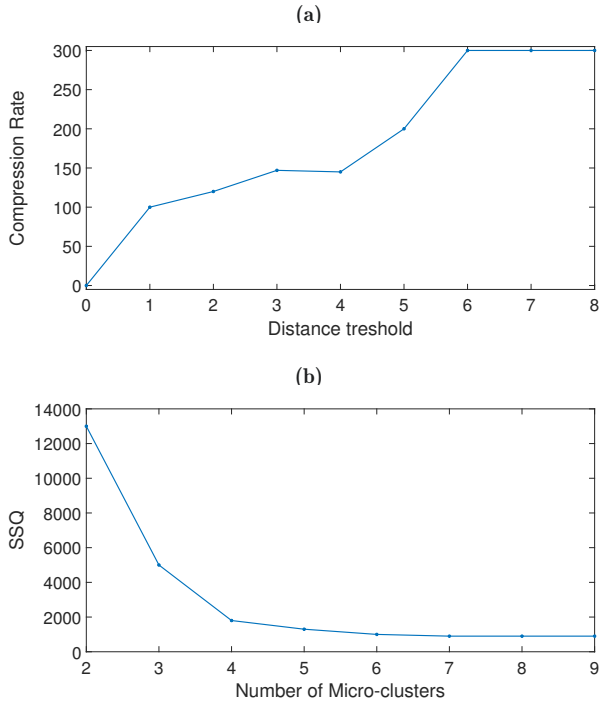

Figure 6.5: Compression rate as a function of radius threshold (a) and sum of square error over the number of clusters (b) for the micro-clustering of the ACC approach.

micro-clusters is evaluated over the whole DS by means of the so-called *compression rate* (i.e., the number of samples divided by the number of the micro-clusters) and the *Elbow*

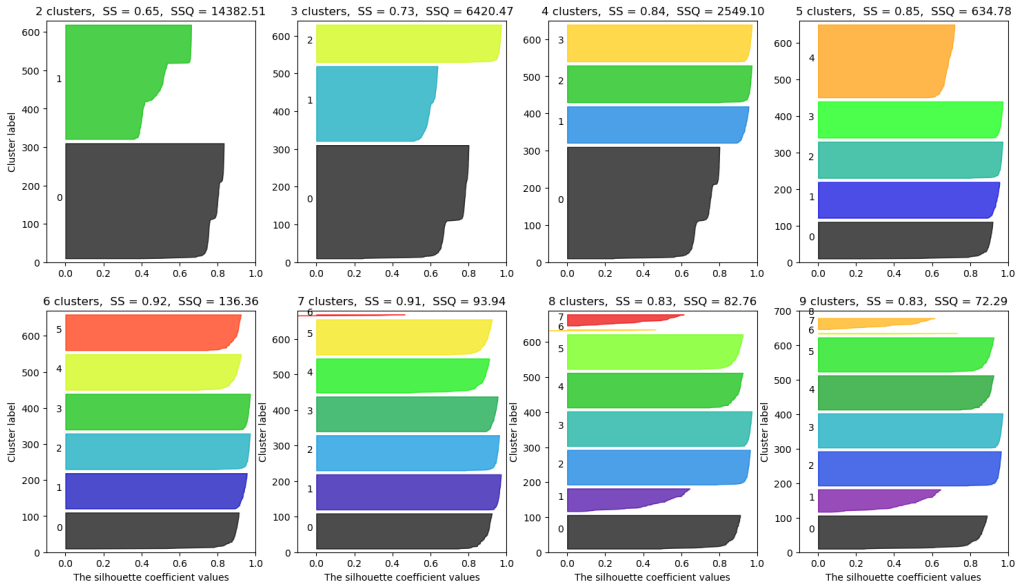

Figure 6.6: Results of the silhouette analysis on micro-clustering varying the number of micro-clusters in the range [2,9].

method (i.e., the heuristics aimed at determining the optimal number of clusters) [15]. From the plots shown in Fig. 6.5 it can be seen that for $d_{threshold} = 1$ the compression rate is equal to 100 (Fig. 6.5 (a)) while the optimal number of micro-clusters is equal to 6 (Fig. 6.5 (b)), implying that each micro-cluster optimally comprises 100 data points. This result is also confirmed by the *Silhouette analysis* that usefully aims at investigating the separation distance between the resulting clusters: Fig. 6.6 reports the average Silhouette score obtained varying the numbers of micro-clusters I, showing that the best values are achieved in the case of 6 clusters.

In addition, as for the macro-clustering, the purity metric is evaluated: the two obtained macro-clusters include respectively 100 and 500 data points, precisely corresponding to the data points representing the nominal and non-nominal classes (ground truth), thus, reaching a purity index equal to 1. This shows the accuracy and reliability of the method in properly classifying the data in accordance with the nominal/non-nominal working conditions.

6.4.4 Comparison with Related Clustering Approaches

In order to show the effectiveness of the proposed ACC, it is compared with both a static (namely, the *Constrained K-means*) and a stream clustering (namely, the Stream K-means) approach.

The Constrained K-means performs a profitable modification of the traditional K-means algorithm: leveraging on background knowledge about the data set, it uses pairwise must-link and cannot-link constraints [25]. Imposing the number of final clusters equal to 2 and using the must-link constraints $\mathcal{F}$ and $\mathcal{H}$, the results are shown in Fig. 6.7. Although it is apparent that the obtained results in terms of final clusters are equal to those achieved by the proposed approach, it is worthwhile noting that our ACC algorithm works in real-time. Therefore, even though Constrained K-means correctly detect the effective working conditions, reaching a purity value equal to 1, it is an offline and therefore slower approach for data mining on the whole data set.

Conversely, the well-known Stream K-means, also called online K-means, is a variant of the classical K-means algorithm designed to handle large data sets [15], [25]. In this approach, the data is processed sequentially in batches, rather than all at once. Each batch is used to update the current set of cluster centroids, which are then refined iteratively. By processing data in a streaming fashion, it is possible to handle large data

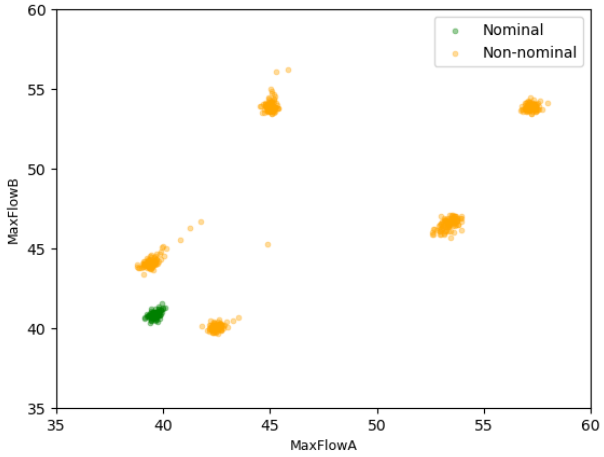

Figure 6.7: Results obtained by the Constrained K-means algorithm with two final clusters.

sets efficiently and in real-time, without requiring the entire data set to be loaded into memory. However, in this approach the number of clusters must be predefined. Moreover, it is worthwhile noting that Stream K-means requires a longer more computational time than our approach due to the iterative process of updating the current set of cluster centroids within each batch. Conversely, the proposed ACC algorithm always maintains a set of $\mathrm{CF}_{\mathcal{N}}$ that summarizes the features of each corresponding micro-cluster, which makes it faster than the Stream K-means, despite requiring more storage memory. Figure 6.8 shows the results of the Stream K-means algorithm applied to the case study, considering batches of 50 points each and two clusters ($K = 2$). It is evident that, during the first iterations, the clusters formed by the first batches are not well-separated; as new batches are added and the algorithm is applied to larger and larger portions of the data, the clusters become less distinct, and begin to overlap in some iterations. In addition, the clusters formed at each iteration are dependent on the order in which the batches of data points are processed. On the contrary, in the ACC method, the order in which the data points arrive does not affect the clustering results at each iteration or at the end of the procedure. Indeed, the Stream K-means algorithm is not very effective at clustering this data set, as the clusters are not well-separated and do not remain stable over time. In terms of purity, the Stream K-means only scores 0.66. Such a result demonstrates that the proposed approach outperforms the other reference streaming method in terms of accuracy.

6.5 Conclusions

This chapter proposes an ACC algorithm for the real-time fault detection of ISs. The aim is detecting the eventual occurrence of faults by clustering data into two groups, representing the nominal and non-nominal work conditions, respectively. To this aim, the proposed methodology is composed of a *micro-clustering* and a *macro-clustering* sub-stages. The former is able to create a certain number of different (micro) clusters by processing the incoming data; besides, some features are extracted at this step. The latter aims at grouping all the micro-clusters previously obtained into two final desired macro-clusters, respectively indicating the nominal and non-nominal working conditions. Finally, the proposed algorithm is tested on a real IS based on two HSAs, outperforming not only a baseline static Constrained K-means, but also an online Stream K-means algorithm.

Future works could consider a dynamical method for updating the distance threshold value (e.g., EWMA) to track DS changes. This could help ensuring that each data point belonging to a micro-cluster is well-represented by the cluster itself, improving the overall accuracy and responsiveness of the whole process. Moreover, an additional fault

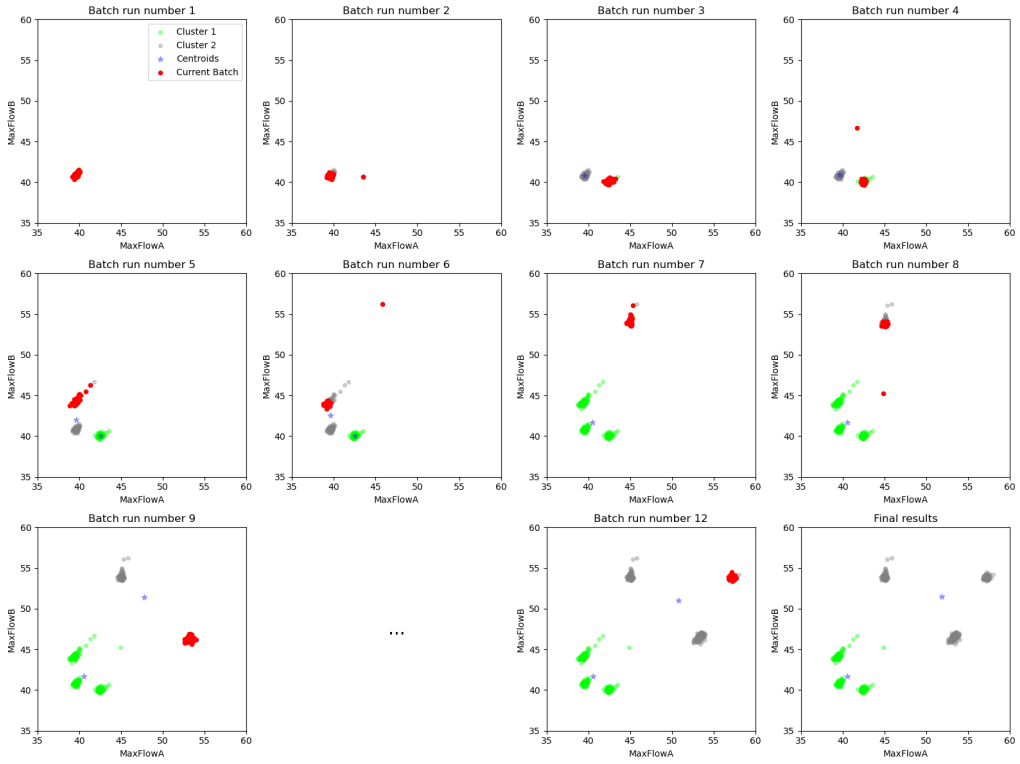

Figure 6.8: Results of the Stream K-means algorithm over batches in the case of batch size $B = 50$. Data points in the current batch are represented by red color, while the light-green and grey colors denote the two clusters, whose updated centroids at each iteration are indicated by the blue stars.

identification sub-stage may be considered for providing more information on the faults root causes, resulting in defining a complete fault diagnosis method.

References

[1] Proia, S., Carli, R., Cavone, G., and Dotoli, M., "Control techniques for safe, ergonomic, and efficient human-robot collaboration in the digital industry: A survey," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 3, pp. 1798–1819, 2022. DOI: 10.1109/TASE.2021.3131011.

[2] Ding, S. X., *Model-based fault diagnosis techniques: design schemes, algorithms, and tools*. Springer Science & Business Media, 2008.

[3] Park, Y.-J., Fan, S.-K. S., and Hsu, C.-Y., "A review on fault detection and process diagnostics in industrial processes," *Processes*, vol. 8, no. 9, p. 1123, 2020.

[4] Abid, A., Khan, M. T., and Iqbal, J., "A review on fault detection and diagnosis techniques: Basics and beyond," *Artificial Intelligence Review*, vol. 54, pp. 3639–3664, 2021.

[5] Okada, K. F. Á., Morais, A. S. de, Oliveira-Lopes, L. C., and Ribeiro, L., "A survey on fault detection and diagnosis methods," in *2021 14th IEEE International Conference on Industry Applications (INDUSCON)*, IEEE, 2021, pp. 1422–1429.

[6] Askari, B., Carli, R., Cavone, G., and Dotoli, M., "Data-driven fault diagnosis in a complex hydraulic system based on early classification," 40, 1st IFAC Workshop on Control of Complex Systems COSY 2022, vol. 55, 2022, pp. 187–192. DOI: https://doi.org/10.1016/j.ifacol.2023.01.070.

[7] Incremona, G. P. and Ferrara, A., "Fault diagnosis for robot manipulators via vision servoing based suboptimal second order sliding mode," in *2019 18th European Control Conference (ECC)*, IEEE, 2019, pp. 3090–3095.

[8] Dalmarco, G., Ramalho, F. R., Barros, A. C., and Soares, A. L., "Providing industry 4.0 technologies: The case of a production technology cluster," *The journal of high technology management research*, vol. 30, no. 2, p. 100 355, 2019.

[9] Cenedese, A., Luvisotto, M., and Michieletto, G., "Distributed clustering strategies in industrial wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 1, pp. 228–237, 2017. DOI: 10.1109/TII.2016.2628409.

[10] Fontes, C. H. and Budman, H., "A hybrid clustering approach for multivariate time series–a case study applied to failure analysis in a gas turbine," *ISA transactions*, vol. 71, pp. 513–529, 2017.

[11] Atoui, M. A. and Cohen, A., "Fault diagnosis using pca-bayesian network classifier with unknown faults," in *2020 European Control Conference (ECC)*, IEEE, 2020, pp. 2039–2044.

[12] Wu, C., Yue, J., Wang, L., and Lyu, F., "Fault diagnosis of recessive weakness in superbuck converter based on kpca-ipnn," in *2020 European Control Conference (ECC)*, IEEE, 2020, pp. 2045–2050.

[13] Uhlmann, E., Pontes, R. P., Geisert, C., and Hohwieler, E., "Cluster identification of sensor data for predictive maintenance in a selective laser melting machine tool," *Procedia Manufacturing*, vol. 24, pp. 60–65, 2018, 4th International Conference on System-Integrated Intelligence: Intelligent, Flexible and Connected Systems in Products and Production.

[14] Diaz-Rozo, J., Bielza, C., and Larrañaga, P., "Clustering of data streams with dynamic gaussian mixture models: An iot application in industrial processes," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3533–3547, 2018. DOI: 10.1109/JIOT.2018.2840129.

[15] Mansalis, S., Ntoutsi, E., Pelekis, N., and Theodoridis, Y., "An evaluation of data stream clustering algorithms," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 11, no. 4, pp. 167–187, 2018.

[16] Alzghoul, A., Löfstrand, M., and Backe, B., "Data stream forecasting for system fault prediction," *Computers & industrial engineering*, vol. 62, no. 4, pp. 972–978, 2012.

[17] Zubaroğlu, A. and Atalay, V., "Data stream clustering: A review," *Artificial Intelligence Review*, vol. 54, no. 2, pp. 1201–1236, 2021.

[18] Ramírez-Gallego, S., Krawczyk, B., García, S., Woźniak, M., and Herrera, F., "A survey on data preprocessing for data stream mining: Current status and future directions," *Neurocomputing*, vol. 239, pp. 39–57, 2017.

[19] Zenisek, J., Holzinger, F., and Affenzeller, M., "Machine learning based concept drift detection for predictive maintenance," *Computers & Industrial Engineering*, vol. 137, pp. 60–65, 2019.

[20] Silva, J. A., Faria, E. R., Barros, R. C., Hruschka, E. R., Carvalho, A. C. d., and Gama, J., "Data stream clustering: A survey," *ACM Computing Surveys (CSUR)*, vol. 46, no. 1, pp. 1–31, 2013.

[21] Ghesmoune, M., Lebbah, M., and Azzag, H., "State-of-the-art on clustering data streams," *Big Data Analytics*, vol. 1, no. 1, pp. 1–27, 2016.

[22] Agrahari, S. and Singh, A. K., "Concept drift detection in data stream mining: A literature review," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 10, pp. 9523–9540, 2022.

[23] Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., and Zhang, G., "Learning under concept drift: A review," *IEEE transactions on knowledge and data engineering*, vol. 31, no. 12, pp. 2346–2363, 2018.

[24] *Masmec*, https://www.masmec.com/en/, Accessed: 2023-04-01.

[25] Ikotun, A. M., Ezugwu, A. E., Abualigah, L., Abuhaija, B., and Heming, J., "K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data," *Information Sciences*, 2022.

Chapter 7

Conclusions

Industrial Systems (ISs) are increasingly implementing the so-called Industry 4.0 paradigm, advancing toward smart manufacturing to meet high production demand, fast-changing requirements, and enhanced connectivity. One promising strategy is the the real-time digital representation of the production environment, which partially addresses these goals. Most of the related literature focuses on the *Digital Twin* (DT) concept, which offers significant advantages by providing perception and cognition abilities in autonomous and intelligent production systems. However, traditional model-based methods are often unpractical for capturing the intrinsic complexity (e.g., nonlinearity and uncertainty) of these systems, due to significant growing of process data volumes. This challenge has recently motivated the adoption of identification techniques aimed at obtaining dynamical models to develop *Data-Driven* (DD) control strategies for monitoring and controlling ISs.

This thesis has explored two main research directions related to the implementation of DD tools for process control and performance preservation of industrial systems.

The first part has introduced *indirect DD process control* schemes based on dynamical models obtained through system identification. We have presented novel optimization-based methodologies for controlling nonlinear systems widely used for industrial processes, in a robust and stable fashion. The two specific contributions are reported hereafter.

- The indirect DD process control methodology based on Model Predictive Control for the deep drawing process presented in Chapter 2 aims at improving the overall efficiency, fault tolerance, and quality of the final workpiece. Numerical experiments in a DT-based software-in-the-loop simulation demonstrate its effectiveness in predicting the profile of the metal sheet to avoid quality defects in stamped parts.
- The DD robust control approach for nonlinear input-affine systems presented in Chapter 3 allows to infer the unknown nonlinear dynamics by directly solving a Semi-Definite Programming (SDP) problem online, without requiring any approximation. The SDP framework also ensures closed-loop stability by constraining the descent of a proper Lyapunov function over time.

Despite the promising results, the first part of this thesis highlights open problems and suggests future research directions:

- The predictive-based process control approach proposed in Chapter 2 lacks adaptive mechanisms to autonomously recognize the operating conditions of the plant. To this aim, future research developments could integrate the process controller into a prognostic and health management system, aligned to the DT paradigm.
- The direct DD robust control for nonlinear systems presented in Chapter 3 may encounter feasibility issues in solving the SDP problem if addressing non-constant reference trajectories. Potential improvements could include using a reference governor, relaxing the Lyapunov condition (upon trajectory changes), or exploiting the previous estimation values to adjust the regression online.

The second part of the thesis has focused on DD methods for real-time monitoring of ISs through adaptive approaches with real-time data. This part has introduced both innovative *adaptive control* strategies for unknown parameters estimation of Hydraulic Servo Actuators (HSAs) and original methods for the real-time *fault detection* of ISs. The specific contributions of each chapter are reported hereafter.

- The Adaptive Model Predictive Control (AMPC) approach presented in Chapter 4 offers a practical and viable solution for controlling HSAs based on *flow control valves*. The nonlinear dynamics is approximated at each iteration with a linear time-varying model through the iterative linearization paradigm used to update the AMPC internal model for effectively provisioning a predicted control action on the system.
- Chapter 5 presents another adaptive control framework for digital multi-chamber HSAs based on *pressure control valves*, providing a twofold contribution. On the one hand, the modeling approach of such systems is streamlined thanks to a novel Average Equivalent Discrete-time Model based on Pulse Width Modulation. On the other hand, a cascaded control-loop architecture based on Model Reference Adaptive Control is designed to guarantee higher energy efficiency, simplify the control architecture, and face all model uncertainties.
- The Adaptive Constrained Clustering (ACC) algorithm presented in Chapter 6 allows clustering the incoming monitoring data for distinguishing nominal from non-nominal working conditions in ISs. Our solution combines the use of a large amount of unlabeled data with a limited quantity of additional information, exploiting the benefits of offline constrained clustering, as well those of data stream clustering approaches to perform a real-time fault detection strategies.

Notwithstanding, open problems in the second part of the thesis point to additional research directions:

- The proposed adaptive control methods described in Chapter 4 and 5 could be enhanced to improve the controller robustness by addressing factors such as leakage, parameter projection, and dead zones. In addition, future works may consider to extend the control architecture to multi-pressure HSAs.
- The ACC algorithm presented in Chapter 6 lacks integrated fault isolation and analysis techniques to provide root-cause insights, making the development of a complete fault diagnosis method an important future objective.

Appendices

Appendix A

Preliminaries

The goal of this appendix is to give a brief overview on the theoretical preliminaries required by this thesis and, specifically, *system identification* methods, *adaptive control* techniques, and the *single-phase DC/AC inverter modeling* approach.

A.1 System Identification

The formulation of a typical identification problem considers the following Multiple-Input Multiple-Output (MIMO) Linear Time-Invariant (LTI) discrete-time system:

$$\begin{cases} \boldsymbol{x}(k+1) = \mathbf{A}\boldsymbol{x}(k) + \mathbf{B}\boldsymbol{u}(k) \\ \boldsymbol{y}(k) = \mathbf{C}\boldsymbol{x}(k) + \mathbf{D}\boldsymbol{u}(k) \end{cases} \tag{A.1}$$

where $\boldsymbol{u}(k) \in \mathbb{R}^m$, $\boldsymbol{y}(k) \in \mathbb{R}^p$, and $\boldsymbol{x}(k) \in \mathbb{R}^n$ represent respectively the input, output, and state vectors, having initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0$, $\forall k \in \mathbb{R}^+$. A solution of system (A.1) is given by

$$\begin{cases} \boldsymbol{x}(k) = \underbrace{\mathbf{A}^k \boldsymbol{x}_0}_{\text{Unforced state response}} + \underbrace{\sum_{\tau=0}^{k-1} \mathbf{A}^{(k-\tau-1)} \mathbf{B}\boldsymbol{u}(\tau)}_{\text{Forced state response}} & \text{(A.2a)} \\ \boldsymbol{y}(k) = \underbrace{\mathbf{C}^k \boldsymbol{x}_0}_{\text{Unforced output response}} + \underbrace{\sum_{\tau=0}^{k-1} \mathbf{C}\mathbf{A}^{(k-\tau-1)} \mathbf{B}\boldsymbol{u}(\tau) + \mathbf{D}\boldsymbol{u}(k)}_{\text{Forced output response}} . & \text{(A.2b)} \end{cases}$$

Consider for the moment the Single-Input Single-Output (SISO) case. If we evaluate the response of system (A.1) to the *impulse*

$$u(k) = \delta(k) = \begin{cases} 1 & k = 0 \\ 0 & k > 0 \end{cases},$$

then, the output signal becomes

$$y(k) = w(k) = \begin{cases} \mathbf{C}\mathbf{A}^{(k-1)}\mathbf{B} & k = 0 \\ \mathbf{D} & k > 0 \end{cases}.$$

If we consider the impulse response of each output related to the corresponding input signal, the following matrix $\mathbf{W}(k)$ can be built:

$$\mathbf{W}(k) = \begin{pmatrix} w_{1,1}(k) & \dots & w_{1,m} \\ w_{2,1}(k) & \dots & w_{2,m} \\ \vdots & \ddots & \vdots \\ w_{p,1}(k) & \dots & w_{p,q} \end{pmatrix} \in \mathbb{R}^{p \times m} \tag{A.3}$$

where the i,j-th $w(k)_{i,j}$ element represents the i-th response to the generic column-element input $u_{i,j}(k) \in \mathbb{R}^{p \times 1}$ defined as

$$u_{i,j}(k) = \begin{cases} \delta(k) & i = j \\ 0 & i \neq j \end{cases}, \qquad \forall i = 1, \dots, p, \quad \forall j = 1, \dots, m.$$

By applying the Z-Transform to (A.3), we obtain the transfer matrix of (A.1):

$$\mathbf{W}(z) = \mathbf{C}(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}. \tag{A.4}$$

Consequently, system (A.1) can be rewritten as a direct relation between the input $\boldsymbol{u}(k)$ and the output $\boldsymbol{y}(k)$ as follows:

$$\mathbf{Y}(z) = \mathbf{W}(z)\mathbf{U}(z) \tag{A.5}$$

where $\mathbf{Y}(z)$ and $\mathbf{U}(z)$ represent respectively the Z-Transform of the output and input signal vectors.

Remark A.1.1

If we impose a change of coordinates characterized by $\widetilde{x}(k) = \mathbf{T}^{-1}x(k)$, through the non singular matrix $\mathbf{T} \in \mathbb{R}^{n\times n}$, such that the system can be rewritten as

$$\begin{cases} \widetilde{\boldsymbol{x}}(k) = & \underbrace{\mathbf{T}^{-1}\mathbf{A}\mathbf{T}}_{\widetilde{\mathbf{A}}}\widetilde{\boldsymbol{x}}(k) \; + \; \underbrace{\mathbf{T}^{-1}\mathbf{B}\mathbf{T}}_{\widetilde{\mathbf{B}}}\widetilde{\boldsymbol{u}}(k) \\ y(k) = & \underbrace{\mathbf{C}\mathbf{T}}_{\widetilde{\mathbf{C}}}\widetilde{\boldsymbol{x}}(k) \; + \; \mathbf{D}\widetilde{\boldsymbol{u}}(k) \end{cases} \tag{A.6}$$

where $\widetilde{\mathbf{A}}$, $\widetilde{\mathbf{B}}$, and $\widetilde{\mathbf{C}}$ represent the new state, input, and output matrices, respectively. Thus, the impulse response of system (A.6) is

$$\boldsymbol{y}(k) = \widetilde{\boldsymbol{w}}(k) = \begin{cases} \widetilde{\mathbf{C}}\widetilde{\mathbf{A}}^{(k-1)}\widetilde{\mathbf{B}} & k > 0 \\ \mathbf{D} & k = 0. \end{cases} \tag{A.7}$$

Thereby, it is easy to see that the impulse response (A.7) and –consequently– the transfer matrix $\mathbf{W}(z)$ of the system is not affected by the change of coordinates in the state space. □

Definition A.1.1 (*Persistence of excitation, [1]*)

Given the training dataset $\{\boldsymbol{u}(k), \boldsymbol{y}(k+1)\}_{k=0}^{\mathrm{N}}$ and the so-called Hankel matrix $\mathbf{H}_{\mathrm{L,N}}(u)$

$$\mathbf{H}_{\mathrm{L,N}}(\boldsymbol{u}) = \begin{pmatrix} \boldsymbol{u}(0) & \boldsymbol{u}(1) & \dots & \boldsymbol{u}(\mathrm{N}-\mathrm{L}) \\ \boldsymbol{u}(1) & \boldsymbol{u}(2) & \dots & \boldsymbol{u}(\mathrm{N}-\mathrm{L}+1) \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{u}(\mathrm{L}-1) & \boldsymbol{u}(\mathrm{L}) & \dots & \boldsymbol{u}(\mathrm{N}-1) \end{pmatrix} \in \mathbb{R}^{m\mathrm{L}\times(\mathrm{N}-\mathrm{L}+1)},$$

the input $\{\boldsymbol{u}(k)\}_{k=0}^{\mathrm{N}-1}$ is said to be Persistently Exciting (PE) of order L if $\operatorname{rank}\big(\mathbf{H}_{\mathrm{L,N}}(\boldsymbol{u})\big) = m\mathrm{L}$, i.e. the Hankel matrix has full rank. □

Let us consider, now, the SISO case. In that case, the transfer matrix (A.4) can be directly rewritten as a *transfer function*:

$$\mathrm{W}(z) = \frac{\mathrm{B}(z)}{\mathrm{A}(z)} \tag{A.8}$$

where $\mathrm{B}(z)$ and $\mathrm{A}(z)$ are two polynomials given by

$$\mathrm{B}(z) = \mathrm{b}_{n-m}z^{m-n} + \cdots + \mathrm{b}_{n-1}z^{-n+1} + \mathrm{b}_n z^{-n} \tag{A.9a}$$

$$\mathrm{A}(z) = 1 - \mathrm{a}_1 z^{-1} + \cdots - \mathrm{a}_{n-1}z^{-n+1} - \mathrm{a}_n z^{-n} \tag{A.9b}$$

with $m \leq n$, where $n-m$ is the *relative degree* of system (A.4). For the sake of expressing the input-output relationship (A.5) in the time domain (i.e., getting rid of the complex frequency z), the following-defined *shift* (or *delay*) *operator* q is introduced:

$$q : q^{-\tau}f(k) = f(k-\tau) \quad \tau \geq 0, \tag{A.10}$$

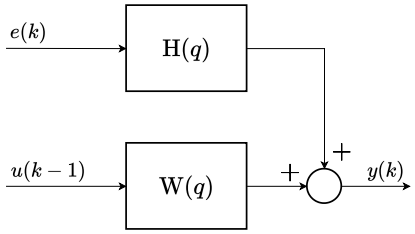

Figure A.1: Typical Box-Jenkins model.

which provides a connection between the complex frequency z and the discrete-time domain k. Hence, by using (A.10), system (A.5) can be easily rewritten as

$$y(k) = \mathrm{W}(q)u(k) \tag{A.11}$$

with

$$\mathrm{W}(q) = \frac{\mathrm{B}(q)}{\mathrm{A}(q)} \tag{A.12}$$

where $\mathrm{B}(q)$ and $\mathrm{A}(q)$ are respectively given by (A.9a)-(A.9b), but rewritten in the so-called shift operator domain q:

$$\mathrm{B}(q) = \mathrm{b}_{n-m}q^{m-n} + \cdots + \mathrm{b}_{n-1}q^{-n+1} + \mathrm{b}_n q^{-n} \tag{A.13a}$$

$$\mathrm{A}(q) = 1 - \mathrm{a}_1 q^{-1} + \cdots - \mathrm{a}_{n-1}q^{-n+1} - \mathrm{a}_n q^{-n}. \tag{A.13b}$$

As a consequence, the following equivalence holds for the SISO system (A.11):

$$\mathrm{A}(q)y(k) = \mathrm{B}(q)u(k).$$

Consider the following SISO LTI discrete-time system represented in Fig. A.1:

$$y(k) = \mathrm{W}(q)u(k) + \mathrm{H}(q)e(k)$$

where $e(k)$ is a with Gaussian noise having the *mean value* μ and *variance* σ^2 respectively as

$$\mu = \mathrm{E}\big[e(k)\big] = 0$$

$$\sigma^2 = \mathrm{E}\big[e(i)e(j)\big] = \begin{cases} 1 & i = j \\ 0 & \text{otherwise} \end{cases},$$

while $\mathrm{W}(q) = \frac{\mathrm{B}(q)}{\mathrm{A}(q)}$, with $\mathrm{B}(q)$ and $\mathrm{A}(q)$ defined in (A.13a)-(A.13b). On the contrary, $\mathrm{H}(q) = \frac{\mathrm{C}(q)}{\mathrm{D}(q)}$ represents the transfer function of the "output error system", where $\mathrm{C}(q)$ and $\mathrm{D}(q)$ have a similar structure of $\mathrm{A}(q)$, i.e.:

$$\mathrm{C}(q) = 1 + \mathrm{c}_1 q^{-1} + \cdots + \mathrm{c}_{n-1}q^{-n+1} + \mathrm{c}_n q^{-n}$$

$$\mathrm{D}(q) = 1 + \mathrm{d}_1 q^{-1} + \cdots - \mathrm{d}_{n-1}q^{-n+1} - \mathrm{d}_n q^{-n}.$$

Let us consider the case where $m = n - 1$ (i.e., $\mathrm{W}(q)$ is strictly proper and has the same number of poles and zeros):

$$\mathrm{B}(q) = \mathrm{b}_1 q^{-1} + \mathrm{b}_2 q^{-2} + \cdots + \mathrm{b}_{n-1}q^{-n+1} + \mathrm{b}_n q^{-n} \tag{A.16a}$$

$$\mathrm{A}(q) = 1 - \mathrm{a}_1 q^{-1} + \cdots - \mathrm{a}_{n-1}q^{-n+1} - \mathrm{a}_n q^{-n}. \tag{A.16b}$$

As for $\mathrm{H}(q)$, instead, according to how it is structured we can have different models belonging to the more general so-called *Box-Jenkins* family (i.e., class of models):

$$y(k) = \frac{\mathrm{B}(q)}{\mathrm{A}(q)}u(k) + \frac{\mathrm{C}(q)}{\mathrm{D}(q)}e(k). \tag{A.17}$$

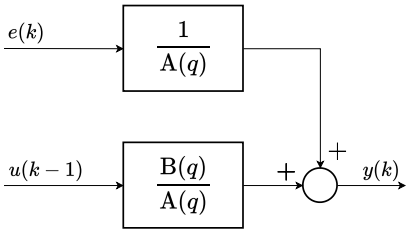

Figure A.2: Typical ARX model.

A.1.1 Box-Jenkins' Models

Box-Jenkins' models represent a cornerstone in the field of system identification, especially for linear systems in the form of (A.1). These models are particularly renowned for their efficacy in handling time series data by incorporating both autoregressive and moving average components, alongside exogenous inputs [2].

A.1.1.1 ARX Models

The simplest Box-Jenkins family's model is given by the AutoRegressive eXogenous (ARX), whose block scheme is represented in Fig. A.2. Such class of models leverage past values of the output and both current and past values of the input to predict future outputs [3]. It is characterized by $\mathrm{C}(q) = 1$ and $\mathrm{D}(q) = \mathrm{A}(q)$, resulting in the following expression:

$$y(k) = \frac{\mathrm{B}(q)}{\mathrm{A}(q)}u(k) + \frac{1}{\mathrm{A}(q)}e(k)$$

which, by multiplying both sides by $\mathrm{A}(q)$, can be rewritten as

$$\mathrm{A}(q)y(k) = \mathrm{B}(q)u(k) + e(k). \tag{A.18}$$

By substituting (A.16a)-(A.16b) in (A.18), we obtain

$$\underbrace{\left(1 - \mathrm{a}_1 q^{-1} + \cdots - \mathrm{a}_n q^{-n}\right)}_{\mathrm{A}(q)} y(k) = \underbrace{\left(\mathrm{b}_1 q^{-1} + \mathrm{b}_2 q^{-2} + \cdots + \mathrm{b}_n q^{-n}\right)}_{\mathrm{B}(q)} u(k) + e(k)$$

which can be easily rewritten as

$$y(k) = \underbrace{\mathrm{a}_1 q^{-1} y(k) + \cdots + \mathrm{a}_n q^{-n} y(k)}_{\text{Autoregressive}} + \underbrace{\mathrm{b}_1 q^{-1} u(k) + \mathrm{b}_2 q^{-2} u(k) + \cdots + \mathrm{b}_n q^{-n} u(k)}_{\text{Exogenous}} + e(k),$$

where the *autoregressive* and the *exogenous* terms are highlighted. Finally, if we collect $y(k)$ and $u(k)$ in the previous equations, it becomes

$$y(k) = \big(1 - \mathrm{A}(q)\big)y(k) + \mathrm{B}(q)u(k) + e(k). \tag{A.19}$$

From equation (A.19), we have to point out that each sample of an autoregressive signal is evaluated as a linear combination of the all previous samples, i.e., it can be viewed as an weighted average of all of them. Moreover, being (A.19) characterized by two different polynomials, i.e., $\mathrm{A}(q)$ and $\mathrm{B}(q)$, it is called to have two degrees of freedom.

A.1.1.2 ARMAX Models

Another Box-Jenkins family's model is given by the AutoRegressive Moving Average eXogenous (ARMAX), whose block scheme is represented in Fig. A.3. In addition to ARX models, this class includes a moving average component, which accounts for the noise structure in the system, thereby providing a better prediction capability [3]. It is characterized by a moving average of the error, resulting in $\mathrm{C}(q) \neq 1$ (with $\mathrm{D}(q) = \mathrm{A}(q)$):

$$y(k) = \frac{\mathrm{B}(q)}{\mathrm{A}(q)}u(k) + \frac{\mathrm{C}(q)}{\mathrm{A}(q)}e(k)$$

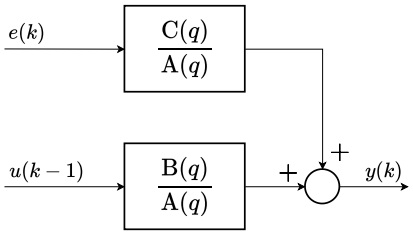

Figure A.3: Typical ARMAX model.

which, also in this case, by multiplying both sides by $\mathrm{A}(q)$, can be rewritten as

$$\mathrm{A}(q)y(k) = \mathrm{B}(q)u(k) + \mathrm{C}(q)e(k). \tag{A.20}$$

By substituting (A.16a)-(A.16b) in (A.20), we obtain

$$\underbrace{\left(1 - \mathrm{a}_1 q^{-1} + \cdots - \mathrm{a}_n q^{-n}\right)}_{\mathrm{A}(q)} y(k) = \underbrace{\left(\mathrm{b}_1 q^{-1} + \mathrm{b}_2 q^{-2} + \cdots + \mathrm{b}_n q^{-n}\right)}_{\mathrm{B}(q)} u(k) + \underbrace{\left(1 + \mathrm{c}_1 q^{-1} + \cdots + \mathrm{c}_n q^{-n}\right)}_{\mathrm{C}(q)} e(k)$$

which can be easily rewritten as

$$y(k) = \underbrace{\mathrm{a}_1 q^{-1} y(k) + \cdots + \mathrm{a}_n q^{-n} y(k)}_{\text{Autoregressive}} + \underbrace{\mathrm{b}_1 q^{-1} u(k) + \mathrm{b}_2 q^{-2} u(k) + \cdots + \mathrm{b}_n q^{-n} u(k)}_{\text{Exogenous}} + \underbrace{e(k) + \mathrm{c}_1 q^{-1} e(k) + \cdots + \mathrm{c}_n q^{-n} e(k)}_{\text{Moving average}}$$

where the *autoregressive*, the *exogenous*, and the *moving average* terms are highlighted. Finally, if we collect $y(k)$, $u(k)$, and $e(k)$ in the previous equations, it becomes

$$y(k) = \big(1 - \mathrm{A}(q)\big)y(k) + \mathrm{B}(q)u(k) + \mathrm{C}(q)e(k). \tag{A.21}$$

A.1.2 Nonlinear Models

The formulation of a typical identification problem for nonlinear systems considers the following MIMO input-affine dynamics [4]:

$$\begin{cases} \dot{\boldsymbol{x}}(t) = f\big(\boldsymbol{x}(t)\big) + g\big(\boldsymbol{x}(t)\big)\boldsymbol{u}(t) \\ \boldsymbol{y}(t) = h\big(\boldsymbol{x}(t)\big) \end{cases} \tag{A.22}$$

where $\boldsymbol{x}(t) \in \mathcal{X} \subseteq \mathbb{R}^n$, $\boldsymbol{u}(t) \in \mathcal{U} \subseteq \mathbb{R}^m$, and $\boldsymbol{y}(t) \in \mathcal{X} \subseteq \mathbb{R}^p$ represent respectively the vector of n states, m inputs, and p outputs, while $f : \mathcal{X} \to \mathbb{R}^n$, $g : \mathcal{U} \to \mathbb{R}^m$, and $h : \mathcal{X} \to \mathbb{R}^p$ are smooth state, input, and output functions, respectively. Moreover, we suppose that system (A.22) is controllable and observable [5].

A.1.2.1 Hammerstein-Wiener Models

Hammerstein-Wiener (HW) models [6], [7] represent an important class of dynamical systems for nonlinear identification, in which the input and output nonlinearities are static. HW models can be used in several areas, from the electromechanical system and radio frequency components modeling, to audio and speech processing, until to predictive control of industrial processes thanks to their simplicity in representation and implementation than heavy-duty nonlinear models, such as neural networks and Volterra models.

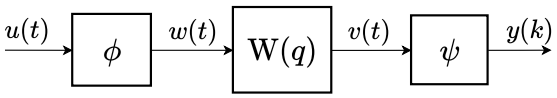

Figure A.4: Block diagram of the complete HW model.

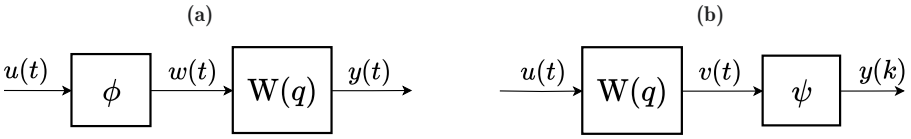

Figure A.5: Block diagram of the *Wiener* (a) and *Hammerstein* model.

The whole block diagram of HW models is represented in Fig. A.4. It is given by a series connection between two static nonlinear blocks and a dynamic linear block (in the middle). The input and output nonlinearities are modeled with purely static (i.e., algebraic) blocks. Furthermore, HW models can be used for modeling separately either just the input or output nonlinearity, respectively resulting in the so-called *Wiener* and *Hammerstein* model (see Fig. A.5). Such a transformation results in the following equivalent system for (A.22) [4]:

$$\dot{\boldsymbol{x}}(t) = \mathbf{A}\boldsymbol{x}(t) + \mathbf{B}\boldsymbol{w}(t) \quad \boldsymbol{v}(t) = \mathbf{C}\boldsymbol{x}(t) \tag{A.23a}$$

$$\boldsymbol{w}(t) = \phi\big(\boldsymbol{u}(t)\big) \quad \boldsymbol{y}(t) = \psi\big(\boldsymbol{v}(t)\big) \tag{A.23b}$$

where $\phi : \mathcal{U} \subseteq \mathbb{R}^m \to \mathcal{U}' \subseteq \mathbb{R}^{m'}$ and $\psi : \mathcal{X}' \subseteq \mathbb{R}^{p'} \to \mathcal{X} \subseteq \mathbb{R}^p$ are the nonlinear functions mapping the input and output signals, respectively. The matrices $\mathbf{A}, \mathbf{B}$, and $\mathbf{C}$ represent respectively the state, input, and output matrix of the linear dynamical subsystem, which can be rewritten in the typical Box-Jenkins model through the Z-Transform (see (A.4)-(A.5)) and the shift operator q (see (A.11)-(A.13b)) [7]:

$$v_j(k) = \sum_{i=1}^{m} \frac{\mathrm{B}_i(q)}{\mathrm{A}_i(q)}\, u_i(k) + e(k) \qquad \forall i = 1, \dots, m \quad \forall j = 1, \dots, p \tag{A.24}$$

where, $v_j(k)$ and $u_i(k)$ are the j-th and i-th linear input and output, respectively.

A.1.2.2 Sparse Identification of Nonlinear Dynamics

The idea of Sparse Identification of Nonlinear Dynamics (SINDy) [8] is to utilize algorithms based on sparse regression to learn the simplest model within an admissible library, required to describe the retrieved data.

For simplicity, suppose to simplify the nonlinear dynamics (A.22) with the following autonomous system [4], [9]:

$$\dot{\boldsymbol{x}}(t) = f\big(\boldsymbol{x}(t)\big) \tag{A.25}$$

where the states $\boldsymbol{x}(t) \in \mathcal{X} \subseteq \mathbb{R}^n$ can be measured. Moreover, assume that the function $f : \mathcal{X} \to \mathbb{R}^n$ can be expanded as a polynomial of degree $d \in \mathbb{N}$ given by a finite combination of multiple candidate basis functions $\theta_1\big(\boldsymbol{x}(t)\big), \theta_2\big(\boldsymbol{x}(t)\big), \dots, \theta_K\big(\boldsymbol{x}(t)\big)$ (consisting of monomials of degree less or equal to d). This results in:

$$\dot{\boldsymbol{x}}(t) = f\big(\boldsymbol{x}(t)\big) \approx \sum_{k=1}^{K} \xi_k \theta_k\big(\boldsymbol{x}(t)\big) \tag{A.26}$$

where $\theta_k \in \mathbb{R}^n$, $k = 1, \dots, K$. Thus, the sum in (A.26) can be compacted as follows

$$\sum_{k=1}^{K} \xi_k \theta_k\big(\boldsymbol{x}(t)\big) = \begin{pmatrix} \xi_{11} & \cdots & \xi_{K1} \\ \vdots & \ddots & \vdots \\ \xi_{1n} & \cdots & \xi_{Kn} \end{pmatrix} \begin{pmatrix} \theta_1\big(\boldsymbol{x}(t)\big) \\ \vdots \\ \theta_K\big(\boldsymbol{x}(t)\big) \end{pmatrix} := \boldsymbol{\Xi}^\top \boldsymbol{\Theta}\big(\boldsymbol{x}(t)\big) \tag{A.27}$$

where $\boldsymbol{\Theta} : \mathcal{X} \to \mathbb{R}^K$ and $\boldsymbol{\Xi} \in \mathbb{R}^{K \times n}$ are the library of all the candidate nonlinear functions and the matrix containing all the sparse vectors, respectively. Thus, by substituting (A.27) in (A.26), the nonlinear dynamics (A.25) can be approximated as follows:

$$\dot{\boldsymbol{x}}(t) \approx \boldsymbol{\Xi}^\top \boldsymbol{\Theta}\big(\boldsymbol{x}(t)\big). \tag{A.28}$$

Each parameter $\xi_k \in \boldsymbol{\Xi}$ can be obtained through a linear regression. For many systems, polynomial terms could be sufficient, despite the presence of nonlinearities in the dynamics. Once S observable data points $\big(\boldsymbol{x}(t)^{(s)}, \dot{\boldsymbol{x}}(t)^{(s)}\big)$ are available (where s represents the i-th retrieved data point $\forall i = 1, \ldots, S$), the parameters ξ_k can be obtained through several sparse optimization algorithms. To this aim, the fundamental assumption of SINDy is that the dynamics may be characterized by relatively few active terms, selected by the sparse element-vector ξ_k. Thus, the following objective function is usually involved for promoting the sparsity in the regression problem [8], [10]:

$$\min_{\boldsymbol{\Xi}} \quad \frac{1}{2}\Big\|\dot{\mathbf{X}} - \boldsymbol{\Xi}^\top \boldsymbol{\Theta}\big(\boldsymbol{x}(t)\big)\Big\|_2^2 + \lambda \|\boldsymbol{\Xi}\|_1, \tag{A.29}$$

usually solved with the Least Absolute Shrinkage and Selection Operator regression method. In the equation above, λ and $\dot{\mathbf{X}}$ are respectively the sparsity-promoting hyper-parameter and the state derivatives matrix for all the retrieved samples, defined as

$$\dot{\mathbf{X}} = \Big(\dot{\boldsymbol{x}}(t)^{(1)^\top} \quad \ldots \quad \dot{\boldsymbol{x}}(t)^{(D)^\top}\Big)^\top,$$

while a possible basis functions matrix $\boldsymbol{\Theta}\big(\boldsymbol{x}(t)\big)$ can be chosen as [8]

$$\boldsymbol{\Theta}\big(\boldsymbol{x}(t)\big) = \Big(1 \quad \boldsymbol{x}^\top \quad \big(x_1(t)^2, \ldots, x_n(t)^2\big)^\top \quad \sin\big(x_1(t)\big) \quad \cdots \quad \Big)^\top.$$

The basic procedure upper-described for SINDy regression method can be also extended to include both the state and control input. In that case, the candidate functions will be $\theta_k\big(\boldsymbol{x}(t), \boldsymbol{u}(t)\big)$. For a wide range of nonlinear systems, SINDy regression is highly useful in rapidly learning accurate models from few examples, which makes them considerably more data-efficient than other methods (e.g., neural networks) [9].

A.1.3 Validation Criteria

Regarding the validation procedure for system identification, several criteria are presented in the literature to aid in model selection, including the Akaike's Information Criterion (AIC), the Akaike's Final Prediction Error (FPE), and the Rissanen's Minimum Description Length (MDL) [11]–[13].

They provide quantitative measures to balance the model goodness of fitting and complexity, helping in preventing overfitting and ensuring that the selected model is good for new data. Let us define the size of the regressors (i.e., the number of parameters to be estimated) as d, and the dimension of the training dataset (i.e., the number of the overall retrieved samples) as N.

AIC is a measure of the relative quality of a statistical model that penalizes complex models to avoid overfitting, preventing the selection of overly complex models that may not generalize well to new data [11]. AIC is calculated using the formula:

$$\left(1 + \frac{2d}{\mathrm{N}}\right) \frac{1}{\mathrm{N}} \sum_{k=0}^{\mathrm{N}-1} e^2(k \mid \theta). \tag{A.30}$$

Lower values indicate a better balance between model fit and complexity. AIC is usually employed when the choice of the model is made on the same dataset used for the training.

FPE is another criterion used for model selection, particularly for the time series analysis. FPE is closely related to AIC, always based on the final prediction error. Also

FPE penalizes the number of parameters in the model to avoid overfitting [12], and it is calculated as follows:

$$\frac{\left(1+\frac{d}{\mathrm{N}}\right)}{\left(1-\frac{d}{\mathrm{N}}\right)}\frac{1}{\mathrm{N}}\sum_{k=0}^{\mathrm{N}-1}e^2(k\mid\theta). \tag{A.31}$$

As for FPE, lower values indicate a better balance between model fit and complexity. It is usually employed when the training and validation datasets are different.

MDL is a criterion for model selection based on the information theory. For the sake of balancing model fitting to data together with the complexity of the model, MDL aims to find the model that minimizes the length of the description of the data [13]. MDL is calculated using the formula:

$$\frac{d\ \ln(\mathrm{N})}{\mathrm{N}}+\frac{1}{\mathrm{N}}\sum_{k=0}^{\mathrm{N}-1}e^2(k\mid\theta). \tag{A.32}$$

Also in this case, lower values indicate a better balance between model fit and complexity. Moreover, like in the case of AIC, also MDL is generally employed when the choice of the model is based on the same training and validation dataset.

A.2 Adaptive Control

Adaptive control aims at achieving optimal performance in systems with uncertain or time-varying parameters, due to modeling inaccuracies and environmental changes. Specifically, adaptive systems allow to estimate such unknown parameters in real-time by properly adjusting the control law iteratively to refine the controller parameters, ensuring robustness [14].

The problem formulation typically addressed by adaptive control is as follows [14]:

$$y(t)=\boldsymbol{\theta}^\top\boldsymbol{\phi}(t)+d(t) \tag{A.33}$$

where $y(t)$ is the system output, $\boldsymbol{\phi}(t)$ and $\boldsymbol{\theta}$ are respectively the known regression terms and unknown parameters vectors, while $d(t)$ represents the disturbances or unmodelled dynamics. The control objective is to design a stable control law $u(t)$ such that:

$$\lim_{t\to\infty}|y(t)-y_m(t)|=0$$

where $y_m(t)$ is the output of a desired reference model. Notwithstanding, due to the inherent nonlinearity of adaptation mechanisms, ensuring stability and convergence is non-trivial in such a problem. To this aim, most of adaptive systems use expressions as follows as candidate Lyapunov functions:

$$V(\tilde{\boldsymbol{\theta}})=\tilde{\boldsymbol{\theta}}^\top\boldsymbol{\Gamma}^{-1}\tilde{\boldsymbol{\theta}}$$

where $\boldsymbol{\Gamma}$ is the adaptation gain matrix.

A.2.1 Model Reference Adaptive Control

Model Reference Adaptive Control (MRAC) represents a fundamental methodology within adaptive control. The MRAC framework for the SISO system in (A.33) is described in Fig. A.6, depicting a reference model that defines the target trajectory $y_m(t)$ for the plant output $y_p(t)$ to track [15]. The tracking error, denoted as $e_1(t)=y_p(t)-y_m(t)$, quantifies the divergence of the plant output from its intended path. The closed-loop plant comprises a conventional feedback control strategy that incorporates both the plant and a controller $C(\boldsymbol{\theta})$, along with an adjustment mechanism that facilitates the real-time computation of the estimated controller parameters $u(t)$. MRAC strategies are categorized into *direct* or *indirect* adaptive laws. In the direct MRAC approach, the controller's parameter vector

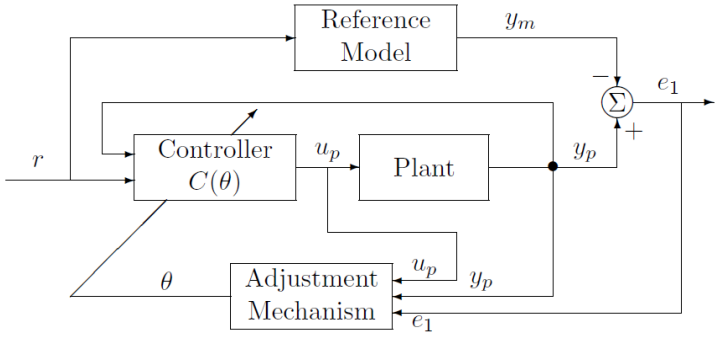

Figure A.6: General structure of MRAC scheme [15].

$\boldsymbol{\theta}$ is directly modified through an adaptive law. Conversely, in an indirect setting, $\boldsymbol{\theta}$ is determined at each time t by resolving an algebraic equation connecting $\boldsymbol{\theta}$ with the online-determined plant parameter estimates.

The problem formulation typically addressed by MRAC schemes is as follows:

$$\dot{x}(t) = ax(t) + bu(t) \tag{A.34}$$

where a and b are the unknown parameters (with the sign of b known). The control objective is to choose an appropriate control law $u(t)$ such that all signals in the closed-loop plant are bounded and the current state $x(t)$ tracks the state $x_m(t)$ of the reference model given by

$$\dot{x}_m(t) = a_m x_m(t) + b_m r(t)$$

where $r(t)$ is the bounded piecewise continuous reference input, $a_m > 0$ and b_m are known, while $x_m(t)$ is measured at each time t.

The control law $u(t)$ in (A.34) is chosen so that the closed-loop plant transfer function from the input $r(t)$ to output $x(t)$ is equal to that of the reference model, i.e.:

$$u(t) = -k^* x(t) + l^* r(t) \tag{A.35}$$

where k^* and l^* are the *true* adaptation parameters that should satisfy the following equations:

$$l^* = \frac{b_m}{b}, \qquad k^* = \frac{a_m + a}{b},$$

which hold under the assumption that the plant (A.34) is controllable (i.e., $b \neq 0$). Since the plant parameters a and b are unknown, (A.35) cannot be practically implemented. Conversely, the control law used in the MRAC scheme is as follows:

$$u(t) = -k(t)x(t) + l(t)r(t) \tag{A.36}$$

where $k(t)$ and $l(t)$ are respectively the estimate of the true control parameters k^* and l^* at time t. Therefore, the adaptive control law aims at finding $k(t)$ and $l(t)$ online through the following adaptation mechanism:

$$\dot{k} = \gamma_1 e_1 x \ \mathrm{sgn}(b), \qquad \dot{l} = \gamma_2 e_1 r \ \mathrm{sgn}(b) \tag{A.37}$$

where $\gamma_1, \gamma_2 > 0$ are adaptive gains, while $\mathrm{sgn}(\cdot)$ represents the *sign* function. The adaptive law (A.36)-(A.37) ensures the derivative of the candidate Lyapunov function

$$V\left(e_1, \widetilde{k}, \widetilde{l}\right) = \frac{e_1^2}{2} + \frac{\widetilde{k}^2}{2\gamma_1}|b| + \frac{\widetilde{l}^2}{2\gamma_2}|b|,$$

i.e.,

$$\dot{V} = -a_m e_1^2 - b\widetilde{k}e_1 x + b\widetilde{l}e_1 r + \frac{|b|\widetilde{k}}{\gamma_1}\gamma_1 e_1 x \mathrm{sgn}(b) + \frac{|b|\widetilde{l}}{\gamma_2}\gamma_2 e_1 r \mathrm{sgn}(b),$$

to boil down to

$$\dot{V} = -a_m e_1^2.$$

Here, $\widetilde{k} = k - k^*$ and $\widetilde{l} = l - l^*$ represent the parameters errors for k^* and l^*, respectively.

A.3 Single-Phase DC/AC Inverter Model

This section aims at illustrating the approach typically used for modelling the single-phase DC/AC inverter. Further details can be found in [16].

If we indicate the constant supply voltage of a half-bridge electronic circuit as $\mathrm{V_s}$, the pulse width of the digital signals commanding the transistors can be modulated in order to deny simultaneous switching, thus preventing a short circuit. Hence, the load voltage drop can be written as follows:

$$v_{AO}(t) = \begin{cases} \frac{\mathrm{V_s}}{2} & \text{for } t_k \leq t \leq t_k + d_k^+ \mathrm{T} \\ -\frac{\mathrm{V_s}}{2} & \text{for } t_k + d_k^+ \mathrm{T} < t < t_k + \mathrm{T} \end{cases} \tag{A.38}$$

where d_k^+ denotes the corresponding duty-cycle at discrete time t_k. The PWM-based formulation approximates the voltage drop signal with its mean value over each period t_k as follows:

$$v_{AO}(t) \approx \overline{v}_{AO,k} = \frac{1}{\mathrm{T}} \int_{t_k}^{t_k+\mathrm{T}} v_{AO}(t)\ dt, \quad t_k \leq t \leq t_k + \mathrm{T}. \tag{A.39}$$

Thus, substituting (A.38) in (A.39), the mean voltage drop value through the electric load can be rewritten as:

$$\overline{v}_{AO,k} = \frac{1}{\mathrm{T}} \left(\int_{t_k}^{t_k+d_k\mathrm{T}} \frac{V_s}{2}\ dt - \int_{t_k+d_k\mathrm{T}}^{t_k+\mathrm{T}} \frac{V_s}{2}\ dt \right) = \frac{V_s}{2}\big(2d_k - 1\big). \tag{A.40}$$

Therefore, by employing Kirchhoff's voltage law in the half-bridge circuit, we can derive the following dynamical model for the single-phase DC/AC inverter:

$$\dot{i_{AO}}(t) = -\frac{\mathrm{R}}{\mathrm{L}} i_{AO}(t) + \frac{1}{\mathrm{L}} v_{AO}(t) \tag{A.41}$$

where L, R, and $i_{AO}(t)$ are the electric inductance, resistance, and current of the load, respectively. Then, discretizing (A.41) by means of the Euler method, the latter can be expressed in the following equivalent discrete-time form:

$$i_{AO,k+1} = a\ i_{AO,k} + b\ v_{AO,k}. \tag{A.42}$$

Substituting (A.40) in (A.42), the following discrete-time dynamical model is obtained for the single-phase DC/AC inverter:

$$i_{AO,k+1} = a\ i_{AO,k} + \mathrm{V_s}\ b\ d_k^+ - \frac{\mathrm{V_s}}{2}$$

with:

$$a = e^{-\frac{\mathrm{L}}{\mathrm{R}}\mathrm{T}}, \quad b = \int_{t_k}^{t_k+\mathrm{T}} \left(\tfrac{1}{\mathrm{L}} e^{-\frac{\mathrm{L}}{\mathrm{R}}(t_k+\mathrm{T}-\tau)} \right) d\tau$$

where $i_{AO,k}$ and d_k are the discrete-time electrical current passing through the load (as state variable and output) and the duty-cycle (as control input), respectively. The quantity $-\frac{\mathrm{V_s}}{2}$, instead, acts as a constant external disturbance input.

References

[1] Berberich, J., Iannelli, A., Padoan, A., Coulson, J., Dörfler, F., and Allgöwer, F., "A quantitative and constructive proof of willems' fundamental lemma and its implications," in *2023 American Control Conference (ACC)*, IEEE, 2023, pp. 4155–4160.

[2] Ljung, L., "System identification," in *Signal analysis and prediction*, Springer, 1998, pp. 163–173.

[3] Box, G. E., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M., *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[4] Tang, W. and Daoutidis, P., "Data-driven control: Overview and perspectives," in *2022 American Control Conf. (ACC)*, IEEE, 2022, pp. 1048–1064.

[5] Isidori, A., *Nonlinear control systems: an introduction*. Springer, 1985.

[6] Boutayeb, H. and Darouach, M., "Recursive identification method for miso wiener-hammerstein model," *IEEE transactions on automatic control*, vol. 40, no. 2, pp. 287–291, 1995.

[7] Zhu, Y., "Estimation of an n–l–n hammerstein–wiener model," *Automatica*, vol. 38, no. 9, pp. 1607–1614, 2002.

[8] Brunton, S. L., Proctor, J. L., and Kutz, J. N., "Sparse identification of nonlinear dynamics with control (sindyc)," *IFAC-PapersOnLine*, vol. 49, no. 18, pp. 710–715, 2016.

[9] Soudbakhsh, D., Annaswamy, A. M., Wang, Y., *et al.*, "Data-driven control: Theory and applications," in *2023 American Control Conf. (ACC)*, IEEE, 2023, pp. 1922–1939.

[10] Fasel, U., Kaiser, E., Kutz, J. N., Brunton, B. W., and Brunton, S. L., "Sindy with control: A tutorial," in *2021 60th IEEE Conf. on Decision and Control (CDC)*, IEEE, 2021, pp. 16–21.

[11] Akaike, H., "A new look at the statistical model identification," *IEEE transactions on automatic control*, vol. 19, no. 6, pp. 716–723, 1974.

[12] Akaike, H., "A bayesian extension of the minimum aic procedure of autoregressive model fitting," *Biometrika*, vol. 66, no. 2, pp. 237–242, 1979.

[13] Rissanen, J., "Modeling by shortest data description," *Automatica*, vol. 14, no. 5, pp. 465–471, 1978.

[14] Narendra, K. S. and Annaswamy, A. M., *Stable adaptive systems*. Courier Corporation, 2012.

[15] Ioannou, P. A. and Sun, J., *Robust adaptive control*. PTR Prentice-Hall Upper Saddle River, NJ, 1996, vol. 1.

[16] Lee, D.-C. and Kim, Y.-S., "Control of single-phase-to-three-phase ac/dc/ac pwm converters for induction motor drives," *IEEE transactions on industrial electronics*, vol. 54, no. 2, pp. 797–804, 2007.