FEDERAL UNIVERSITY OF AMAZONAS
FACULTY OF TECHNOLOGY
GRADUATE PROGRAM IN ELECTRICAL ENGINEERING

Active Fault-Tolerant Control Strategy Based on Moving Horizon Estimation and Model Predictive Control

ARLLEM DE OLIVEIRA FARIAS

MANAUS-AM
2025

PPGEE
Programa de Pós-Graduação em
Engenharia Elétrica - UFAM

ARLLEM DE OLIVEIRA FARIAS

Active Fault-Tolerant Control Strategy Based on Moving Horizon Estimation and Model Predictive Control

Master's thesis submitted to the Graduate Program in Electrical Engineering at the Federal University of Amazonas, in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering, with a concentration in Systems Control and Automation.

Supervisor: Prof. Dr. Iury Valente de Bessa
Co-Supervisor: Prof. Dr. Renan Landau Paiva de Medeiros

MANAUS-AM
2025

Ficha Catalográfica

Elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

F224a Farias, Arllem de Oliveira

Active Fault-Tolerant Control Strategy Based on Moving Horizon Estimation and Model Predictive Control / Arllem de Oliveira Farias. - 2025.

59 f. : il., color. ; 31 cm.

Orientador(a): Iury Valente de Bessa.

Coorientador(a): Renan Landau Paiva de Medeiros.

Dissertação (mestrado) - Universidade Federal do Amazonas, Programa de Pós-Graduação em Engenharia Elétrica, Manaus, 2025.

1. Fault-tolerant control. 2. Model predictive control. 3. Moving horizon estimation. 4. Fault accommodation. 5. Constrained control. I. Bessa, Iury Valente de. II. Medeiros, Renan Landau Paiva de. III. Universidade Federal do Amazonas. Programa de Pós-Graduação em Engenharia Elétrica. IV. Título

Ministério da Educação
Universidade Federal do Amazonas
Coordenação do Programa de Pós-Graduação em Engenharia Elétrica

FOLHA DE APROVAÇÃO

Poder Executivo Ministério da Educação
Universidade Federal do Amazonas
Faculdade de Tecnologia
Programa de Pós-graduação em Engenharia Elétrica

Pós-Graduação em Engenharia Elétrica. Av. General Rodrigo Octávio Jordão Ramos, nº 3.000 - Campus Universitário, Setor Norte - Coroado, Pavilhão do CETELI. Fone/Fax (92) 99271-8954 Ramal:2607. E-mail: ppgee@ufam.edu.br

ARLLEM DE OLIVEIRA FARIAS

ACTIVE FAULT-TOLERANT CONTROL BASED ON MOVING HORIZON ESTIMATION AND MODEL PREDICTIVE CONTROL

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da
Universidade Federal do Amazonas, como
requisito parcial para obtenção do título de Mestre
em Engenharia Elétrica na área de concentração
Controle e Automação de Sistemas.

Aprovada em 29 de agosto de 2025.

BANCA EXAMINADORA
Prof. Dr. Iury Valente de Bessa- Presidente
Profa. Dra. Rosileide de Oliveira Lopes - Membro Titular 1 - Externo
Profa. Dra. Márcia Luciana da Costa Peixoto - Membro Titular 2 - Externo

Manaus, 14 de agosto de 2025.

Documento assinado eletronicamente por **Márcia Luciana da Costa Peixoto**, **Usuário Externo**, em 01/09/2025, às 10:01, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Rosileide de Oliveira Lopes**, **Usuário Externo**, em 01/09/2025, às 13:32, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Iury Valente de Bessa**, **Professor do Magistério Superior**, em 03/09/2025, às 19:18, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015 .

A autenticidade deste documento pode ser conferida no site https://sei.ufam.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **2744313** e o código CRC **88498AF8**.

Av. General Rodrigo Octávio Jordão Ramos, nº 3.000 - Bairro Coroado Campus Universitário, Setor Norte - Telefone: 99271-8954
CEP 69080-900 Manaus/AM - Pavilhão do CETELI. E-mail: ppgee@ufam.edu.br

Referência: Processo nº 23105.032998/2025-03 SEI nº 2744313

Dedico este trabalho
aos meus pais, Mauro e Edy, que sempre acreditaram em mim,
à minha amada esposa, Rafaela, pelo amor e incentivo incondicionais,
e a todas as pessoas que me motivaram a nunca desistir dos meus objetivos.

Agradecimentos

A Deus, pelo dom da vida, pelo Seu amor, por este momento e por todas as conquistas e vitórias alcançadas até aqui.

Aos meus pais e a toda a minha família, por todo o amor e apoio fornecidos durante toda a minha vida.

À minha esposa, pelo seu amor, pela compreensão nas horas em que estive ausente, pelas palavras de carinho e confiança que me ajudaram a prosseguir até aqui.

Ao meu orientador, Prof. Dr. Iury Valente de Bessa, por todo o conhecimento compartilhado, pelas palavras de incentivo, por se mostrar sempre presente e disposto a ajudar e, principalmente, por acreditar em mim.

A todos os professores que contribuíram com este trabalho, compartilhando seus conhecimentos em salas de aula e laboratórios.

À Universidade Federal do Amazonas (UFAM), por todos os recursos e serviços fornecidos ao longo dessa trajetória.

A todos que, de alguma forma, contribuíram para a realização deste trabalho, o meu sincero muito obrigado.

“… Nunca deixe que lhe digam que não vale a pena acreditar no sonho que se tem,
ou que seus planos nunca vão dar certo, ou que você nunca vai ser alguém.
… Quem acredita, sempre alcança.”

— Renato Russo e Flávio Venturini, *Mais Uma Vez*

Resumo

Este trabalho aborda o projeto e a avaliação de uma estrutura integrada de Controle Ativo Tolerante a Falhas (AFTC), que combina Controle Preditivo por Modelo (MPC) e Estimativa de Horizonte Móvel (MHE) para acomodação de falhas em tempo real em sistemas multivariáveis com restrições. A metodologia proposta é validada em um modelo linearizado do sistema de três tanques, tanto sob condições nominais quanto na presença de falhas nos atuadores e na planta.

A formulação do MPC incorpora restrições poliédricas de estado e entrada, um conjunto terminal elipsoidal e sua aproximação poliédrica interna, garantindo a factibilidade e a estabilidade recursivas. Um estimador de parâmetros baseado em MHE amplia o modelo do sistema para, de forma simultânea e online, realizar a estimativa de estados e de resíduos de falhas, permitindo a reconfiguração dinâmica do sistema em malha fechada.

Para o cenário nominal (livre de falhas), os resultados das simulações demonstram que a estrutura proposta alcança um rastreamento preciso do ponto de ajuste, com erro desprezível em regime permanente, respeitando todas as restrições. Em situações de falhas abruptas do atuador, o método restaura o desempenho próximo ao obtido com conhecimento perfeito das falhas, apresentando apenas pequena degradação devido ao atraso na estimativa. Já em casos de falhas graves na planta, envolvendo não linearidades significativas, como bloqueios de tubulações e vazamentos em tanques, a abordagem proposta proporciona apenas recuperação parcial, com desempenho superior ao da configuração nominal, embora limitado pela capacidade de convergência do estimador.

O estudo conclui que a integração de MPC e MHE em uma estrutura AFTC unificada aumenta a resiliência do sistema a falhas, mantendo a satisfação das restrições. A abordagem mostra-se particularmente adequada para aplicações lineares e estabelece uma base para futuras extensões, incluindo estimativas robustas e não lineares, estratégias avançadas de controle preditivo e implementação em hardware em tempo real.

Palavras-chave: Controle tolerante a falhas, Controle preditivo baseado em modelo, Estimação por horizonte móvel, Compensação de falhas, Controle com restrições, Sistema de três tanques.

Abstract

This work addresses the design and evaluation of an integrated Active Fault-Tolerant Control (AFTC) framework that combines Model Predictive Control (MPC) and Moving Horizon Estimation (MHE) to achieve real-time fault accommodation in constrained multivariable systems. The proposed methodology is validated on a linearized three-tank system model, both under nominal conditions and in the presence of actuator and plant faults.

The MPC formulation incorporates polyhedral state and input constraints, an ellipsoidal terminal set, and its inner polyhedral approximation to ensure recursive feasibility and stability. A parameter estimator based on MHE augments the system model to simultaneously estimate states and fault residuals online, enabling the dynamic reconfiguration of the closed-loop system.

For the nominal (fault-free) scenario, the simulation results demonstrate that the proposed AFTC framework achieves accurate setpoint tracking with negligible steady-state error, while respecting all constraints. In situations of abrupt actuator fault, the method restores performance close to that obtained with perfect fault knowledge, presenting only minor degradation due to estimation delay. In cases of severe plant faults involving significant nonlinearities, such as pipe blockages and tank leakages, the proposed approach provides only partial recovery, with superior performance to the nominal-model configuration, although limited by the convergence capacity of the estimator.

The study concludes that integrating MPC and MHE into a unified AFTC framework enhances system resilience to faults while ensuring constraint satisfaction. The proposed approach is particularly well-suited for linear applications and provides a foundation for future extensions, including robust and nonlinear estimations, advanced predictive control strategies, and real-time hardware implementation.

Keywords: Fault-tolerant control, Model predictive control, Moving horizon estimation, Fault accommodation, Constrained control, Three-tank system.

List of Figures

2.1 Prediction and control horizon scheme in MPC. 6

2.2 Ellipsoidal domain of attraction $\mathcal{D}_N(\Omega) \subset \mathcal{X} \subset \mathbb{R}^2$. 10

2.3 Estimation horizon scheme in MHE. 11

3.1 The three-tank system. All tanks and pipes are identical; the pumps are independent and can be directly controlled. All valves can operate as actuators, and all levels and flows are measurable. 17

3.2 Configuration of the three-tank system used as the case study. The inlet flows to the lateral tanks (T_1 and T_2) are regulated by valves K_{P_1} and K_{P_2}. The valves K_{13}, K_{23}, and K_3 are assumed to be permanently open (i.e., $K_{13} = K_{23} = K_3 = 1$), ensuring fixed hydraulic connections between the tanks and the reservoir, and therefore do not operate in flow regulation. All other valves and pipes not depicted are considered inactive or neglected for the purposes of this study. 19

4.1 Representation of the MHE connection structure. 35

4.2 Proposed closed-loop architecture integrating the fault residual estimator and the control readjustment model to perform fault accommodation. . . . 38

5.1 Polyhedral sets of admissible state and input deviations relative to $(x_{\mathrm{op}}, u_{\mathrm{op}})$. 44

5.2 Polyhedral representation of the terminal set and the domain of attraction. 44

5.3 Setpoint tracking of the controlled variable Q_3 (a), control inputs (b), system states (c), and fault residual estimation (d) under nominal conditions. 45

5.4 Closed-loop results using the $\underline{\text{nominal model}}$ $\big(A_0,\, B_0\big)$ under an abrupt actuator fault with a maximum magnitude of $f_1 = 0.8$. 48

5.5 Closed-loop results using the $\underline{\text{exact model}}$ $\big(A(f_k),\, B(f_k)\big)$ under an abrupt actuator fault with a maximum magnitude of $f_1 = 0.8$. 48

5.6 Closed-loop results using the $\underline{\text{estimated model}}$ $\big(\hat{A}(f_k),\, \hat{B}(f_k)\big)$ under an abrupt actuator fault with a maximum magnitude of $f_1 = 0.8$. 49

5.7 Closed-loop results using the $\underline{\text{nominal model}}$ $\big(A_0,\, B_0\big)$ under an incipient plant fault with a maximum magnitude of $f_6 = 0.9$. 50

5.8 Closed-loop results using the $\underline{\text{exact model}}$ $\big(A(f_k),\, B(f_k)\big)$ under an incipient plant fault with a maximum magnitude of $f_6 = 0.9$. 50

5.9 Closed-loop results using the estimated model $\big(\hat{A}(f_k),\, \hat{B}(f_k)\big)$ under an incipient plant fault with a maximum magnitude of $f_6 = 0.9$. 51
5.10 Closed-loop results using the nominal model $\big(A_0,\, B_0\big)$ under an incipient plant fault with a maximum magnitude of $f_9 = 1$. 51
5.11 Closed-loop results using the exact model $\big(A(f_k),\, B(f_k)\big)$ under an incipient plant fault with a maximum magnitude of $f_9 = 1$. 52
5.12 Closed-loop results using the estimated model $\big(\hat{A}(f_k),\, \hat{B}(f_k)\big)$ under an incipient plant fault with a maximum magnitude of $f_9 = 1$. 52

List of Tables

3.1 Transmission flow scenarios for the three-tank system. 18

3.2 List of faults for the default case study configuration. 20

4.1 Physical parameters of the three-tank system. 39

5.1 Closed-loop performance metrics under an abrupt actuator fault (loss of effectiveness) initiated at around $t = 150$ s with a maximum magnitude of $f_1 = 0.8$. 48

5.2 Closed-loop performance metrics under an incipient plant fault (pipe blockage) initiated at approximately $t = 85$ s with a maximum magnitude of $f_6 = 0.9$. 49

5.3 Closed-loop performance metrics under an incipient plant fault (tank leakage) initiated at approximately $t = 115$ s with a maximum magnitude of $f_9 = 1$. 51

List of Abbreviations

AFTC Active Fault-Tolerant Control.

ESO Extended State Observers.

FD Fault Diagnosis.

FE Fault Estimation.

FRE Fault Residual Estimator.

FTC Fault-Tolerant Control.

FTMPC Fault-Tolerant MPC.

KF Kalman Filter.

LMI Linear Matrix Inequality.

LTI Linear Time-Invariant.

MHE Moving Horizon Estimation.

MIMO Multiple-Input Multiple-Output.

MPC Model Predictive Control.

PFTC Passive Fault-Tolerant Control.

RMSE Root Mean Square Error.

SMO Sliding Mode Observer.

SSE Steady-State Error.

TS Takagi-Sugeno.

UAVs Unmanned Aerial Vehicles.

UIO Unknown Input Observer.

ZOH Zero-Order Hold.

Contents

1 Introduction **1**

1.1 Research Objectives . . . 3

1.1.1 General Objective . . . 3

1.1.2 Specific Objectives . . . 3

1.2 Thesis Outline . . . 4

2 Background **5**

2.1 Model Predictive Control . . . 5

2.1.1 General Formulation . . . 6

2.1.2 Constraints . . . 7

2.1.3 Stability and Feasibility . . . 8

2.1.4 Domain of Attraction . . . 9

2.2 Moving Horizon Estimation . . . 10

2.2.1 General Formulation . . . 11

2.2.2 Cost Function and Constraints . . . 11

2.2.3 Horizon Estimation and the MHE/MPC Duality . . . 13

2.3 Related Work . . . 13

3 The Three-Tank System **16**

3.1 Plant Description . . . 16

3.2 Nonlinear Model . . . 18

3.3 Case Study Definition . . . 19

3.3.1 Fault Model . . . 20

3.3.2 Linear Fault Model . . . 21

3.3.3 Discrete-Time Model . . . 22

4 Methodology **24**

4.1 MPC Implementation . . . 24

4.1.1 Computation of the Terminal Set . . . 26

4.1.2 Definition of State and Input Constraints . . . 28

4.1.3 Formulation of the Terminal Constraint . . . 28

4.1.4 Approximation of the Domain of Attraction . . . 32

4.1.5 Setpoint Tracking Strategy 33
4.2 MHE Implementation 35
4.3 Fault Residual Estimation 36
4.4 Control Reconfiguration 37
4.5 Simulation Framework 38
4.5.1 Simulation Environment 38
4.5.2 System Configuration and Physical Parameters 39
4.5.3 MPC and MHE Setup 39

5 Results 43
5.1 Admissible and Terminal Sets Visualization 43
5.2 Nominal Operation Scenario 45
5.3 Comparative Evaluation Methodology for Fault Scenarios 46
5.4 Actuator Fault: Loss of Effectiveness in K_{P_1} 47
5.5 Plant Fault: Blocking in flow Q_{13} 48
5.6 Plant Fault: Leakage in Tank T_2 50

6 Conclusions 53
6.1 Future Work 54

Bibliography 55

Chapter 1

Introduction

The complexity and cost of modern systems have steadily increased with technological advancements. Simultaneously, the demand for safety, reliability, and efficiency has become fundamental in the design and control of such systems. Due to the natural wear of components, poor maintenance, misuse of the equipment, or malfunction of critical components, such as sensors and actuators, the presence of faults is an inevitable reality.

In safety-critical sectors—such as aerospace, automotive, and healthcare—the occurrence of faults can pose significant risks and cause irreversible damage to the system and, consequently, to the people involved. In this context, Fault-Tolerant Control (FTC) emerges as a fundamental strategy for a reliable implementation of complex dynamic systems. In addition to its ability to detect, diagnose, and mitigate the adverse effects of faults, FTC plays a vital role in ensuring safe and efficient system operation, maintaining operational integrity, and preserving closed-loop stability and desired performance within acceptable limits.

FTC techniques are commonly categorized into two classes: passive and active. In Passive Fault-Tolerant Control (PFTC), the controller is designed to be robust against fault-induced variations in system parameters and can handle a limited set of faults without modifying the control law. In contrast, Active Fault-Tolerant Control (AFTC) adapts the control law dynamically in response to faults. Typically, an AFTC approach comprises two sequential steps: fault diagnosis, which detects the fault, isolates the faulty components, and identifies a model of the faulty system; and control readjustment, which determines a reconfigured controller to replace the nominal one [1].

The control readjustment step can be implemented in three distinct ways: through **fault accommodation** [1], when the sets of manipulated and measured signals remain unchanged, and the adjustment is limited to the controller dynamics; through **control reconfiguration** [2], when changes are made to the controller dynamics, the closed-loop structure, and the reference signal; or through **fault hiding** [3], where a reconfiguration block is introduced into the control loop to mask the effects of sensor or actuator faults from the nominal controller [4].

AFTC frameworks are designed to monitor and respond online to different kinds of faults that might occur in dynamic systems, aiming to preserve the system performance—with minimal degradation—and closed-loop stability. The model that adequately represents the faulty system is selected based on the outcome of the Fault Diagnosis (FD) module. An alternative way to address FD is through Fault Estimation (FE) techniques. If a fault signal is estimated—revealing its occurrence (detection)—for each possible fault scenario—identifying its location (isolation)—and its magnitude reflects the fault severity and dynamic behavior (identification), then a complete fault diagnosis is achieved. On the other hand, if only a fault indicator is estimated, additional steps are required to reconstruct the fault behavior and perform fault diagnosis [5]. Once reconstructed, the estimated fault signals can be directly incorporated into the control law to mitigate their effects on the system [6].

Several FE techniques have been proposed in the literature to improve the efficiency and accuracy of fault detection and compensation. Unknown Input Observer (UIO) allows fault estimation in the presence of unknown disturbances, isolating faults while minimizing the influence of external effects, making it robust against uncertainties [7, 8]. Sliding Mode Observer (SMO) exhibits similar robustness by employing a discontinuous control law to ensure fast and accurate fault detection [9, 10, 11]. For nonlinear systems, Extended State Observers (ESO) simultaneously estimates both system states and faults, allowing real-time fault compensation [12, 13]. Kalman Filter (KF) techniques—such as the extended and unscented KF—provide optimal state and fault estimates based on probabilistic models, updating the estimations at each sampling instant [14, 15]. Moving Horizon Estimation (MHE) formulates an optimization problem over a sliding window to estimate states and faults, making it particularly suitable for systems with constraints, since these can be explicitly incorporated into the estimation problem as equality or inequality conditions [16, 17]. Each technique presents distinct advantages: UIO and SMO excel in dealing with model uncertainties; MHE is effective in managing constraints and nonlinearities; and KF-based methods offer broad applicability across diverse domains. The selection of a suitable method depends on the system dynamics and fault characteristics. In many cases, hybrid approaches can provide the most effective solution for maintaining the system performance with minimal degradation.

In AFTC design, it is often assumed that the FD module provides ideal responses, i.e., the correct fault signal is always available for consultation at any time, as considered in [18]. This assumption allows the designer to focus specifically on the control readjustment problem. However, integrating FD techniques with control readjustment strategies in real-world applications presents several challenges. One such challenge is the occurrence of false alarms, often caused by measurement noise or model uncertainties. To address this, the authors in [19] proposed a learning-based switching function. Another critical issue is missed detection, where low-magnitude faults go unnoticed, preventing the activation of control readjustment and potentially leading to performance degradation or even loss of stability. To mitigate this, the nominal controller should be robust to variations introduced by such faults [20]. A further concern is the response time of the FD module. If the fault information is not delivered promptly, the delay in control adaptation can also lead to performance degradation or instability, as demonstrated in [21]. These challenges are fundamentally tied to the reliability of the FD module, i.e., its ability to deliver accurate and timely responses when a fault occurs, as discussed in [22].

Numerous methodologies for implementing FTC have been proposed in the literature [23, 4, 24]. Among these, approaches capable of handling constraints are particularly attractive for practical applications, since real-world systems are inherently subject to physical limitations in usage and capacity. In this context, and with the continuous growth of computational power, Model Predictive Control (MPC) has attracted significant attention over the past decades [25, 26, 27]. This is primarily attributed to its key advantage: the ability to solve finite-horizon optimal control problems subject to strict equality and inequality constraints on control inputs and system states at each sampling instant [28]. A conceptually related method is MHE, whose integration with MPC has also received increasing interest across diverse research domains [29, 30, 31], since both are formulated as online optimization problems with explicit constraints [32]. Given its shared optimization structure with MPC, MHE is often referred to as its dual problem [33].

This work proposes an active fault-tolerant control framework that combines MHE and MPC for a Linear Time-Invariant (LTI) Multiple-Input Multiple-Output (MIMO) system subject to state and input constraints. The core idea is to adapt the MHE optimization problem to simultaneously estimate the system states and the fault residual signals, and then use this information—together with past control inputs—to estimate the faulty system parameters required by the MPC to perform fault accommodation in actuators and plant malfunctions, thereby mitigating their effects in closed-loop operation. This integrated strategy is designed to maintain closed-loop stability and ensure setpoint tracking with minimal performance degradation, even in the presence of faults.

1.1 Research Objectives

1.1.1 General Objective

To develop and implement an active fault-tolerant control framework that integrates moving horizon estimation and model predictive control with input and state constraints to accommodate actuator faults and plant malfunctions in an LTI MIMO system.

1.1.2 Specific Objectives

1. To design and implement a model predictive control scheme for setpoint tracking.

2. To implement a moving horizon estimation approach for state estimation.

3. To integrate model predictive control and moving horizon estimation into a unified framework.
4. To formulate a fault residual estimator based on moving horizon estimation.
5. To design an active fault-tolerant control strategy using model predictive control.
6. To evaluate the proposed methodology through numerical simulations.

1.2 Thesis Outline

The remainder of this thesis is structured as follows:

- Chapter 2 provides theoretical background on MPC and MHE, including constraints, stability guarantees, and duality. This chapter also presents relevant related works that combine MPC and MHE to build an AFTC framework.
- Chapter 3 describes the benchmark three-tank system and its mathematical models. A default case study is defined, and a linear faulty model is obtained to describe its behavior in the presence of faults in the actuator, plant, and sensors.
- Chapter 4 presents the proposed methodology, including control and estimation strategies, fault residual estimation, and model reconfiguration.
- Chapter 5 shows the simulation results obtained under different fault scenarios.
- Chapter 6 concludes the work and outlines possible future research directions.

Chapter 2

Background

This chapter presents the theoretical foundations and core concepts underlying the control and estimation strategies employed in this work. Section 2.1 summarizes the key elements of Model Predictive Control (MPC) over a finite horizon with state and input constraints, while Section 2.2 outlines the formulation and implementation of Moving Horizon Estimation (MHE). Finally, Section 2.3 reviews relevant studies that combine MPC and MHE to implement Active Fault-Tolerant Control (AFTC) frameworks. These works serve as a basis for the development of the methodology proposed in this study, which takes advantage of the synergy between estimation and control to ensure system performance and robustness in the presence of faults.

2.1 Model Predictive Control

Model Predictive Control (MPC) is a control strategy widely applied to multivariable dynamic systems subject to operational constraints. It employs a dynamic model of the system to predict its behavior over a prediction horizon of N steps from a given current state. Based on these predictions, MPC computes the optimal control input that steers the system states toward the desired targets while ensuring that the predictions closely match the most likely actual states of the system.

Figure 2.1 illustrates the core concept of an MPC strategy. At the current time step k, the reference trajectory, the measured or estimated state of the system, and the previous control input are available. Based on this data, the system model generates a sequence of state predictions $\{\bar{x}_i\}_{i=k+1}^{k+N}$ over the next N time steps. These predictions are within a set known as the prediction horizon. The corresponding sequence of control inputs $\{\bar{u}_i\}_{i=k}^{k+N-1}$ is within the control horizon, which corresponds to the maximum number of time steps ahead that the controller can act effectively on the system. At each sampling instant, an optimization problem is solved to compute the sequence of control inputs that minimizes the deviation between the sequence of state predictions and the reference trajectory. This process involves minimizing a cost function subject to constraints on both the state and control input sequences.

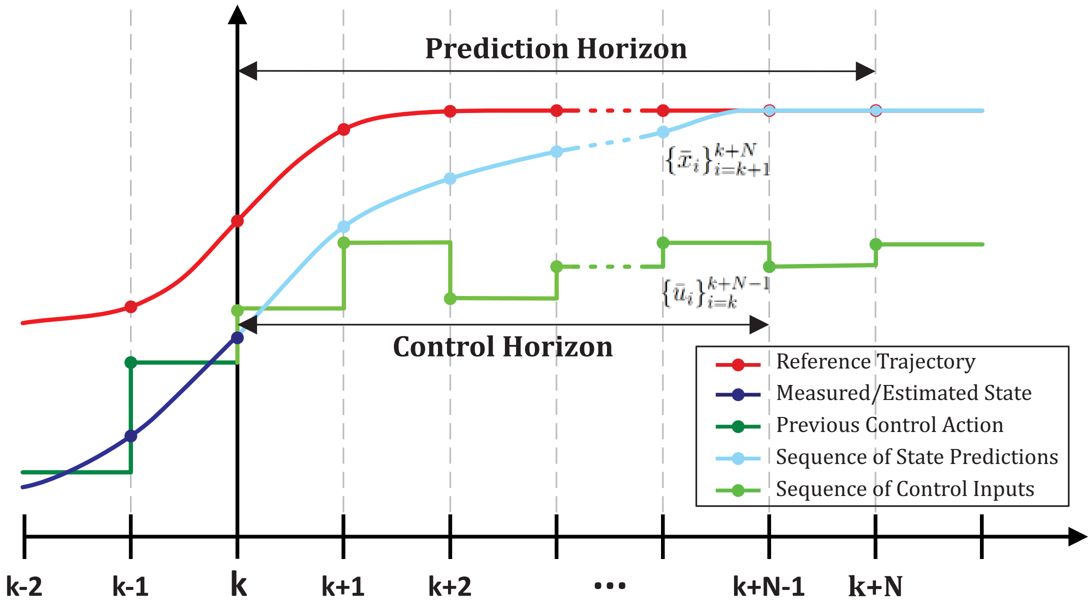


Figure 2.1: Prediction and control horizon scheme in MPC.

2.1.1 General Formulation

Let the system dynamics be described by the following discrete-time model:

$$\begin{aligned} x_k^+ &= f(x_k, u_k) \\ y_k &= g(x_k, u_k), \end{aligned} \tag{2.1}$$

where $x_k^+ \in \mathbb{R}^n$ is the successor state vector, determined by the current state $x_k \in \mathbb{R}^n$ and control input $u_k \in \mathbb{R}^m$; and $y_k \in \mathbb{R}^p$ is the current output vector, which likewise depends on x_k and u_k.

The MPC control law is computed at each sampling instant k by solving the finite-horizon optimal control problem (2.2), where $\ell(\bar{x}_i, \bar{u}_i)$ denotes the stage cost, $F(\bar{x}_{k+N})$ represents the terminal cost, and N defines the prediction horizon length. Although the problem includes equality constraints to initialize the predicted state trajectory at the current system state and to enforce the system dynamics at each time step, it is commonly referred to as an <u>unconstrained</u> optimal control problem. This terminology reflects the fact that no explicit bounds or inequality constraints are imposed on the state and control variables; that is, the entire spaces $\mathbb{R}^n$ and $\mathbb{R}^m$ are admissible for the state and input sequences, respectively.

$$\mathcal{P}_N(x_k) : \begin{cases} \displaystyle\min_{\{\bar{x}_i, \bar{u}_i\}} \sum_{i=k}^{k+N-1} \ell(\bar{x}_i, \bar{u}_i) + F(\bar{x}_{k+N}) \\ \text{subject to:} \\ \quad \bar{x}_k = x_k, \\ \quad \bar{x}_i^+ = f(\bar{x}_i, \bar{u}_i), \quad i = k, \dots, k+N-1. \end{cases} \tag{2.2}$$

In this formulation, the stage cost $\ell(\bar{x}_i, \bar{u}_i)$ penalizes deviations of the predicted states and control inputs from the desired behavior at each step of the prediction horizon. In contrast, the terminal cost $F(\bar{x}_{k+N})$ penalizes the final predicted state at the end of the prediction horizon to ensure desirable properties such as stability and convergence. Together, these costs define the objective function minimized by the MPC, balancing tracking performance, control effort, and closed-loop stability.

The solution of the optimization problem (2.2) yields an optimal sequence of feasible control inputs, $\{\bar{u}_i\}_{i=k}^{k+N-1}$, along with the corresponding state trajectory, $\{\bar{x}_i\}_{i=k+1}^{k+N}$, from a given initial state x_k. However, to preserve prediction accuracy and continuously incorporate new measurements, only the first element of the control horizon, $\bar{u}_k$, is applied to the system as a control input. The remaining elements of this sequence are discarded, and at the next sampling instant, new optimal sequences are computed based on the updated state information.

2.1.2 Constraints

The model (2.1) describes a discrete-time system that may be subject to constraints on both the state and control input sequences. In this context, these variables must satisfy (2.3) and (2.4), where $\mathcal{X} \subset \mathbb{R}^n$ is a convex, closed set representing admissible states, and $\mathcal{U} \subset \mathbb{R}^m$ is a convex, compact set defining admissible control inputs [28]. Both sets are assumed to contain the origin in their interior. The set $\mathcal{X}$ typically encodes safe operating conditions or regulatory requirements, whereas $\mathcal{U}$ reflects actuator limitations or safety bounds on the control inputs.

$$x_k \in \mathcal{X} \tag{2.3}$$

$$u_k \in \mathcal{U} \tag{2.4}$$

In addition, a terminal constraint set $\Omega \subset \mathcal{X}$ is often introduced to ensure closed-loop stability and recursive feasibility [34]. This set is also chosen to be convex and closed, and it is required to contain the origin in its interior. The terminal set Ω defines a region of the state space where a local stabilizing control law can be applied to ensure that the system remains within $\mathcal{X}$ and satisfies the input constraints $\mathcal{U}$ [28]. Typically, Ω is designed to be positively invariant under the terminal control law, ensuring that the system trajectories converge to the desired equilibrium point [35]. Accordingly, the terminal state prediction is required to satisfy the terminal constraint specified in (2.5).

$$x_{k+N} \in \Omega \tag{2.5}$$

To ensure that all previously defined state and control constraints are satisfied, the unconstrained optimization problem in (2.2) is reformulated into the so-called finite-horizon constrained optimal control problem in (2.6), in which explicit constraints are imposed on the state and input trajectories.

$$\mathcal{P}_N(x_k) : \begin{cases} \min\limits_{\{\bar{x}_i,\bar{u}_i\}} \sum\limits_{i=k}^{k+N-1} \ell(\bar{x}_i,\bar{u}_i) + F(\bar{x}_{k+N}) \\ \text{subject to:} \\ \quad \bar{x}_k = x_k, \\ \quad \bar{x}_i^+ = f(\bar{x}_i,\bar{u}_i), \quad i = k,\dots,k+N-1, \\ \quad \bar{x}_i \in \mathcal{X}, \quad i = k+1,\dots,k+N, \\ \quad \bar{u}_i \in \mathcal{U}, \quad i = k,\dots,k+N-1, \\ \quad \bar{x}_{k+N} \in \Omega. \end{cases} \tag{2.6}$$

2.1.3 Stability and Feasibility

To guarantee closed-loop asymptotic stability and recursive feasibility, it is essential to impose appropriate assumptions on the components that define the constrained model predictive control problem (2.6). These assumptions are well-established in the literature [28] and are stated below.

A1: There exists a terminal set $\Omega \subset \mathcal{X}$ that is closed and satisfies $0 \in \Omega$.

A2: There exists an admissible control law $\kappa_\Omega(x_k) \in \mathcal{U}$ for all $x_k \in \Omega$.

A3: The terminal set Ω is positively invariant under $\kappa_\Omega(x_k)$; that is,

$$f(x_k, \kappa_\Omega(x_k)) \in \Omega, \quad \forall x_k \in \Omega.$$

A4: The terminal cost $F(x_k)$ is a local Lyapunov function under $\kappa_\Omega(x_k)$; that is,

$$F(f(x_k, \kappa_\Omega(x_k))) - F(x_k) \leq -\ell(x_k, \kappa_\Omega(x_k)).$$

Assumption A1 ensures that the terminal region is well-defined, satisfies the state constraints, and contains the desired equilibrium point. If the equilibrium point is different from the origin, this condition can still be satisfied by applying a suitable change of coordinates. Assumption A2 guarantees that, within the terminal set Ω, there always exists a control input that satisfies the input constraints. Assumption A3 ensures that if the system starts within Ω and is controlled by $\kappa_\Omega(x_k)$, it will remain in Ω for all future steps, guaranteeing that the terminal region is self-contained. Finally, Assumption A4 establishes that the terminal cost function $F(x_k)$ must act as a local Lyapunov function under the terminal control law $\kappa_\Omega(x_k)$. This means that $F(x_k)$ is positive definite on Ω and decreases along the closed-loop trajectories generated by $\kappa_\Omega(x_k)$. Formally, starting from the definition of $F(x_k)$ as a candidate Lyapunov function, its change along the system trajectories is given by

$$\Delta F(x_k) = F(f(x_k, \kappa_\Omega(x_k))) - F(x_k).$$

To guarantee stability, this difference must satisfy

$$\Delta F(x_k) \leq -\ell(x_k, \kappa_\Omega(x_k)), \quad \forall x_k \in \Omega,$$

where $\ell(x_k, \kappa_\Omega(x_k))$ is the stage cost, which is positive definite on Ω. This inequality ensures that $F(x_k)$ decreases along the system's state trajectories, indicating convergence toward an equilibrium point—typically the origin—while maintaining the state within the admissible set.

2.1.4 Domain of Attraction

An important concept in the analysis and design of MPC is the domain of attraction, which defines the set of initial states $x_k \in \mathcal{X}$ that can be steered into the terminal set Ω within N steps or fewer, where N denotes the control horizon length [36]. Although the true domain of attraction is difficult to characterize, it can be approximated and formally defined as

$$\mathcal{D}_N(\Omega) = \{x_k \in \mathcal{X} \mid \exists j \in \{0, 1, \ldots, N\} : x_{k+j} \in \Omega\}.$$

This definition explicitly emphasizes that any initial state contained within the set $\mathcal{D}_N(\Omega)$ can be driven to the terminal set Ω after a finite number of iterations. A special case arises for $N = 0$, where the domain of attraction is trivially the terminal set itself, i.e.,

$$\mathcal{D}_0(\Omega) = \Omega.$$

The region determined by the approximation of the domain of attraction can assume any convex and closed shape contained within $\mathcal{X} \subset \mathbb{R}^n$. Figure 2.2 illustrates an ellipsoidal domain of attraction, $\mathcal{D}_N(\Omega)$, in a two-dimensional state space. As depicted, this region is typically a subset of the admissible state space $\mathcal{X}$ that contains the terminal set Ω, i.e., $\Omega \subset \mathcal{D}_N(\Omega) \subset \mathcal{X}$. Under Assumptions A1–A4, the terminal set Ω is positively invariant under the terminal control law $\kappa_\Omega(x)$ and serves as a local region of guaranteed stability, ensuring that all trajectories initiated within Ω converge asymptotically to the desired equilibrium point—or to the origin.

The size of $\mathcal{D}_N(\Omega)$ depends on both the size of the terminal set Ω and the length of the control horizon N. Increasing either of these parameters generally enlarges the domain of attraction. The most common approach is to extend the control horizon N (since $\mathcal{D}_1(\Omega) \subset \mathcal{D}_2(\Omega) \subset \cdots \subset \mathcal{D}_N(\Omega)$), which, although effective, results in a larger number of decision variables and consequently higher computational complexity. Alternatively, enlarging the terminal set Ω also expands the domain of attraction (since $\Omega \subset \mathcal{D}_N(\Omega)$) without

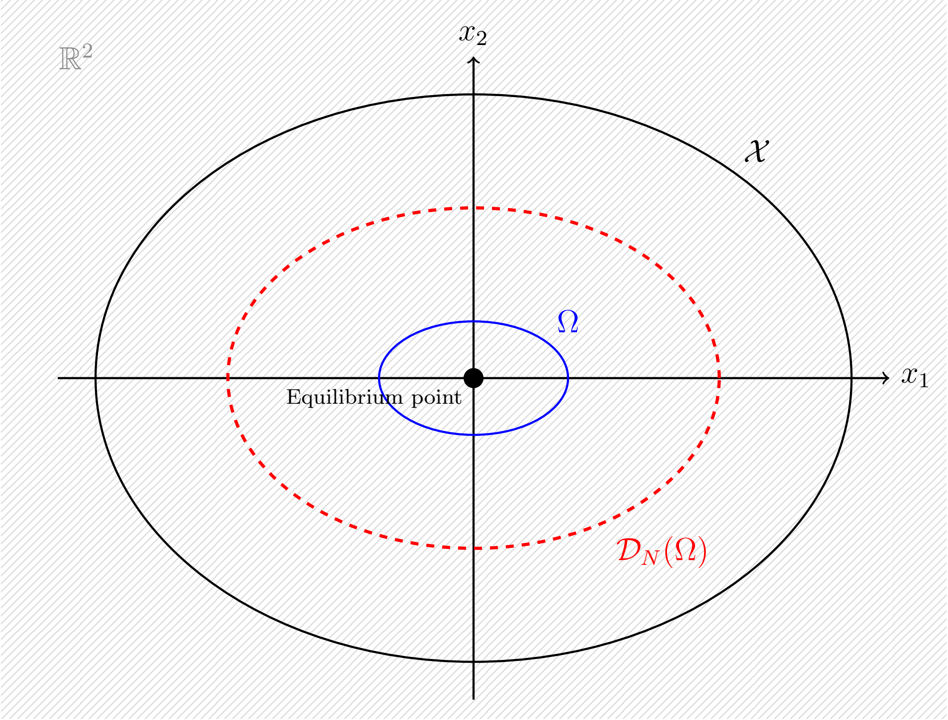


Figure 2.2: Ellipsoidal domain of attraction $\mathcal{D}_N(\Omega) \subset \mathcal{X} \subset \mathbb{R}^2$.

increasing the computational burden, making it an attractive and practical strategy [37].

2.2 Moving Horizon Estimation

State estimation plays a crucial role in the control and monitoring of dynamic systems, particularly when not all state variables are directly measurable. Among the advanced techniques available for this purpose, Moving Horizon Estimation (MHE) stands out as an optimization-based approach that handles constraints and estimates the system states by solving an optimization problem over a receding horizon at each time step. MHE belongs to the class of recursive estimation methods and is characterized by the use of a fixed-length sliding window—referred to as the estimation horizon—over which a sequence of past measurements is processed to infer the current state of the system.

Figure 2.3 illustrates the core concept of an MHE strategy. At each time step k, the estimator uses the most recent M measurements to reconstruct the corresponding M values of the state trajectory over a fixed-length moving horizon. Assuming the first measurement is acquired at $k = 0$, the estimation window becomes fully populated only after $k \geq M - 1$. As time progresses, this window shifts forward, incorporating new measurements and discarding the oldest ones. In this manner, the state trajectory is continuously updated, ensuring that the current estimate reflects the most recent information available.

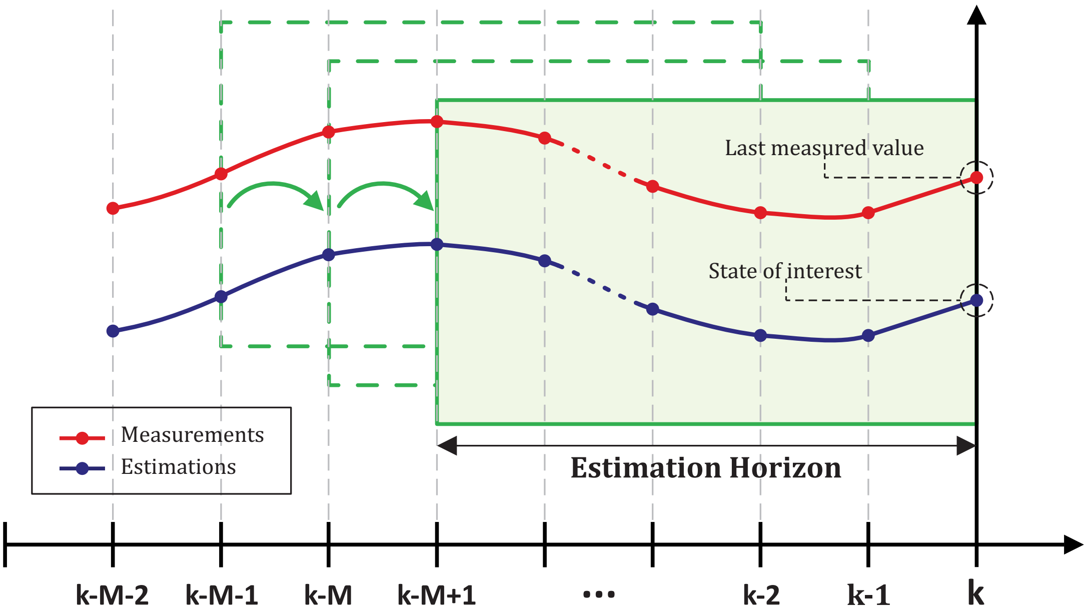


Figure 2.3: Estimation horizon scheme in MHE.

2.2.1 General Formulation

Considering a system governed by the dynamic model in (2.1), the MHE estimate $\hat{x}_k$ is computed by solving the following constrained optimization problem

$$\mathcal{E}_M\big(\{y_i, u_i\}\big) : \begin{cases} \min\limits_{\{\hat{x}_i\}} \; V(\hat{x}_{k-M+1}) + \sum\limits_{i=k-M+1}^{k} \ell_e(\hat{x}_i, y_i, u_i) \\ \text{subject to:} \\ \quad \hat{x}_i^+ = f(\hat{x}_i, u_i), \quad i = k-M+1, \dots, k-1, \\ \quad y_i = g(\hat{x}_i, u_i), \quad i = k-M+1, \dots, k, \\ \quad \hat{x}_i \in \mathcal{X}, \quad i = k-M+1, \dots, k, \end{cases} \tag{2.7}$$

where y_i and u_i correspond to the elements of the estimation horizon, while $\hat{x}_i$ denotes the decision variables. The term $V(\hat{x}_{k-M+1})$ penalizes deviations of the initial estimate at the beginning of the horizon, whereas $\ell_e(\hat{x}_i, u_i, y_i)$ penalizes the output prediction error [34].

2.2.2 Cost Function and Constraints

The MHE problem formulated in (2.7) is defined by an objective function and a set of constraints that reflect both the structure of the system and the available prior knowledge.

The cost function is composed of two main components:

- The arrival cost $V(\hat{x}_{k-M+1})$, which penalizes deviations of the initial state at the beginning of the estimation horizon from a prior estimate. This term encodes information from measurements obtained before the current horizon and helps ensure temporal consistency between successive MHE solutions.

- The stage cost $\ell_e(\hat{x}_i, u_i, y_i)$, which penalizes the discrepancy between the measured outputs y_i and the predicted outputs $g(\hat{x}_i, u_i)$. This term may also include penalties for deviations of the estimated states from nominal trajectories or bounds, depending on the problem formulation.

Both cost components may incorporate weighting matrices that reflect the confidence in model dynamics, sensor measurements, and prior estimates. In linear systems, these terms are typically quadratic, resulting in a convex quadratic program [34].

The constraints in the MHE optimization problem reflect the physical and operational limitations of the system and include:

- Dynamic constraints:

$$\hat{x}_i^+ = f(\hat{x}_i, u_i), \quad i = k - M + 1, \ldots, k - 1,$$

  which ensures that the estimated state trajectory is consistent with the system model and known inputs.

- Output consistency constraints:

$$y_i = g(\hat{x}_i, u_i), \quad i = k - M + 1, \ldots, k,$$

  which ensures that the predicted outputs match the measured outputs as closely as possible under the model.

- State constraints:

$$\hat{x}_i \in \mathcal{X}, \quad i = k - M + 1, \ldots, k,$$

  where $\mathcal{X} \subset \mathbb{R}^n$ defines the admissible set for the state variables, often based on physical limitations (e.g., non-negativity, safety bounds, actuator limits).

Additional constraints can be incorporated into the MHE formulation in (2.7) depending on the specific requirements of the application and the chosen estimation strategy. These constraints may include:

- Disturbance bounds: constraints on process disturbances or noise terms, typically represented as bounded sets (e.g., polytopes or norm-balls), ensuring that uncertainty remains within known limits.

- Soft constraints: relaxed constraints that allow limited violations, introduced via slack variables and penalized in the cost function to preserve feasibility while avoiding overly conservative behavior.

- Algebraic equality or inequality constraints: structural relationships among states, inputs, or outputs that must be satisfied, such as conservation laws, actuator couplings, or physical balance equations.

- Estimation error constraints: restrictions on the allowable deviation between measured and predicted outputs, often used in robust estimation schemes to limit residuals or enforce detectability margins.

The ability to explicitly encode these constraints is one of the key advantages of MHE over classical estimation approaches such as the Kalman Filter, which are typically unconstrained and assume Gaussian noise distributions.

2.2.3 Horizon Estimation and the MHE/MPC Duality

The estimation horizon length is a key factor in the performance of MHE. A longer horizon can improve estimation accuracy and robustness, particularly in the presence of model uncertainties or bounded disturbances, by providing more temporal context for interpreting measurements. However, increasing the horizon length also raises the computational burden, as the size of the optimization problem grows with each additional time step. This trade-off between estimation performance and computational complexity must be carefully balanced, especially in real-time applications.

MHE shares strong structural similarities with MPC. Both methods depend on an accurate system model and involve solving a constrained optimization problem at each time step over a finite horizon. In MPC, the objective is to compute a sequence of future control inputs that optimally steer the system toward a desired reference while respecting constraints. In contrast, MHE operates in the estimation domain, seeking to reconstruct the most probable sequence of past states that best explains the observed outputs, given the known inputs and system dynamics.

Despite this difference in purpose—control versus estimation—MHE and MPC are conceptually complementary. Their integration in closed-loop systems enables the design of advanced control architectures that can optimize performance while ensuring accurate, constraint-aware state feedback in the presence of uncertainty.

2.3 Related Work

MPC is a powerful control strategy that computes an optimal sequence of control inputs over a finite prediction horizon at each time step, enabling the system to track desired trajectories while maintaining stability and performance. Its inherent ability to handle constraints on inputs and states makes it particularly well-suited for developing FTC schemes. When combined with its estimation counterpart (i.e., the MHE), the two methods form the core of an AFTC framework that can be adapted to a wide range of dynamic systems. As presented below, numerous studies have demonstrated the effectiveness of this integration, showing that combining MPC and MHE enables the development of robust AFTC architectures capable of handling nonlinear dynamics, constraints, and faults in complex applications.

In the domain of Unmanned Aerial Vehicles (UAVs), the authors in [38] proposed an AFTC scheme for a quadrotor helicopter subject to actuator faults. The method combines nonlinear MHE for simultaneous estimation of unmeasured states and multiplicative actuator faults with constrained MPC to reconfigure the control law and compensate for the fault effects in real time. A more recent approach for UAVs is presented in [39], where a nonlinear AFTC framework is designed to tolerate up to four simultaneous actuator faults. The estimation layer employs a nonlinear MHE scheme, while the control layer solves a constrained MPC problem to ensure trajectory tracking even under severe actuator degradation.

In the context of autonomous ground vehicles, [40] introduced a Takagi-Sugeno (TS) fuzzy modeling approach to design both the MPC and the MHE estimators. The TS-MPC controls the vehicle's position using a nonlinear kinematic model, while the TS-MHE estimates unmeasured states and the friction force acting on the vehicle, thus reducing control effort and improving robustness.

In [41], a compact optimization-based AFTC architecture is developed for industrial microgrids composed of heterogeneous energy sources, each with its own constraints and load demands. MHE is used for estimating both system states and incipient faults in energy generation subsystems, while MPC performs optimal energy management under operational constraints, ensuring reliability and performance even in faulty scenarios.

A different application is explored in [42], where a two-layer control architecture for hydroelectricity generation in inland waterway management is proposed. The first layer is a supervisory MPC controller that plans water level trajectories, and the second layer employs a local MPC combined with MHE for real-time tracking and fault accommodation using on/off pumps. MHE is crucial in this framework for estimating unmeasured states and identifying actuator degradation, ensuring robust operation.

Finally, [43] presents an AFTC strategy for variable-speed wind turbines. The system is modeled using a TS fuzzy structure to capture plant nonlinearities and constraints. Fault estimation is performed using TS-MHE, and the reconfigured control is implemented through a TS-MPC scheme. Simulation results demonstrate that the strategy maintains system performance and stability under actuator faults.

These works highlight the versatility and effectiveness of combining MPC and MHE in the design of fault-tolerant strategies across diverse applications, including aerial vehicles, autonomous ground platforms, smart grids, water management systems, and renewable energy generation. The ability of these methods to handle constraints and nonlinear dynamics while incorporating online fault estimation makes them especially attractive for safety-critical and resource-constrained environments.

The main contribution of this work is centered on adapting the MHE optimization problem to simultaneously estimate both the system states and the fault residuals, thereby eliminating the need for separate estimation modules. Furthermore, by exploiting fault residuals, the proposed method enables the online identification of faulty system parameters, which are then directly incorporated into the MPC module. This integration allows the controller to compute optimal inputs for fault accommodation while explicitly enforcing state and input constraints, ensuring reliable closed-loop performance even under fault conditions.

Chapter 3

The Three-Tank System

This chapter presents the three-tank system benchmark problem and derives a mathematical model to describe its behavior. This benchmark exhibits a hybrid and nonlinear behavior and is subject to different kinds of perturbations, faults, and noises. It is suitable for designing and evaluating fault detection and diagnosis algorithms and fault-tolerant control techniques. This chapter is structured as follows: Section 3.1 provides a brief overview of the three-tank system's structure and operation. Section 3.2 presents a general nonlinear state-space model of the plant. Finally, Section 3.3 defines a case study, detailing the plant's configuration and deriving a fault model specific to this scenario.

3.1 Plant Description

The plant consists of three cylindrical tanks interconnected by four pipes that enable bidirectional fluid exchange between the lateral tanks (T_1 and T_2) and the central tank (T_3), as illustrated in Figure 3.1. The dashed arrows indicate the reference direction of each flow. The upper pipes and valves that connect the lateral tanks to the central tank are positioned at the same height h_0 and are called transmission pipes and valves. In contrast, the lower pipes and valves are aligned with the base of the tanks and are called connection pipes and valves. At the bottom of each tank are the output pipes and valves.

The tanks are identical and have the same radius and maximum height. Each tank is equipped with a level sensor and is supplied by an independent liquid transfer pump. Similarly, all pipes have the same radius and are fitted with a flow sensor and a control valve to regulate the fluid exchange. In total, the system has thirteen sensors: three level sensors (one per tank) and ten flow sensors (one per valve). Additionally, any valve can be configured as an actuator. When a valve is enabled to be controlled, it becomes an actuator, and an actuator fault (loss of effectiveness) is considered. Otherwise, a plant fault is considered, which could manifest as clogging if the valve operation mode is defined as Open, or leakage if its operation mode is set to Closed.

The volumetric flow rates provided by the independent pumps P_1, P_2, and P_3 (i.e., Q_{P_1}, Q_{P_2}, and Q_{P_3}) are finite and known inputs to the system. The flow rates through

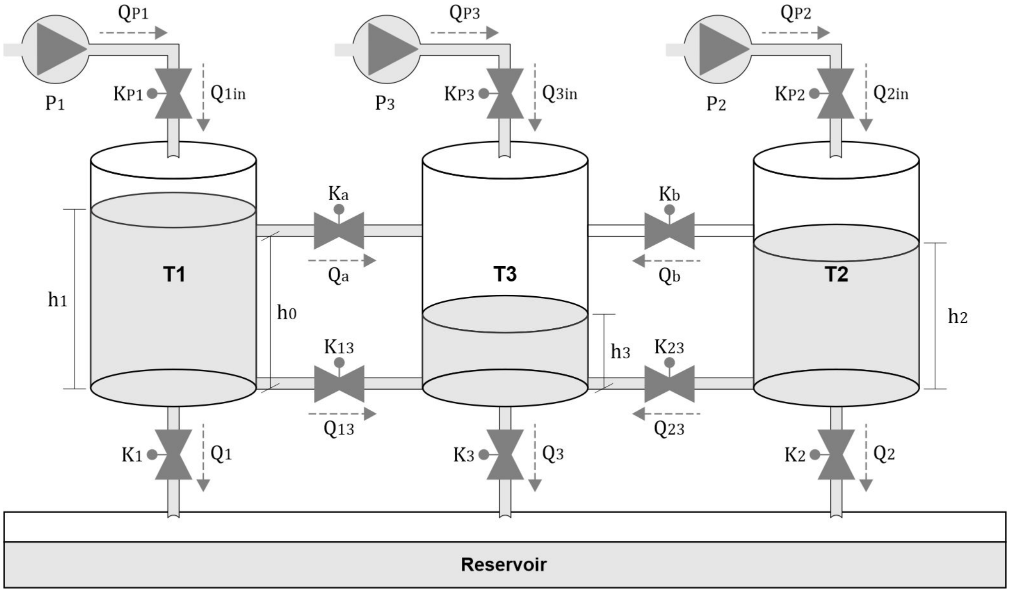


Figure 3.1: The three-tank system. All tanks and pipes are identical; the pumps are independent and can be directly controlled. All valves can operate as actuators, and all levels and flows are measurable.

the transmission, connection, and output pipes are determined by equations (3.1)–(3.3), respectively. In these expressions, K_v, K_{i3}, and $K_j \in [0,1]$ represent the states of the corresponding flow control valves, and $\beta = \mu S\sqrt{2g}$, where μ is the flow correction factor, S is the pipe cross-sectional area, and g is the gravitational acceleration constant. The function $\operatorname{sgn}(\cdot)$ indicates the flow direction in the pipes and is defined in (3.4). The level difference Δh_v in (3.1) is determined by the relative positions of the tank levels (h_1, h_2, and h_3) with respect to the transmission pipe height (h_0). This relationship defines eight possible scenarios for the transmission flow, as summarized in Table 3.1.

$$Q_v = K_v\beta \operatorname{sgn}(\Delta h_v)\sqrt{|\Delta h_v|}, \quad v = \text{a, b}. \tag{3.1}$$

$$Q_{i3} = K_{i3}\beta \operatorname{sgn}(h_i - h_3)\sqrt{|h_i - h_3|}, \quad i = 1,\, 2. \tag{3.2}$$

$$Q_j = K_j\beta\sqrt{h_j}, \quad j = 1,\, 2,\, 3. \tag{3.3}$$

$$\operatorname{sgn}(x) = \begin{cases} 1, & \text{if } \ x \geq 0 \\ -1, & \text{if } \ x < 0. \end{cases} \tag{3.4}$$

The three-tank system is widely adopted in the fault-tolerant control literature as a benchmark for testing estimation and reconfiguration strategies under actuator, plant, and sensor faults. Its nonlinear dynamics, multiple inputs and outputs, and the interconnection between tanks make it an ideal platform for evaluating model-based estimation and control techniques. Numerous studies have used this system to validate approaches such as model-based fault diagnosis, model predictive control, and integrated estimation-control schemes [44, 45, 46].

Table 3.1: Transmission flow scenarios for the three-tank system.

Scenarios			Δh_a	Δh_b
$h_1 \leq h_0$	$h_2 \leq h_0$	$h_3 \leq h_0$	0	0
$h_1 \leq h_0$	$h_2 \leq h_0$	$h_3 > h_0$	$h_0 - h_3$	$h_0 - h_3$
$h_1 \leq h_0$	$h_2 > h_0$	$h_3 \leq h_0$	0	$h_2 - h_3$
$h_1 \leq h_0$	$h_2 > h_0$	$h_3 > h_0$	$h_0 - h_3$	$h_2 - h_3$
$h_1 > h_0$	$h_2 \leq h_0$	$h_3 \leq h_0$	$h_1 - h_0$	0
$h_1 > h_0$	$h_2 \leq h_0$	$h_3 > h_0$	$h_1 - h_3$	$h_0 - h_3$
$h_1 > h_0$	$h_2 > h_0$	$h_3 \leq h_0$	$h_1 - h_0$	$h_2 - h_0$
$h_1 > h_0$	$h_2 > h_0$	$h_3 > h_0$	$h_1 - h_3$	$h_2 - h_3$

3.2 Nonlinear Model

The volume variation, $\dot{V}$, inside a cylindrical tank of cross-sectional area S_c, can be described by equation (3.5), where $\dot{h}$ is the level variation inside the tank, $\sum Q_{\text{in}}$ is the sum over all input flows into the tank, and $\sum Q_{\text{out}}$ is the sum over all output flows from the tank.

$$\dot{V} = S_c\dot{h} = \sum Q_{\text{in}} - \sum Q_{\text{out}} \tag{3.5}$$

Using equation (3.5) to perform the mass balance in each tank of the system shown in Figure 3.1, the following set of state equations can be obtained:

$$\begin{aligned}
\dot{h}_1 &= -\frac{1}{S_c}(Q_a + Q_{13} + Q_1) + \frac{1}{S_c}K_{P_1}Q_{P_1} \\
\dot{h}_2 &= -\frac{1}{S_c}(Q_b + Q_{23} + Q_2) + \frac{1}{S_c}K_{P_2}Q_{P_2} \\
\dot{h}_3 &= \frac{1}{S_c}(Q_a + Q_b + Q_{13} + Q_{23} - Q_3) + \frac{1}{S_c}K_{P_3}Q_{P_3}
\end{aligned} \tag{3.6}$$

The output vector consists of all system variables that can be measured via level and flow sensors. In particular, for the system illustrated in Figure 3.1 the output vector has the following form:

$$y = [h_1,\, h_2,\, h_3,\, Q_{1\text{in}},\, Q_{2\text{in}},\, Q_{3\text{in}},\, Q_a,\, Q_b,\, Q_{13},\, Q_{23},\, Q_1,\, Q_2,\, Q_3]^\top , \tag{3.7}$$

where $Q_{j\text{in}} = K_{P_j}Q_{P_j}$ for $j = 1,\, 2,\, 3$.

The nonlinear model defined by equations (3.6)–(3.7) captures the essential dynamics of the system and serves as the basis for the design of controllers, estimators, and simulation studies. It describes the hydraulic interactions between the tanks, the influence of valve positions on flow distribution, and the contribution of external inflows supplied by the independent pumps.

3.3 Case Study Definition

The three-tank system benchmark provides a versatile framework for defining case studies under various configurations and interconnections between the tanks. These configurations enable the exploration of different operating conditions, fault scenarios, and control strategies. In this work, the case study focuses on a representative configuration illustrated in Figure 3.2. This configuration allows the analysis of the core dynamics of the system while simplifying the model to focus on key interactions relevant to control and fault diagnosis.

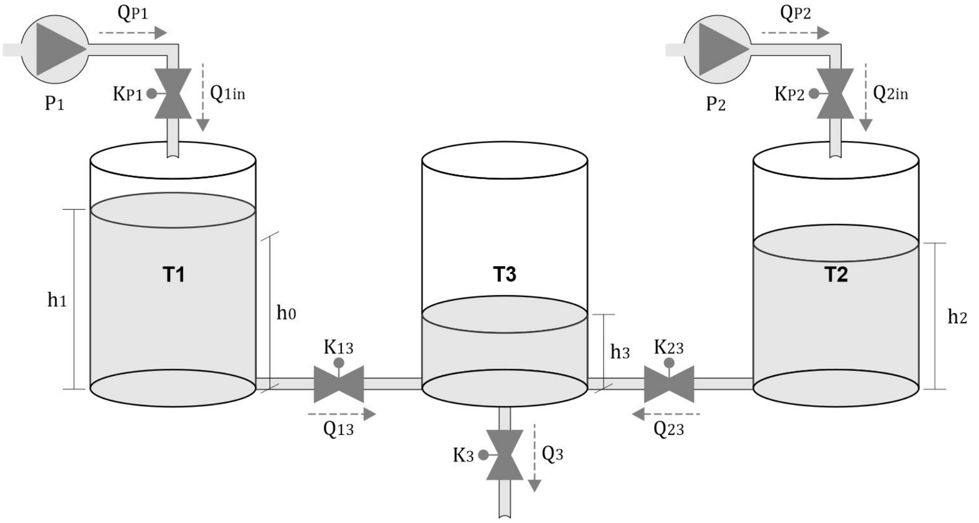


Figure 3.2: Configuration of the three-tank system used as the case study. The inlet flows to the lateral tanks (T_1 and T_2) are regulated by valves K_{P_1} and K_{P_2}. The valves K_{13}, K_{23}, and K_3 are assumed to be permanently open (i.e., $K_{13} = K_{23} = K_3 = 1$), ensuring fixed hydraulic connections between the tanks and the reservoir, and therefore do not operate in flow regulation. All other valves and pipes not depicted are considered inactive or neglected for the purposes of this study.

Based on the nonlinear state model (3.6) and assuming that the input and state vectors are defined as $u = [K_{\mathrm{P}_1},\, K_{\mathrm{P}_2}]^\top$ and $x = [h_1,\, h_2,\, h_3]^\top$. Then, the state equations of the scenario illustrated in Figure 3.2 can be expressed as:

$$\begin{gathered}
\dot{x}_1 = -\frac{\beta}{S_\mathrm{c}}\operatorname{sgn}(x_1 - x_3)\sqrt{|x_1 - x_3|} + \frac{Q_{\mathrm{P}_1}}{S_\mathrm{c}}u_1 \\
\dot{x}_2 = -\frac{\beta}{S_\mathrm{c}}\operatorname{sgn}(x_2 - x_3)\sqrt{|x_2 - x_3|} + \frac{Q_{\mathrm{P}_2}}{S_\mathrm{c}}u_2 \\
\dot{x}_3 = \frac{\beta}{S_\mathrm{c}}\operatorname{sgn}(x_1 - x_3)\sqrt{|x_1 - x_3|} + \frac{\beta}{S_\mathrm{c}}\operatorname{sgn}(x_2 - x_3)\sqrt{|x_2 - x_3|} - \frac{\beta}{S_\mathrm{c}}\sqrt{x_3}
\end{gathered} \tag{3.8}$$

The output vector includes measurable variables chosen to support both control and fault diagnosis objectives and can be written in the form of $y = [h_1,\, h_2,\, Q_{13},\, Q_{23},\, Q_3]^\top$, where each element is shown in detail in (3.9). Although x_3 is measurable, it is intentionally omitted from the output vector to emphasize the role of the state estimator.

$$\begin{aligned} y_1 &= x_1 \\ y_2 &= x_2 \\ y_3 &= \beta \operatorname{sgn}(x_1 - x_3)\sqrt{|x_1 - x_3|} \\ y_4 &= \beta \operatorname{sgn}(x_2 - x_3)\sqrt{|x_2 - x_3|} \\ y_5 &= \beta\sqrt{x_3} \end{aligned} \tag{3.9}$$

This case study provides a meaningful context for testing control algorithms, estimator designs, and fault detection techniques, as it captures the essential coupling between tanks and the nonlinear flow dynamics characteristic of hydraulic systems.

3.3.1 Fault Model

Faults might occur in the three-tank system due to mechanical or electrical damage. They generally appear due to the wear of essential devices for the system operation, e.g., valves, pumps, level and flow sensors. Additionally, faults might occur due to physical damage in the system structure, which might cause leakage in tanks or clogging in pipes of the system [47]. In this work, the faults affecting the configuration shown in Figure 3.2 are listed in Table 3.2. Each fault is represented by a normalized magnitude $f_\gamma \in [0, 1]$, where γ is the fault identifier, and f_γ denotes the fault magnitude. Therefore, $f_\gamma = 1$ indicates that the fault γ occurs with maximum magnitude, and $f_\gamma = 0$ signifies that the fault γ does not occur in the system.

Table 3.2: List of faults for the default case study configuration.

Symbol	Description	Symbol	Description
f_1	Loss of effectiveness in K_{P_1}	f_{11}	Scaling fault in level sensor h_1
f_2	Loss of effectiveness in K_{P_2}	f_{12}	Scaling fault in level sensor h_2
f_6	Blocking in flow Q_{13}	f_{19}	Scaling fault in flow sensor Q_{13}
f_9	Leakage in tank T_2	f_{20}	Scaling fault in flow sensor Q_{23}
f_{10}	Blocking in outflow Q_3	f_{23}	Scaling fault in flow sensor Q_3

The nonlinear state and output equations in (3.8) and (3.9) are reformulated to explicitly account for the effects of the faults listed in Table 3.2. Consequently, the state and output equations can be expressed as the following fault-augmented model:

$$\begin{aligned} \dot{x}_1 &= -(1 - f_6)\frac{\beta}{S_\mathrm{c}} \operatorname{sgn}(x_1 - x_3)\sqrt{|x_1 - x_3|} + (1 - f_1)K_{\mathrm{P}_1}\frac{Q_{\mathrm{P}_1}}{S_\mathrm{c}} \\ \dot{x}_2 &= -\frac{\beta}{S_\mathrm{c}} \operatorname{sgn}(x_2 - x_3)\sqrt{|x_2 - x_3|} - f_9\frac{\beta}{S_\mathrm{c}}\sqrt{x_2} + (1 - f_2)K_{\mathrm{P}_2}\frac{Q_{\mathrm{P}_2}}{S_\mathrm{c}} \\ \dot{x}_3 &= (1 - f_6)\frac{\beta}{S_\mathrm{c}} \operatorname{sgn}(x_1 - x_3)\sqrt{|x_1 - x_3|} + \frac{\beta}{S_\mathrm{c}} \operatorname{sgn}(x_2 - x_3)\sqrt{|x_2 - x_3|} - (1 - f_{10})\frac{\beta}{S_\mathrm{c}}\sqrt{x_3} \end{aligned} \tag{3.10}$$

$$\begin{aligned} y_1 &= (1-f_{11})x_1 \\ y_2 &= (1-f_{12})x_2 \\ y_3 &= (1-f_{19})(1-f_6)\beta \operatorname{sgn}(x_1-x_3)\sqrt{|x_1-x_3|} \\ y_4 &= (1-f_{20})\beta \operatorname{sgn}(x_2-x_3)\sqrt{|x_2-x_3|} \\ y_5 &= (1-f_{23})(1-f_{10})\beta\sqrt{x_3} \end{aligned} \tag{3.11}$$

3.3.2 Linear Fault Model

The nonlinear fault model given by (3.10) and (3.11) exhibits significant nonlinearities due to the flow dynamics. To mitigate the nonlinearity and maintain the fault degradation characteristics, the model is linearized around a selected operating point $(x_{\mathrm{op}}, u_{\mathrm{op}})$, assuming that fault magnitudes remain constant over the linearization interval. As a result, the following model is obtained:

$$\begin{aligned} \delta\dot{x} &= A(f)\,\delta x + B(f)\,\delta u \\ \delta y &= C(f)\,\delta x + D(f)\,\delta u, \end{aligned} \tag{3.12}$$

where

$$\begin{aligned} \delta x &= x - x_{\mathrm{op}}, \\ \delta u &= u - u_{\mathrm{op}}, \\ f &= [f_1,\ f_2,\ f_6,\ f_9,\ f_{10},\ f_{11},\ f_{12},\ f_{19},\ f_{20},\ f_{23}]^\top . \end{aligned}$$

The Jacobian matrices that define the model dynamics are computed as follows:

$$\begin{aligned} A(f) &= \left.\frac{\partial g(x,u)}{\partial x}\right|_{\substack{x=x_{\mathrm{op}}\\ u=u_{\mathrm{op}}}}, \qquad & B(f) &= \left.\frac{\partial g(x,u)}{\partial u}\right|_{\substack{x=x_{\mathrm{op}}\\ u=u_{\mathrm{op}}}}, \\ C(f) &= \left.\frac{\partial h(x,u)}{\partial x}\right|_{\substack{x=x_{\mathrm{op}}\\ u=u_{\mathrm{op}}}}, \qquad & D(f) &= \left.\frac{\partial h(x,u)}{\partial u}\right|_{\substack{x=x_{\mathrm{op}}\\ u=u_{\mathrm{op}}}}. \end{aligned} \tag{3.13}$$

For a given operating point $x_{\mathrm{op}} = [x_1^0,\ x_2^0,\ x_3^0]^\mathsf{T}$ satisfying $x_1^0,\ x_2^0 \geq x_3^0$, the corresponding system matrices are given by:

$$A(f) = \frac{\beta}{2S_c}\begin{bmatrix} -\frac{1-f_6}{\sqrt{x_1^0 - x_3^0}} & 0 & \frac{1-f_6}{\sqrt{x_1^0 - x_3^0}} \\ 0 & -\frac{1}{\sqrt{x_2^0 - x_3^0}} - \frac{f_9}{\sqrt{x_2^0}} & \frac{1}{\sqrt{x_2^0 - x_3^0}} \\ \frac{1-f_6}{\sqrt{x_1^0 - x_3^0}} & \frac{1}{\sqrt{x_2^0 - x_3^0}} & -\left(\frac{1-f_6}{\sqrt{x_1^0 - x_3^0}} + \frac{1}{\sqrt{x_2^0 - x_3^0}} + \frac{1-f_{10}}{\sqrt{x_3^0}}\right) \end{bmatrix},$$

$$B(f) = \begin{bmatrix} (1-f_1)\frac{Q_{P_1}}{S_c} & 0 \\ 0 & (1-f_2)\frac{Q_{P_2}}{S_c} \\ 0 & 0 \end{bmatrix},$$

$$C(f) = \begin{bmatrix} 1-f_{11} & 0 & 0 \\ 0 & 1-f_{12} & 0 \\ \frac{(1-f_{19})(1-f_6)\beta}{2\sqrt{x_1^0 - x_3^0}} & 0 & -\frac{(1-f_{19})(1-f_6)\beta}{2\sqrt{x_1^0 - x_3^0}} \\ 0 & \frac{(1-f_{20})\beta}{2\sqrt{x_2^0 - x_3^0}} & -\frac{(1-f_{20})\beta}{2\sqrt{x_2^0 - x_3^0}} \\ 0 & 0 & \frac{(1-f_{23})(1-f_{10})\beta}{2\sqrt{x_3^0}} \end{bmatrix}, \quad D(f) = 0. \tag{3.14}$$

Since tanks T1 and T2 are similar and supplied by identical pumps, it is reasonable to assume $Q_{P_1} = Q_{P_2} = Q_P$, $x_1^0 = x_2^0 = x^0$, and $u_1^0 = u_2^0$. Furthermore, at the chosen operating point of the linearized model, the system is at steady state (i.e., $\dot{x} = 0$). Under these conditions, and using (3.8), the operating point corresponding to a given x^0 is

$$(x_{op}, u_{op}) = \left(\left[x^0, x^0, \frac{4}{5}x^0\right]^\top, \frac{\beta}{Q_P}\left[\sqrt{\frac{x^0}{5}}, \sqrt{\frac{x^0}{5}}\right]^\top\right). \tag{3.15}$$

This linear approximation is particularly useful for real-time implementations where computational efficiency is essential. The choice of operating point and the assumption of constant fault magnitudes must, however, be carefully considered to ensure the validity of the linear model within the expected range of system operation.

3.3.3 Discrete-Time Model

To enable the digital implementation of control, estimation, and fault diagnosis techniques, a discrete-time model of the system is required. This discrete model is derived by applying the Zero-Order Hold (ZOH) method to the linearized continuous-time fault model given by (3.12). Considering a sampling time T_s, the discrete-time representation is expressed as:

$$\begin{aligned} \delta x_k^+ &= A_d(f_k)\,\delta x_k + B_d(f_k)\,\delta u_k \\ \delta y_k &= C(f_k)\,\delta x_k + D(f_k)\,\delta u_k, \end{aligned}$$

where the discrete-time matrices $A_d(f_k)$ and $B_d(f_k)$ are computed as

$$A_d(f_k) = e^{A(f)\,T_\mathrm{s}}, \quad B_d(f_k) = \int_0^{T_\mathrm{s}} e^{A(f)\,\tau} B(f)\,d\tau.$$

For practical implementation and to reduce computational burden, the matrix exponential and integral are typically approximated. In this work, a truncated series expansion is used to compute $A_d(f_k)$ and $B_d(f_k)$:

$$\begin{aligned} A_d(f_k) &= I + A(f)\,T_\mathrm{s} + \frac{1}{2!}A(f)^2\,{T_\mathrm{s}}^2 + \frac{1}{3!}A(f)^3\,{T_\mathrm{s}}^3 + \ldots \\ B_d(f_k) &= B(f)\,T_\mathrm{s} + \frac{1}{2!}A(f)B(f)\,{T_\mathrm{s}}^2 + \frac{1}{3!}A(f)^2B(f)\,{T_\mathrm{s}}^3 + \ldots \end{aligned}$$

To simplify the formulation and enhance computational efficiency for real-time applications, $A_d(f_k)$ is approximated by retaining only the first two terms of its expansion, whereas $B_d(f_k)$ is limited to its first term. Higher-order terms are discarded.

$$\begin{aligned} A_d(f_k) &\approx I + A(f)\,T_\mathrm{s} \\ B_d(f_k) &\approx B(f)\,T_\mathrm{s} \end{aligned}$$

This corresponds to Euler's forward approximation of the continuous-time dynamics (i.e., $\dot{x} \approx \frac{x_k^+ - x_k}{T_\mathrm{s}}$), which is adequate when the sampling time T_s is sufficiently small relative to the system time constants.

For notational simplicity, the delta symbol δ used to denote deviations will be omitted. The variables x_k, u_k, and y_k will represent the discrete-time (sampled) system states, inputs, and outputs, respectively. The resulting discrete-time LTI model under fault conditions can be expressed as:

$$\begin{aligned} x_k^+ &= A(f_k)\,x_k + B(f_k)\,u_k \\ y_k &= C(f_k)\,x_k + D(f_k)\,u_k, \end{aligned} \tag{3.16}$$

where $A(f_k)$ and $B(f_k)$ denote the matrices of the discrete-time faulty system, computed as $A(f_k) = I + A(f)\,T_\mathrm{s}$ and $B(f_k) = B(f)\,T_\mathrm{s}$, respectively, based on the continuous-time matrices $A(f)$ and $B(f)$ presented in (3.14).

This discrete model forms the basis for digital simulation, for model-based controller design, and for estimators. The choice of discretization method and sampling period T_s must be made carefully to balance model accuracy and computational efficiency.

Chapter 4

Methodology

This chapter presents the methodology developed to implement the proposed AFTC strategy based on the integration of MHE and MPC. The main objective is to enable real-time fault detection and compensation through model adaptation, ensuring closed-loop stability, constraint satisfaction, and reliable setpoint tracking even in the presence of faults. Section 4.1 details the design and implementation of the MPC controller, including the definition of cost functions, state and input constraints, and the derivation of terminal ingredients to ensure stability and recursive feasibility. The computation of the terminal set using LMIs, the approximation of the domain of attraction, and the formulation of the tracking control strategy are also addressed. Section 4.2 introduces the MHE formulation used for state estimation, emphasizing the minimization of prediction errors over a receding horizon and the treatment of input and output constraints. Section 4.3 describes the fault residual estimation process. By augmenting the system model to include fault residuals as additional states, the MHE is adapted to estimate their values online. Section 4.4 presents the control reconfiguration procedure, which uses the estimated fault residuals to update the system matrices and modify the internal prediction model of the MPC, forming a Fault-Tolerant MPC (FTMPC) capable of mitigating the effects of faults. Finally, Section 4.5 outlines the simulation framework used to validate the proposed strategy, including details on the simulation environment, system setup, and MPC and MHE configuration parameters.

4.1 MPC Implementation

Consider the discrete-time LTI state-space model defined in (4.1), with n states, m control inputs, and p outputs. In this formulation, the system assumes no direct feedthrough term from input to output, i.e., $D \in \mathbb{R}^{p\times m}$ is zero. The matrices are defined as follows: $A \in \mathbb{R}^{n\times n}$ is the state matrix, $B \in \mathbb{R}^{n\times m}$ is the input matrix, and $C \in \mathbb{R}^{p\times n}$ is the output matrix.

$$\begin{aligned} x_k^+ &= Ax_k + Bu_k \\ y_k &= Cx_k \end{aligned} \tag{4.1}$$

The objective is to compute a sequence of control inputs that optimally steers the system states toward a desired target—typically the origin—while minimizing the deviation between the predicted and actual state trajectories over a finite prediction horizon of N time steps (see Figure 2.1). For the MPC to operate correctly, full knowledge of the system states is initially required. Since not all states are directly measurable, this requirement implies that the system must be fully observable, thereby ensuring that the complete state vector can be reconstructed from the available output measurements. Under these assumptions, the unconstrained MPC optimization problem (2.2) can be reformulated to establish the following relationship between the predicted state and control input sequences:

$$\mathcal{P}_N(x_k) : \begin{cases} \min\limits_{\{\bar{x}_i, \bar{u}_i\}} \sum\limits_{i=k}^{k+N-1} \left(|\bar{x}_i|_Q^2 + |\bar{u}_i|_R^2\right) + \left(|\bar{x}_{k+N}|_P^2\right) \\ \text{subject to:} \\ \quad \bar{x}_k = x_k, \\ \quad \bar{x}_i^+ = A\bar{x}_i + B\bar{u}_i, \quad i = k, \dots, k+N-1, \end{cases} \tag{4.2}$$

where the notations $|\bar{x}_i|_Q^2$, $|\bar{u}_i|_R^2$, and $|\bar{x}_{k+N}|_P^2$ represent the quadratic forms $\bar{x}_i^\top Q \bar{x}_i$, $\bar{u}_i^\top R \bar{u}_i$, and $\bar{x}_{k+N}^\top P \bar{x}_{k+N}$, respectively. The penalty matrices $Q \in \mathbb{R}^{n\times n}$, $R \in \mathbb{R}^{m\times m}$, and $P \in \mathbb{R}^{n\times n}$ are assumed to be real and symmetric, with $Q \succeq 0$ and $P \succeq 0$, and $R \succ 0$. These conditions are sufficient to guarantee the convexity of the quadratic cost function and ensure the existence and uniqueness of the optimal solution to the linear quadratic control problem (4.2) [34].

Note that the objective function in (4.2) covers all elements of the prediction and control horizons and is composed of two terms:

- Stage cost: penalizes deviations of the predicted states and control inputs at each time step within the control horizon,

  $$\ell(\bar{x}_i, \bar{u}_i) = \bar{x}_i^\top Q \bar{x}_i + \bar{u}_i^\top R \bar{u}_i, \tag{4.3}$$

  which ensures performance objectives, such as tracking and energy efficiency;

- Terminal cost: enforces convergence properties at the end of the prediction horizon,

  $$F(\bar{x}_{k+N}) = \bar{x}_{k+N}^\top P \bar{x}_{k+N}, \tag{4.4}$$

  which promotes asymptotic stability when combined with an appropriate terminal set.

4.1.1 Computation of the Terminal Set

The design of an appropriate terminal set $\Omega \subset \mathbb{R}^n$ is essential to ensure both closed-loop asymptotic stability and recursive feasibility in MPC problems. As stated in Assumptions A1–A4, the terminal set defines a region in the state space where a local control law can be applied to guarantee constraint satisfaction and convergence to the origin. In this study, the terminal set Ω is obtained via Lyapunov-based analysis for discrete-time LTI systems and formulated as a set of LMI constraints.

Let the system (4.1) be controlled by the following linear state-feedback control law:

$$\kappa_\Omega(x_k) := u_k = Kx_k, \tag{4.5}$$

where $K \in \mathbb{R}^{m\times n}$ is a stabilizing gain matrix. The resulting closed-loop dynamics are:

$$x_k^+ = (A + BK)x_k = A_K x_k. \tag{4.6}$$

According to Assumption A4, the terminal cost function (4.4) serves as a local Lyapunov function under the terminal control law (4.5). From this condition, the following inequality must be satisfied:

$$\begin{aligned}
&F(x_k^+) - F(x_k) \leq -\ell(x_k, Kx_k) \\
&\Rightarrow x_k^\top A_K^\top P A_K x_k - x_k^\top P x_k \leq -\left(x_k^\top Q x_k + x_k^\top K^\top R K x_k\right) \\
&\Rightarrow x_k^\top \left(A_K^\top P A_K - P + Q + K^\top R K\right) x_k \leq 0,
\end{aligned}$$

where A_K denotes the closed-loop matrix (4.6).

Therefore, the terminal cost function (4.4) strictly decreases along closed-loop trajectories, consistent with the stage cost function (4.3), under the control law (4.5), whenever the following LMI holds:

$$A_K^\top P A_K - P + Q + K^\top R K \prec 0,$$

for $P \succ 0$, $Q \succ 0$, and $R \succ 0$. This condition is equivalently expressed as the following LMI via the Schur complement:

$$\begin{bmatrix} P^{-1} & A_K \\ A_K^\top & P - Q - K^\top R K \end{bmatrix} \succ 0. \tag{4.7}$$

By lifting inequality (4.7), applying the Schur complement to the block $(2,2)$, and performing an appropriate congruence transformation, a new equivalent LMI is obtained:

$$\begin{bmatrix} P & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}^\top \left[\begin{array}{cc|c} P^{-1} & A_K & 0 \\ A_K^\top & P - Q & K^\top \\ \hline 0 & K & R^{-1} \end{array}\right] \begin{bmatrix} P & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \succ 0 \quad \Rightarrow \quad \begin{bmatrix} P & PA_K & 0 \\ \star & P - Q & K^\top \\ \star & \star & R^{-1} \end{bmatrix} \succ 0, \tag{4.8}$$

where $(\star)$ denotes the symmetric terms implied by the LMI structure.

Substituting the closed-loop matrix $A_K = A+BK$ into (4.8) and applying a congruence transformation with $\operatorname{diag}(P^{-1},\, P^{-1}, I)$ yields:

$$\begin{bmatrix} P^{-1} & AP^{-1} + BKP^{-1} & 0 \\ \star & P^{-1} - P^{-1}QP^{-1} & P^{-1}K^\top \\ \star & \star & R^{-1} \end{bmatrix} \succ 0.$$

To address the nonlinearity in the block $(2,2)$, the Schur complement is reapplied, resulting in another equivalent LMI:

$$\left[\begin{array}{cc|c|c} P^{-1} & AP^{-1} + BKP^{-1} & 0 & 0 \\ \star & P^{-1} & P^{-1} & 0 \\ \hline \star & \star & Q^{-1} & P^{-1}K^\top \\ \hline \star & \star & \star & R^{-1} \end{array}\right] \succ 0.$$

By introducing the change of variables $Y = Q^{-1} \in \mathbb{R}^{n\times n}$, $Z = R^{-1} \in \mathbb{R}^{m\times m}$, $W = P^{-1} \in \mathbb{R}^{n\times n}$, and $L = KP^{-1} \in \mathbb{R}^{m\times n}$, the following LMI problem is formulated:

$$\mathcal{L}(A,B,Y,Z) : \begin{cases} W = W^\top \succ 0, \\ \begin{bmatrix} W & AW + BL & 0 & 0 \\ \star & W & W & 0 \\ \star & \star & Y & L^\top \\ \star & \star & \star & Z \end{bmatrix} \succ 0, \end{cases} \tag{4.9}$$

where the variables of interest are recovered as $P = W^{-1}$ and $K = LP$.

Once a feasible solution $\{W, L\}$ to (4.9) is obtained, the corresponding Lyapunov matrix $P = W^{-1}$ not only characterizes the terminal cost function (4.4) but also defines the ellipsoidal terminal set described by

$$\Omega := \left\{ x_k \in \mathbb{R}^n \;\middle|\; x_k^\top P x_k \leq 1 \right\}, \tag{4.10}$$

which guarantees asymptotic stability of the closed-loop system under the terminal control law $\kappa_{\Omega(x_k)} := LPx_k$. However, this ellipsoidal set is derived under the assumption of an unconstrained system and may not lie entirely within the admissible region defined by the state and input constraints. The next subsections introduce a refinement of the terminal set construction by incorporating state and input constraints in the form of polytopes, enabling the design of a constraint-admissible invariant terminal set.

4.1.2 Definition of State and Input Constraints

The MPC formulation explicitly incorporates constraints on both the system states and control inputs to ensure that the closed-loop system operates safely and within feasible limits. These constraints are typically represented as box constraints:

$$x_{\text{lb}} \leq x_k \leq x_{\text{ub}} \quad \Rightarrow \quad x_k \in \mathcal{X} := \left\{ x_k \in \mathbb{R}^n \;\middle|\; \begin{bmatrix} I \\ -I \end{bmatrix} x_k \leq \begin{bmatrix} x_{\text{ub}} \\ -x_{\text{lb}} \end{bmatrix} \right\}, \tag{4.11}$$

$$u_{\text{lb}} \leq u_k \leq u_{\text{ub}} \quad \Rightarrow \quad u_k \in \mathcal{U} := \left\{ u_k \in \mathbb{R}^m \;\middle|\; \begin{bmatrix} I \\ -I \end{bmatrix} u_k \leq \begin{bmatrix} u_{\text{ub}} \\ -u_{\text{lb}} \end{bmatrix} \right\}, \tag{4.12}$$

where x_{lb} and x_{ub} define the lower and upper bounds on the state vector, and u_{lb} and u_{ub} define the admissible bounds for the control input vector. These constraints are fundamental in practical control applications, as they ensure that the system operates within predefined safety margins and that the computed control inputs remain consistent with the physical and operational limitations, such as actuator saturation and safety-critical thresholds.

To ensure consistency with these constraints, all predicted states and control inputs over the horizons (see Figure 2.1) must also satisfy the corresponding bounds. Formally, the state and input constraint sets (4.11) and (4.12) are extended to these elements and compactly represented as linear inequalities in the following form:

$$\bar{x}_i \in \mathcal{X} := \left\{ \bar{x}_i \in \mathbb{R}^n \;\middle|\; \begin{bmatrix} I \\ -I \end{bmatrix} \bar{x}_i \leq \begin{bmatrix} x_{\text{ub}} \\ -x_{\text{lb}} \end{bmatrix}, \forall i \in \{k+1, \dots, k+N\} \right\}, \tag{4.13}$$

$$\bar{u}_i \in \mathcal{U} := \left\{ \bar{u}_i \in \mathbb{R}^m \;\middle|\; \begin{bmatrix} I \\ -I \end{bmatrix} \bar{u}_i \leq \begin{bmatrix} u_{\text{ub}} \\ -u_{\text{lb}} \end{bmatrix}, \forall i \in \{k, \dots, k+N-1\} \right\}. \tag{4.14}$$

Incorporating these constraints not only prevents the violation of operational boundaries but also contributes to the closed-loop stability and recursive feasibility of the predictive controller, particularly when complemented by an appropriately designed terminal cost and a well-defined terminal set, as discussed in the next subsection.

4.1.3 Formulation of the Terminal Constraint

An H-polyhedron is defined as the intersection of a finite number of closed half-spaces, each corresponding to a linear inequality. The state constraint set (4.11) can be equivalently expressed as:

$$\mathcal{X} := \left\{ x_k \in \mathbb{R}^n \;\middle|\; a_j^\top x_k \leq 1,\, \forall j \in \{1,\, 2,\, \dots,\, l\} \right\}, \tag{4.15}$$

where l is the number of hyperplanes that define the polyhedral set $\mathcal{X}$, and $a_j^\top \in \mathbb{R}^{1\times n}$ represents the normal vector to the j-th hyperplane.

To ensure that the terminal set (4.10) is entirely contained within the admissible set $\mathcal{X} \subset \mathbb{R}^n$, defined by (4.15), the following conditions must be simultaneously satisfied:

$$\begin{aligned} g_j(x_k) &:= a_j^\top x_k - 1 \leq 0, \quad \forall j \in \{1, 2, \ldots, l\}, \\ g_{l+1}(x_k) &:= x_k^\top P x_k - 1 \leq 0. \end{aligned}$$

The S-procedure can be employed to ensure that all constraints $g_j(x_k) \leq 0$ hold whenever $g_{l+1}(x_k) \leq 0$; that is, to satisfy $g_j(x_k) \leq g_{l+1}(x_k)$, $\forall j \in \{1, 2, \ldots, l\}$. Multiplying the term $g_j(x_k)$ by 2 to balance the cross-term leads to a sufficient condition that guarantees $x_k \in \Omega \subset \mathcal{X} \subset \mathbb{R}^n$. This condition can be compactly expressed as the following inequality, as established in [48]:

$$x_k^\top P x_k - 2a_j^\top x_k + 1 \geq 0, \quad \forall j \in \{1, 2, \ldots, l\}.$$

This inequality can be rewritten in its completed square form as:

$$\left(x_k - P^{-1}a_j\right)^\top P \left(x_k - P^{-1}a_j\right) + 1 - a_j^\top P^{-1} a_j \geq 0, \quad \forall j \in \{1, 2, \ldots, l\}.$$

Since the first term is always non-negative ($P \succ 0$), this inequality holds if and only if:

$$1 - a_j^\top P^{-1} a_j \geq 0, \quad \forall j \in \{1, 2, \ldots, l\},$$

which is similar, via Schur complement, to the following set of LMIs:

$$\begin{bmatrix} P & a_j \\ a_j^\top & 1 \end{bmatrix} \succeq 0, \quad \forall j \in \{1, 2, \ldots, l\}.$$

To express these LMIs in a form compatible with (4.9), using the variable substitutions $W = P^{-1}$ and $L = KP^{-1}$, a congruence transformation with $\mathrm{diag}(P^{-1}, I)$ is applied. This yields the following equivalent formulation:

$$\begin{bmatrix} W & Wa_j \\ a_j^\top W & 1 \end{bmatrix} \succeq 0, \quad \forall j \in \{1, 2, \ldots, l\}.$$

Minimizing the trace of P, whose eigenvalues are inversely proportional to the volume of the ellipsoidal region, yields the largest admissible terminal set $\Omega \subset \mathcal{X} \subset \mathbb{R}^n$. Using the substitution $W = P^{-1}$, the following equivalent convex optimization problem is formulated to simultaneously enforce the Lyapunov stability condition and the satisfaction of all l hyperplane constraints:

$$\mathcal{L}(A, B, Y, Z, a_1, \dots, a_l) : \begin{cases} \min\limits_{(W,\,L)} \ -\text{trace}(W) \\ \text{subject to:} \\ \quad W = W^\top \succ 0, \\ \quad \begin{bmatrix} W & AW + BL & 0 & 0 \\ \star & W & W & 0 \\ \star & \star & Y & L^\top \\ \star & \star & \star & Z \end{bmatrix} \succ 0, \\ \quad \begin{bmatrix} W & Wa_j \\ \star & 1 \end{bmatrix} \succeq 0, \quad \forall j \in \{1,\, 2,\, \dots,\, l\}, \end{cases} \tag{4.16}$$

where $Y = Q^{-1}$ and $Z = R^{-1}$.

If the optimization problem (4.16) is feasible, it yields the matrices $W \in \mathbb{R}^{n \times n}$ and $L \in \mathbb{R}^{m \times n}$. Recovering the Lyapunov matrix $P = W^{-1}$, the following terminal set can be defined:

$$\Omega := \left\{ x_k \in \mathcal{X} \;\middle|\; x_k^\top P x_k \leq 1 \right\}, \tag{4.17}$$

which is positively invariant under the terminal control law

$$\kappa_\Omega(x_k) := LPx_k. \tag{4.18}$$

It is important to observe that, in the definition of the terminal set (4.17), the system states are restricted to the admissible set $\mathcal{X} \subset \mathbb{R}^n$, in contrast to the definition provided in (4.10).

Internal Polyhedral Approximation of Ellipsoidal Sets

In the context of MPC and invariant set computations, it is often necessary to approximate nonlinear or curved sets—such as ellipsoids—by polytopes. This is particularly useful when compatibility with linear inequality constraints or polyhedral operations (e.g., intersections, containment tests, and feasibility sets) is required [49]. An internal polyhedral approximation ensures that all points of the approximating set lie strictly within the original ellipsoid, thus preserving feasibility and invariance properties essential for constrained control design.

Given the ellipsoidal set defined in (4.17), the objective is to construct a convex polyhedron $\Omega_{\text{ap}} \subset \Omega$ such that it remains entirely contained within the ellipsoid. This guarantees that every point in the approximated set satisfies the original ellipsoidal constraint, allowing its safe integration into predictive control formulations.

The construction of this internal approximation is based on the geometric principle of inscribing a polyhedron within the ellipsoid. This is achieved by evaluating the ellipsoidal boundary along a finite number of directions that are uniformly distributed over the unit sphere $\mathbb{S}^2 \subset \mathbb{R}^3$. The boundary points obtained from these directions are then used as the vertices of the approximating polyhedron. The process involves the following key steps:

1. **Uniform direction generation:** Generate a set of V approximately equally spaced unit vectors $\{d_i\}_{i=1}^{V} \subset \mathbb{S}^2$, representing directions in space. In this work, the Fibonacci lattice method [50] is employed to efficiently and uniformly distribute these points across the spherical surface, avoiding clustering effects and ensuring angular homogeneity.

2. **Boundary point computation:** For each direction vector d_i, compute a scaling factor $\alpha_i = \frac{1}{\sqrt{d_i^\top P d_i}}$ such that the resulting point $x_i = \alpha_i d_i$ lies exactly on the ellipsoidal boundary, i.e., $x_i^\top P x_i = 1$.

3. **Vertex set construction:** Collect the set of boundary points $\{x_i\}_{i=1}^{V}$ computed in the previous step. These points, by construction, lie on the surface of the ellipsoid and are thus eligible to serve as vertices of an inscribed polyhedron.

4. **Convex polyhedron formation:** Form the convex hull of the boundary point set to obtain the internal polyhedral approximation:

$$\Omega_{\text{ap}} := \text{convhull}(\{x_i\}_{i=1}^{V}), \quad \text{with } \Omega_{\text{ap}} \subset \Omega.$$

   Because all vertices lie on the ellipsoid surface, and convexity is preserved under the convex hull operation, the resulting polyhedron is fully contained within Ω, satisfying the required ellipsoidal constraint.

5. **Conversion to *H*-representation:** To enable compatibility with linear inequality formulations commonly used in MPC, the polyhedron Ω_{ap} can be equivalently represented in half-space (H-) representation as:

$$\Omega_{\text{ap}} := \left\{ x_k \in \mathcal{X} \mid H_{\Omega}\, x_k \leq b_{\Omega} \right\}, \tag{4.19}$$

   where $H_{\Omega} \in \mathbb{R}^{\eta \times 3}$, $b_{\Omega} \in \mathbb{R}^{\eta}$, and η is the number of supporting hyperplanes of the polyhedron.

Overall, this method provides a computationally efficient and geometrically reliable way to construct a conservative inner approximation of a three-dimensional ellipsoidal set. It is especially useful for applications in terminal set synthesis, feasibility analysis, and the computation of control-invariant sets within the broader scope of constrained predictive control.

MPC problem formulation

In summary, the proposed MPC design satisfies all conditions stated in Assumptions A1–A4. As a result, the terminal cost function (4.4), the inner polyhedral approximation of the terminal set (4.19), and the terminal control law (4.18) collectively guarantee recursive feasibility and asymptotic stability of the closed-loop system while ensuring compliance with all imposed state and input constraints, as defined by the polytopic sets in (4.13) and (4.14).

Based on these guarantees, the constrained MPC optimization problem (2.6) for the discrete-time LTI system (4.1) can be formally stated as follows:

$$
\mathcal{P}_N(x_k) : \begin{cases}
\min\limits_{\{\bar{x}_i, \bar{u}_i\}} \displaystyle\sum_{i=k}^{k+N-1} \left(|\bar{x}_i|_Q^2 + |\bar{u}_i|_R^2\right) + \left(|\bar{x}_{k+N}|_P^2\right) \\
\text{subject to:} \\
\quad \bar{x}_k = x_k, \\
\quad \bar{x}_i^+ = A\bar{x}_i + B\bar{u}_i, \quad i = k, \dots, k+N-1, \\
\quad \begin{bmatrix} I \\ -I \end{bmatrix} \bar{x}_i \leq \begin{bmatrix} x_{\text{ub}} \\ -x_{\text{lb}} \end{bmatrix}, \quad i = k+1, \dots, k+N, \\
\quad \begin{bmatrix} I \\ -I \end{bmatrix} \bar{u}_i \leq \begin{bmatrix} u_{\text{ub}} \\ -u_{\text{lb}} \end{bmatrix}, \quad i = k, \dots, k+N-1, \\
\quad H_\Omega \, \bar{x}_{k+N} \leq b_\Omega,
\end{cases} \tag{4.20}
$$

where $Q \succ 0$ and $R \succ 0$ are symmetric positive definite weighting matrices associated with the state and control input, respectively, and $P = W^{-1}$ denotes the terminal penalty matrix, obtained by solving the optimization problem (4.16). Assigning large values to Q relative to R prioritizes fast convergence of the system states to the origin, potentially at the cost of aggressive control inputs. Conversely, increasing the weights in R relative to Q imposes a stronger penalty on control effort, resulting in smoother control inputs and a slower rate of convergence to the desired equilibrium [34].

4.1.4 Approximation of the Domain of Attraction

A key concept in approximating the domain of attraction is the one-step set, which comprises all states that can be driven into a given terminal set in a single step of control, respecting all state and input constraints. For the system model defined in (4.1), the one-step set associated with a terminal set Ω is defined as:

$$
\mathcal{Q}(\Omega) := \left\{ x_k \in \mathcal{X} \;\middle|\; \exists u_k \in \mathcal{U} : x_k^+ \in \Omega \right\},
$$

where $\mathcal{X}$ and $\mathcal{U}$ denote the admissible sets of states and inputs, respectively.

In this study, the domain of attraction is approximated by iteratively enlarging the terminal set through a contractive sequence of one-step sets. This is achieved by replacing the standard terminal constraint with a contractive terminal constraint, which is defined by a sequence of sets computed offline. At each iteration, the set is expanded by including the states that can be reached from the previous set in one step of control while satisfying all constraints. This recursive expansion continues until convergence, yielding a larger invariant set that enhances the feasibility region of the MPC [36].

However, computing one-step sets can be intractable for high-dimensional systems. To address this, the one-step operator $\mathcal{Q}(\cdot)$ can be replaced by a computationally tractable inner approximation $\mathcal{Q}_{\text{ap}}(\cdot) \subset \mathcal{Q}(\cdot)$, as suggested in [36]. Based on this inner approximation and the initial terminal set $\Omega_0 = \Omega$, an estimate of the domain of attraction, $\mathcal{D}_N(\Omega)$, is constructed as the union of a contractive sequence of reachable sets defined by the recursion:

$$\Omega_i = \mathcal{Q}_{\text{ap}}(\Omega_{i-1}) \cap \mathcal{X}, \quad \text{with } \Omega_0 = \Omega. \tag{4.21}$$

This sequence of sets satisfies the nested inclusion property:

$$\Omega_0 \subset \Omega_1 \subset \cdots \subset \Omega_i \subset \mathcal{X}, \quad \forall i.$$

This procedure provides a scalable and systematic method for approximating the domain of attraction, enabling its integration into controller synthesis to enhance robustness and feasibility margins.

The inner approximation $\mathcal{Q}_{\text{ap}}(\Omega)$ is computed using the algorithm proposed in [51], which iteratively refines a collection of state-input boxes. The algorithm starts with an initial list L of boxes spanning the admissible regions $\mathcal{X} \times \mathcal{U}$. At each iteration, the box with the largest diameter is selected and tested: if its image under the system dynamics lies entirely within Ω, its state projection is added to the approximation; if it lies completely outside, it is discarded. If the result is uncertain or the box exceeds a resolution threshold ε, it is bisected along its largest dimension, and the resulting sub-boxes are queued for further evaluation. This process continues until all boxes are either included, discarded, or refined below ε, resulting in a union of boxes that forms an inner approximation of the one-step set.

4.1.5 Setpoint Tracking Strategy

Setpoint tracking refers to the process of steering a system's measurable outputs toward a specific and constant reference value—often referred to as setpoint [34]. In the model described in (4.1), the output vector y_k contains all measurable signals from the system. However, in many control applications, only a subset of these outputs is required to track setpoints. To address this, a reduced output vector z_k is defined by selecting the relevant components of y_k. As shown in (4.22), the matrix E extracts these components, thereby defining the controlled variables z_k that are regulated by the MPC to perform the setpoint tracking.

$$z_k = Ey_k \quad \Rightarrow \quad z_k = ECx_k \tag{4.22}$$

At steady state, it holds that $x_k^+ = x_k = x_\mathrm{s}$ for some corresponding control action u_s. Substituting these steady-state values into (4.1) and (4.22) yields the set of linear equations given in (4.23), where the controlled output z_k is expected to match a given constant reference.

$$\begin{bmatrix} I - A & -B \\ EC & 0 \end{bmatrix} \begin{bmatrix} x_\mathrm{s} \\ u_\mathrm{s} \end{bmatrix} = \begin{bmatrix} 0 \\ z_\mathrm{ref} \end{bmatrix} \tag{4.23}$$

If a solution to (4.23) exists, the following deviation variables can be defined:

$$\begin{aligned} \tilde{x}_k &= x_k - x_\mathrm{s}, \\ \tilde{u}_k &= u_k - u_\mathrm{s}, \end{aligned} \tag{4.24}$$

which satisfy the deviation dynamics:

$$\tilde{x}_k^+ = A\tilde{x}_k + B\tilde{u}_k. \tag{4.25}$$

The control objective then becomes finding a sequence of deviation inputs $\tilde{u}_k$ that drives the deviation state $\tilde{x}_k$ to the origin in (4.25), which corresponds to reaching the desired steady-state point $(x_\mathrm{s}, u_\mathrm{s})$ in the original coordinate space. This steady-state pair is obtained by solving the optimization problem defined in (4.26), where the tracking weight matrices Q_s and R_s are real, symmetric, and satisfy $Q_\mathrm{s} \succeq 0$ and $R_\mathrm{s} \succ 0$ [34].

$$\mathcal{R}(z_\mathrm{ref}) : \begin{cases} \min\limits_{(x_\mathrm{s}, u_\mathrm{s})} \left(|u_\mathrm{s}|_{R_\mathrm{s}}^2 + |Cx_\mathrm{s}|_{Q_\mathrm{s}}^2 \right) \\ \text{subject to:} \\ \quad \begin{bmatrix} I - A & -B \\ EC & 0 \end{bmatrix} \begin{bmatrix} x_\mathrm{s} \\ u_\mathrm{s} \end{bmatrix} = \begin{bmatrix} 0 \\ z_\mathrm{ref} \end{bmatrix}, \\ \quad \begin{bmatrix} I \\ -I \end{bmatrix} x_\mathrm{s} \leq \begin{bmatrix} x_\mathrm{ub} \\ -x_\mathrm{lb} \end{bmatrix}, \\ \quad \begin{bmatrix} I \\ -I \end{bmatrix} u_\mathrm{s} \leq \begin{bmatrix} u_\mathrm{ub} \\ -u_\mathrm{lb} \end{bmatrix}. \end{cases} \tag{4.26}$$

Once the steady-state pair $(x_\mathrm{s}, u_\mathrm{s})$ is obtained, a coordinate transformation is performed according to (4.24). The resulting constrained optimization problem (4.20) is then solved for $\mathcal{P}(\tilde{x}_k)$, where all constraints are reformulated in the deviation coordinates. From the optimal input sequence obtained, only the first control input, $\tilde{u}_k$, is applied to the system. The actual control input is subsequently recovered by applying the inverse transformation:

$$u_k = \tilde{u}_k + u_\mathrm{s}. \tag{4.27}$$

4.2 MHE Implementation

The objective of an estimator is to reconstruct the true state of the system as accurately as possible. This involves minimizing the mismatch between the system's dynamic model and the observed measurements. Figure 4.1 illustrates a structure in which the plant represents the true plant dynamics, while the estimator replicates the plant's behavior to compute state estimates $\hat{x}_k$, using the known inputs u_k and outputs y_k.

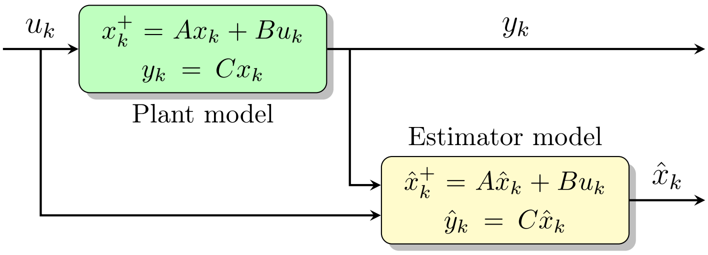


Figure 4.1: Representation of the MHE connection structure.

The cost function of the MHE problem is derived by analyzing the differences between the plant and estimator equations. These differences define the prediction errors, which are penalized in the optimization process. The state prediction error is obtained by the difference between the state equations of the plant and the estimator, as follows:

$$e_x := \begin{cases} x_k^+ = Ax_k + Bu_k \\ -\hat{x}_k^+ = -A\hat{x}_k - Bu_k \end{cases} \Rightarrow \quad e_x := x_k^+ - Ax_k = \hat{x}_k^+ - A\hat{x}_k.$$

Similarly, for the output prediction error:

$$e_y := \begin{cases} y_k = Cx_k \\ -\hat{y}_k = -C\hat{y}_k \end{cases} \Rightarrow \quad e_y := y_k - Cx_k = \hat{y}_k - C\hat{x}_k.$$

As the control input u_k and measurements y_k are known, the predicted output is typically set to match the measured value, i.e., $\hat{y}_k = y_k$. Note that u_k does not influence the estimation directly, since it is equally applied to both plant and estimator.

Based on these relationships, the MHE optimization problem (2.7) for the discrete-time LTI system (4.1) can be formally stated as follows:

$$\mathcal{E}_M\big(\{y_i,u_i\}\big):\begin{cases}\min\limits_{\{\hat{x}_i\}}\sum\limits_{i=k-M+1}^{k-1}\big|\hat{x}_i^+ - A\hat{x}_i\big|_{Q_e}^2+\sum\limits_{i=k-M+1}^{k}|y_i-C\hat{x}_i|_{R_e}^2\\ \text{subject to:}\\ \quad\hat{x}_i^+ = A\hat{x}_i+Bu_i,\quad i=k-M+1,\dots,k-1,\\ \quad y_i = C\hat{x}_i,\quad i=k-M+1,\dots,k,\\ \quad\begin{bmatrix}I\\-I\end{bmatrix}\hat{x}_i\le\begin{bmatrix}x_{\text{ub}}\\-x_{\text{lb}}\end{bmatrix},\quad i=k-M+1,\dots,k,\end{cases}\tag{4.28}$$

where the arrival cost $V(\hat{x}_{k-M+1})$ is assumed to be zero, thereby discarding all information before the estimation horizon. Under this formulation, the MHE must be able to asymptotically reconstruct the system state using only the most recent M measurements [34].

4.3 Fault Residual Estimation

Let A_0 and B_0 denote the nominal (fault-free) system matrices of the model (3.16), defined as $A_0 = A(f_k)\big|_{f_k=0}$ and $B_0 = B(f_k)\big|_{f_k=0}$. The occurrence of faults induces deviations in the system dynamics, which can be captured through a fault residual vector $\xi_k\in\mathbb{R}^n$, defined as:

$$\xi_k=\big(A(f_k)-A_0\big)x_k+\big(B(f_k)-B_0\big)u_k.\tag{4.29}$$

Based on this definition, the fault-affected system can be rewritten in the following equivalent form:

$$x_k^+ = A_0x_k+B_0u_k+\xi_k.\tag{4.30}$$

It is straightforward to verify that substituting (4.29) into (4.30) yields the original fault-inclusive model (3.16).

Assuming that the fault residuals evolve slowly and can be approximated as constant over short horizons (i.e., $\xi_k^+=\xi_k$), the model can be augmented by treating the fault residuals as additional states, as shown below:

$$\begin{aligned}\begin{bmatrix}x_k^+\\ \xi_k^+\end{bmatrix}&=\begin{bmatrix}A_0 & I\\ 0 & I\end{bmatrix}\begin{bmatrix}x_k\\ \xi_k\end{bmatrix}+\begin{bmatrix}B_0\\ 0\end{bmatrix}u_k\\ y_k&=\begin{bmatrix}C_0 & 0\end{bmatrix}\begin{bmatrix}x_k\\ \xi_k\end{bmatrix}\end{aligned}\tag{4.31}$$

The MHE optimization problem (4.28) can be adapted to estimate the system states and fault residuals simultaneously:

$$\mathcal{E}_M\big(\{y_i, u_i\}\big) : \begin{cases} \displaystyle \min_{\{\hat{x}_i, \hat{\xi}_i\}} \sum_{i=k-M+1}^{k-1} \left| \mathring{x}_i^+ - \mathring{A}\mathring{x}_i - \mathring{B}u_i \right|_{Q_e}^2 + \sum_{i=k-M+1}^{k} \left| y_i - \mathring{C}\mathring{x}_i \right|_{R_e}^2 \\ \text{subject to:} \\ \quad \mathring{x}_i^+ = \mathring{A}\mathring{x}_i + \mathring{B}u_i, \quad i = k-M+1, \dots, k-1, \\ \quad y_i = \mathring{C}\mathring{x}_i, \quad i = k-M+1, \dots, k, \\ \quad \begin{bmatrix} I \\ -I \end{bmatrix} \hat{x}_i \leq \begin{bmatrix} x_{\text{ub}} \\ -x_{\text{lb}} \end{bmatrix}, \quad i = k-M+1, \dots, k, \end{cases} \tag{4.32}$$

where $(\mathring{*})$ denotes variables and matrices associated with the augmented model (4.31), and the input arguments $\{u_i, y_i\}$ represent the sequences of control inputs and measurements from $i = k - M + 1$ to k.

4.4 Control Reconfiguration

Once the estimates $\hat{\xi}_k$ are obtained, they can be used to relate the actual system matrices with the nominal model as:

$$A(f_k)x_k + B(f_k)u_k = A_0 x_k + B_0 u_k + \xi_k \tag{4.33}$$

Using the same moving horizon window as the MHE, the most recent M state and control input samples are collected into the matrix

$$Z = \begin{bmatrix} X^\top \\ U^\top \end{bmatrix},$$

where $X \in \mathbb{R}^{M\times n}$ and $U \in \mathbb{R}^{M\times m}$. The fault residual estimates are represented by the matrix $\Xi \in \mathbb{R}^{n\times M}$. Based on (4.33), the fault-affected system matrices are estimated as follows:

$$\begin{bmatrix} \hat{A}(f_k) & \hat{B}(f_k) \end{bmatrix} = \begin{bmatrix} A_0 & B_0 & I \end{bmatrix} \begin{bmatrix} Z \\ \Xi \end{bmatrix} Z^\dagger,$$

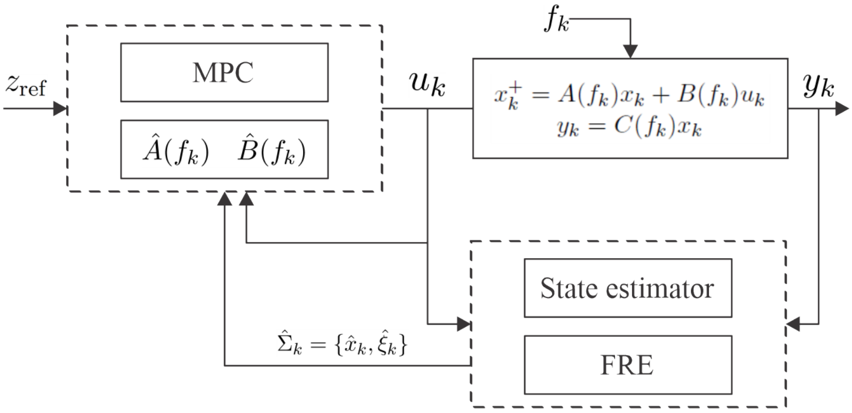
where $Z^\dagger$ denotes the pseudo-inverse of Z.

The estimated matrices $\hat{A}(f_k)$ and $\hat{B}(f_k)$ capture the altered dynamics of the system under faults and can be directly used to update the internal model of the MPC controller. By incorporating these updated matrices into the prediction model, the controller adapts its decisions to compensate for the fault, ensuring continued performance and constraint satisfaction.

This procedure defines a Fault-Tolerant MPC (FTMPC) scheme, where the predictive model is reconfigured online based on real-time estimates of system degradation. The

MPC optimization problem is solved at each time step using the updated model, and the control action is computed accordingly.

Figure 4.2 illustrates the overall proposed architecture in a closed-loop. The system is subject to potential faults f_k, which change its nominal behavior. The Fault Residual Estimator (FRE) module processes the measured outputs and the previous control inputs to estimate the fault residual signals ξ_k, from which the modified matrices $\hat{A}(f_k)$ and $\hat{B}(f_k)$ are obtained. These matrices are then used by the MPC module to compute the optimal control input u_k that compensates for the fault effects and maintains reference tracking around the desired setpoint z_{ref}, ensuring stability and satisfaction of the state and input constraints.



f_k
z_{ref}
MPC
$\hat{A}(f_k)$ $\hat{B}(f_k)$
u_k
$x_k^+ = A(f_k)x_k + B(f_k)u_k$
$y_k = C(f_k)x_k$
y_k
State estimator
FRE
$\hat{\Sigma}_k = \{\hat{x}_k, \hat{\xi}_k\}$

Figure 4.2: Proposed closed-loop architecture integrating the fault residual estimator and the control readjustment model to perform fault accommodation.

4.5 Simulation Framework

This section details the simulation setup employed to assess the performance of the proposed AFTC framework. It describes the modeling tools and computational environment, the system parameters and configuration, and the specific design settings adopted for the MPC-based controller and the MHE-based estimator.

4.5.1 Simulation Environment

All simulations were carried out using `MATLAB R2021b` [52] on a desktop computer equipped with an `Intel Core i7` processor (2.90 GHz) and `16 GB` of RAM. The following tools were employed to implement and test the proposed control and estimation framework:

- `Sim3Tanks`: a benchmark simulator for the three-tank system, developed to emulate hybrid and nonlinear dynamics under external disturbances and faults, including actuator and sensor malfunctions, as well as plant-level anomalies such as clogging and leakage [47].

- YALMIP: a modeling toolbox for formulating optimization problems, used to define the quadratic programming problems in MPC and MHE [53].
- SeDuMi: a solver for quadratic and semidefinite programming problems involved in controller and observer design [54].
- MPT3: a toolbox for parametric optimization, computational geometry, and MPC problems [55].

4.5.2 System Configuration and Physical Parameters

The simulated plant corresponds to the default case study illustrated in Figure 3.2, in which the inlet flows to the tanks T_1 and T_2 are regulated through the control valves K_{P_1} and K_{P_2}, respectively. The remaining valves (K_{13}, K_{23}, and K_3) are considered to be constantly open and are not subject to control actions. Table 4.1 summarizes the physical and structural parameters of the three-tank system used throughout the simulations.

Table 4.1: Physical parameters of the three-tank system.

Parameter	**Value**	**Unit**	**Parameter**	**Value**	**Unit**
Tank radius (R)	5	cm	Flow correction factor (μ)	1	–
Tank height ($h_{\max}$)	50	cm	Gravity constant (g)	981	cm/s^2
Pipe radius (r)	0.6	cm	Min. pump flow ($Q_{\min}$)	0	cm^3/s
Trans. pipe height (h_0)	30	cm	Max. pump flow ($Q_{\max}$)	120	cm^3/s

All simulations are based on the discrete-time LTI faulty model defined in (3.16), with a sampling time of $\boldsymbol{T_s = 0.1}$ seconds. This model explicitly accounts for the faults listed in Table 3.2, including actuator, sensor, and plant-level anomalies.

To ensure consistency with the equilibrium conditions presented in (3.15), the following steady-state operating point was adopted:

$$x_{\mathrm{op}} = \begin{bmatrix} 10 \\ 10 \\ 8 \end{bmatrix}, \quad u_{\mathrm{op}} = \begin{bmatrix} 0.5904 \\ 0.5904 \end{bmatrix}.$$

This operating point represents a symmetric flow configuration where the lateral tanks maintain equal fluid levels, and the central tank stabilizes at a lower height, ensuring nonzero flow across all interconnecting pipes.

4.5.3 MPC and MHE Setup

The configuration parameters used in the implementation of the optimization problems presented in (4.20), (4.26), and (4.32) are described below. These formulations correspond to the linear constrained MPC, the steady-state computation for setpoint tracking, and the augmented MHE used for fault residual estimation, respectively.

MPC for State Regulation — Optimization Problem (4.20): the linear constrained MPC problem was configured with the following parameters:

- The prediction horizon was defined as $N = 5$.
- The weighting matrices Q and R were initially selected to normalize the magnitudes of the state and input controls. To promote smoother control actions, the values in R were subsequently increased to impose a stronger penalty on control effort. As a result, the final weighting matrices were defined as

$$\begin{aligned} Q &= \mathrm{diag}(10^{-4},\, 10^{-4},\, 10^{-4}), \\ R &= \mathrm{diag}(10^{-1},\, 10^{-1}). \end{aligned}$$

- The terminal cost matrix was obtained by solving the convex optimization problem defined in (4.16), which resulted in the matrix

$$P = \begin{bmatrix} 0.0100 & 0.0003 & -0.0002 \\ 0.0003 & 0.0100 & -0.0002 \\ -0.0002 & -0.0002 & 0.0156 \end{bmatrix}. \tag{4.34}$$

- The state constraints were defined based on the deviation of the tank levels relative to the operating point:

$$\begin{aligned} x_{\mathrm{ub}}^{\top} &= [h_{\max},\, h_{\max},\, h_{\max}]^{\top} - x_{\mathrm{op}}^{\top}, \\ x_{\mathrm{lb}}^{\top} &= [0,\, 0,\, 0]^{\top} - x_{\mathrm{op}}^{\top}. \end{aligned} \tag{4.35}$$

- Similarly, the input constraints were defined according to the capacity of the opening of the valves:

$$\begin{aligned} u_{\mathrm{ub}}^{\top} &= [1,\, 1]^{\top} - x_{\mathrm{op}}^{\top}, \\ u_{\mathrm{lb}}^{\top} &= [0,\, 0]^{\top} - u_{\mathrm{op}}^{\top}. \end{aligned} \tag{4.36}$$

- The terminal region Ω_{10} was computed by iteratively expanding the terminal set Ω over 10 steps using the procedure described in (4.21). As a result, the domain of attraction was defined as $\mathcal{D}_5(\Omega_{10}) := \mathcal{D}_{15}(\Omega)$.

MPC for Setpoint Tracking — Optimization Problem (4.26): to compute the steady-state reference pair $(x_{\mathrm{s}}, u_{\mathrm{s}})$ associated with a given setpoint z_{ref}, the following parameters were used:

- The output tracking penalty matrix was defined as

  $$Q_\mathrm{s} = \mathrm{diag}(10^{-1}, 10^{-1}, 10^{-3}, 10^{-3}, 10^{-4}),$$

  assigning higher weights to critical outputs while allowing flexibility in less sensitive variables.

- The input tracking penalty matrix was selected as

  $$R_\mathrm{s} = \mathrm{diag}(10, 10),$$

  to discourage aggressive control actions and promote smoother steady-state inputs.

- The output selection matrix was defined as

  $$E = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

  which selects the outflow Q_3 as the variable to be tracked by the controller.

- The state and input constraints follow the same bounds established for the MPC regulation setup, as given in (4.35) and (4.36).

MHE for State and Fault Residual Estimation — Optimization Problem (4.32): the augmented MHE problem used for estimating the state and fault residual vector ξ_k was configured with the following parameters:

- The estimation horizon was set to $M = 20$, which provides sufficient memory depth to reconstruct the system state trajectory and detect persistent fault signatures.

- The process noise weight matrix was defined as

  $$Q_e = \mathrm{diag}(10,\ 10,\ 10,\ 10,\ 10,\ 10),$$

  balancing sensitivity to deviations in the state and the evolution of fault residuals.

- The measurement noise weight matrix was selected as

  $$R_e = \mathrm{diag}(10,\ 10,\ 10,\ 10,\ 10).$$

- The state constraints x_lb and x_ub were inherited from the MPC formulation, as defined in (4.35), ensuring consistency with the physical operating limits of the system.

This formulation allows the real-time estimation of fault effects by augmenting the system dynamics with a fault residual vector ξ_k. The residual estimates are used in the online reconstruction of the faulty system matrices $\hat{A}(f)$ and $\hat{B}(f)$ to adapt the MPC controller accordingly.

Chapter 5

Results

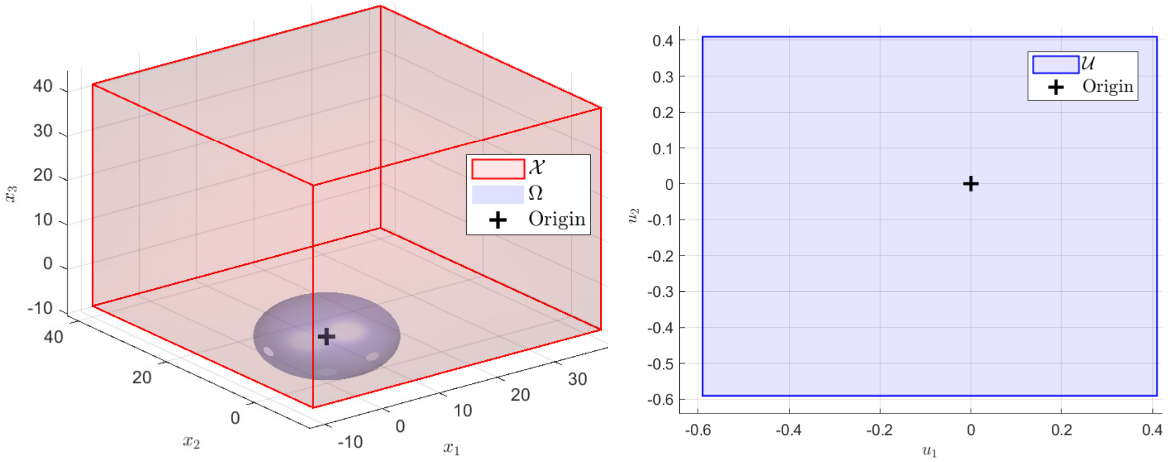
This chapter presents the simulation results obtained with the proposed fault-tolerant control framework, which integrates Model Predictive Control (MPC) and Moving Horizon Estimation (MHE). The aim is to assess the effectiveness of the integrated strategy in accommodating faults and maintaining setpoint tracking in the presence of both actuator and plant faults in the three-tank system. The results are organized into three main categories: nominal operation, actuator fault accommodation, and plant fault accommodation.

5.1 Admissible and Terminal Sets Visualization

Before evaluating the control performance under nominal and faulty conditions, this section presents a geometric visualization of the sets that define the feasible operating region of the closed-loop system. These include the admissible input and state constraints, the computed terminal set, and the domain of attraction approximation. All sets are constructed based on the parameter values described in Section 4.5.

Figure 5.1 illustrates the admissible region defined by box constraints on the state and control input deviated from the selected operating point $(x_{\text{op}}, u_{\text{op}})$. The state constraints are imposed on the tank levels and follow the bounds given in (4.35), whereas the control constraints reflect the physical limits of the actuated valves, as shown in (4.36). These constraints are encoded as H-polyhedra, in the form of (4.11)–(4.12), and are enforced at each time step of the MPC prediction horizon to guarantee that all predicted trajectories remain within the feasible and safe operating region around the equilibrium point. In addition, Figure 5.1a depicts the ellipsoidal terminal set Ω defined by terminal penalty matrix (4.34), which is fully contained within the admissible state set $\mathcal{X}$.

Figure 5.2a presents the inner polyhedral approximation of the terminal set, as defined in (4.19). As illustrated, all states contained in this set respect the admissible constraints and can be driven to the origin via a local linear state-feedback controller, thereby ensuring recursive feasibility and closed-loop asymptotic stability. In turn, Figure 5.2b shows a projection of the polyhedral approximation of the domain of attraction $\mathcal{D}_5(\Omega_{10}) := \mathcal{D}_{15}(\Omega)$

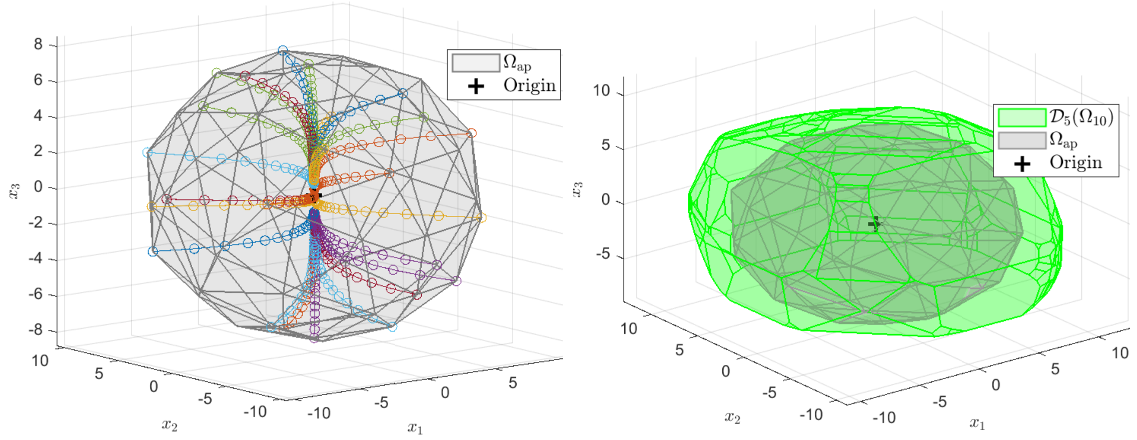


(a) State and terminal set $\Omega \subset \mathcal{X} \subset \mathbb{R}^3$.

(b) Control input set $\mathcal{U} \subset \mathbb{R}^2$.

Figure 5.1: Polyhedral sets of admissible state and input deviations relative to $(x_{\text{op}}, u_{\text{op}})$.

obtained by a recursive sequence of one-step admissible sets, as described in (4.21). The inner polyhedron represents the terminal set Ω_{ap}, while the outer set corresponds to the expanded region from which the terminal set is reachable under admissible inputs.



(a) Inner terminal set $\Omega_{\text{ap}} \subset \Omega \subset \mathcal{X} \subset \mathbb{R}^3$.

(b) Domain of attraction $\mathcal{D}_5(\Omega_{10})$.

Figure 5.2: Polyhedral representation of the terminal set and the domain of attraction.

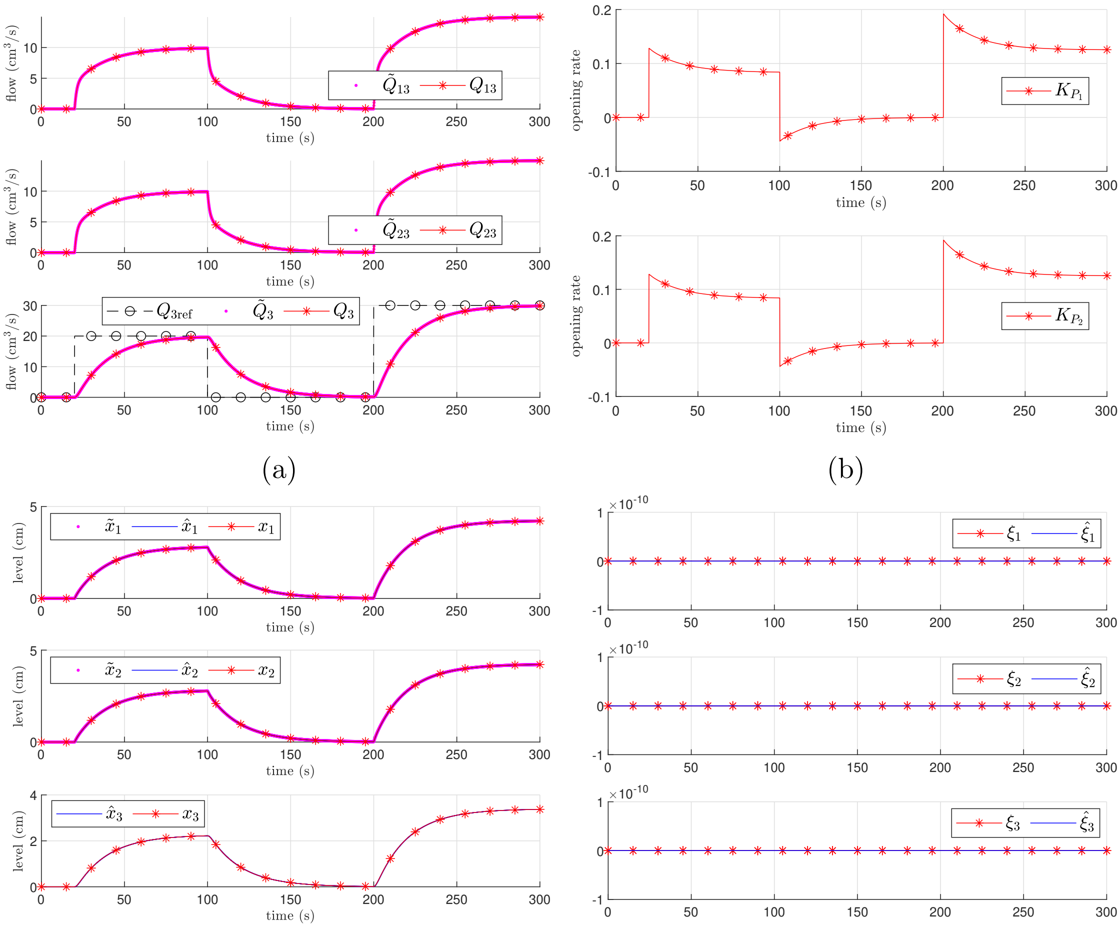
These geometric constructs serve as the foundation for constraint enforcement and feasibility guarantees in the MPC formulation. During operation, all predicted trajectories are required to remain within the admissible set, and the final state must lie inside the terminal set. In the presence of faults, the controller dynamically updates its internal model and continues enforcing these constraints to maintain safe and stable operation.

The next sections evaluate the closed-loop behavior of the system under nominal and faulty conditions, using the control architecture designed based on these constraint sets.

5.2 Nominal Operation Scenario

In the first scenario, no faults are introduced in the system. The FTMPC is initialized with the nominal model and receives state feedback from the FRE module, as illustrated in Figure 4.2. The control objective is to regulate the outlet flow rate Q_3 to a predefined setpoint $Q_{3\text{ref}}$ while satisfying state and input constraints.

Figure 5.3 presents the closed-loop system response under these nominal operating conditions, where $(\tilde{*})$ and $(\hat{*})$ denote, respectively, the measured and estimated values of the corresponding variable. This notation is used consistently throughout this chapter. In Figure 5.3a, the output trajectory closely follows the reference signal with negligible steady-state error, exhibiting the effectiveness of the control strategy. Figure 5.3b shows the corresponding control inputs, while Figures 5.3c and 5.3d depict the state and the fault residual estimations from the FRE module, respectively. Both the inputs and states remain within their admissible bounds, exhibiting smooth transitions without signs of saturation or instability. Overall, these results illustrate the nominal performance of the proposed control framework before the occurrence of any fault.



(a)

(b)

(c)

(d)

Figure 5.3: Setpoint tracking of the controlled variable Q_3 (a), control inputs (b), system states (c), and fault residual estimation (d) under nominal conditions.

5.3 Comparative Evaluation Methodology for Fault Scenarios

To ensure a consistent and objective comparison across different fault scenarios, three controller configurations are evaluated for each case:

1. **Nominal Model $\left(A_0,\, B_0\right)$:** the FTMPC operates with the nominal, fault-free model. No reconfiguration is applied after the fault occurs, leaving the model–plant mismatch uncorrected.

2. **Exact Fault Model $\left(A(f_k),\, B(f_k)\right)$:** the FTMPC is provided with the exact faulty system model, as if the fault magnitude were perfectly known. This represents the upper bound of achievable performance.

3. **Estimated Fault Model $\left(\hat{A}(f_k),\, \hat{B}(f_k)\right)$ – Proposed Approach:** the FTMPC is reconfigured online using the model estimated by the MHE-based Fault Residual Estimator (FRE) module. This case reflects the actual performance of the proposed AFTC framework.

Evaluation Metrics

For each scenario, the following quantitative metrics are computed to evaluate and compare the closed-loop performance after the occurrence of a fault:

- ***Post-Fault* Root Mean Square Error ($\mathbf{RMSE_{post}}$):** the $\text{RMSE}_{\text{post}}$ quantifies the tracking performance after the occurrence of a fault and is defined as

$$\text{RMSE}_{\text{post}} = \sqrt{\frac{1}{N_{\text{post}}} \sum_{i=1}^{N_{\text{post}}} (y_i - r_i)^2},$$

  where y_i is the controlled output, r_i is the reference signal, and N_{post} is the number of samples within the fault duration. All variables in this expression refer exclusively to the time interval between the fault initiation and its clearance within the simulation.

- **Steady-State Error (SSE):** the mean absolute tracking error, $|y_k - r_k|$, is computed over a fixed-length steady-state window of $N_{\text{SSE}} = 200$ samples preceding the end of the simulation. SSE quantifies the long-term tracking accuracy once transient effects have decayed, and is given by

$$\text{SSE} = \frac{1}{N_{\text{SSE}}} \sum_{i=1+N_{\text{last}}-N_{\text{SSE}}}^{N_{\text{last}}} |y_i - r_i|,$$

  where N_{last} is the index of the final simulation sample.

- **Estimation Delay:** defined as the elapsed time from the fault occurrence t_{fault} to the first instant where the absolute estimation error $|\hat{\xi}_k - \xi_k|$ enters and remains within a predefined tolerance band of ±2% of $\max|\xi_k|$ for at least a dwell time of 3 seconds. This metric quantifies how quickly the FRE converges to a reliable fault residual estimate.

- **Recovery Time:** defined as the elapsed time from t_{fault} to the first instant where the absolute tracking error $|y_k - r_k|$ enters and remains within a tolerance band of ±4% track for at least a settling dwell time of 8 seconds. This measures how quickly the closed-loop system regains acceptable tracking performance after a fault.

Graphical Representation

For each fault type and controller configuration, the results are displayed in two aligned panels:

- **Left panel:** controlled output trajectory along with the reference signal and control input trajectories. The fault period is highlighted with a transparent red background.

- **Right panel:** injected fault signal, corresponding fault residual estimation, and the actual fault trajectory.

Tabular Comparison

For each fault type, a table summarizes the four performance metrics for the three controller configurations.

5.4 Actuator Fault: Loss of Effectiveness in K_{P_1}

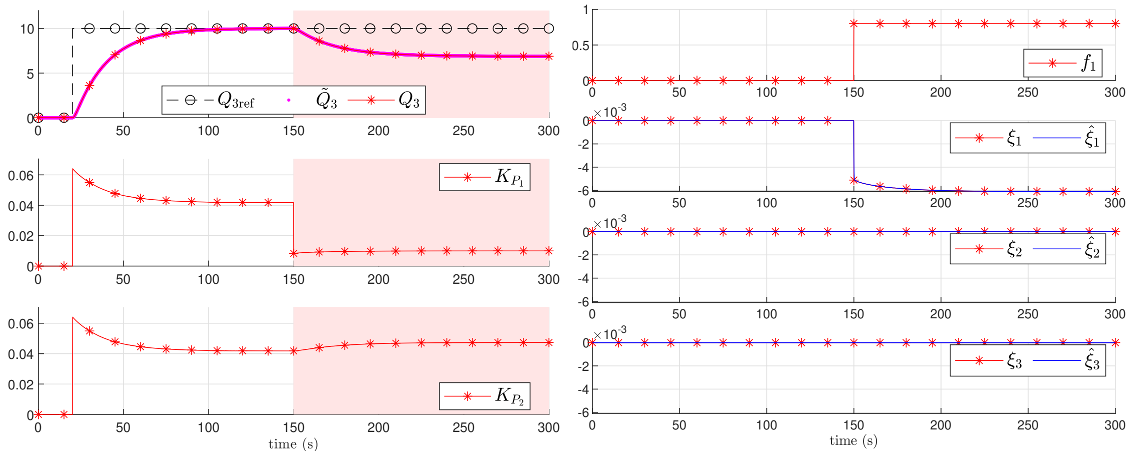
Fault description: At around $t = 150$ s, an abrupt loss of effectiveness of 80% ($f_1 = 0.8$) is introduced in the actuated valve K_{P_1}.

Results: The results are presented in Figures 5.4–5.6, while the corresponding performance metrics are summarized in Table 5.1.

- Nominal Model: unrecovered tracking (infinite recovery time), long fault estimation time, and high tracking error values.

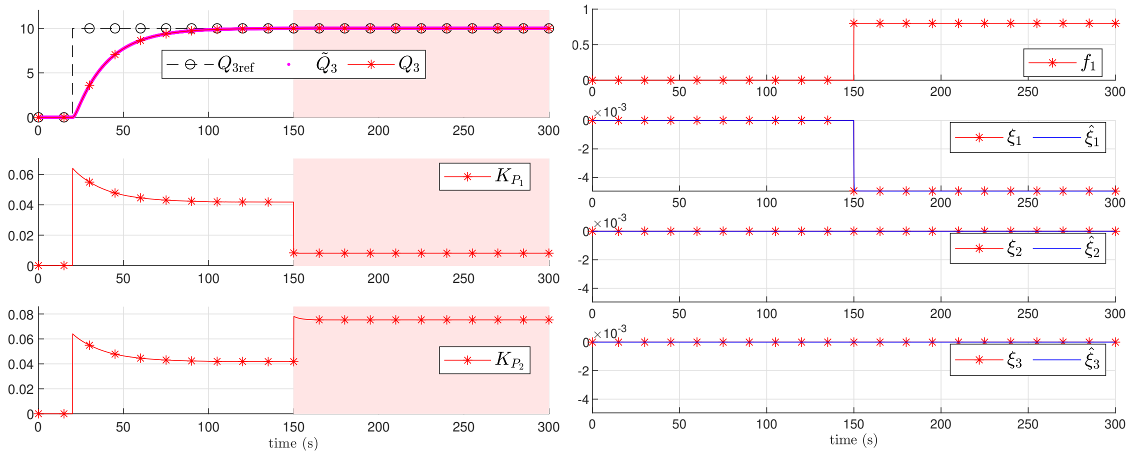
- Exact Fault Model: immediate recovery with negligible tracking error and no actuator saturation.

- Estimated Model: slight performance degradation, characterized by a low RMSE and a small offset error ($\sim 0.6\,\text{cm}^3/\text{s}$), along with a short estimation delay ($\sim 1.4\,\text{s}$).



(a) (b)

Figure 5.4: Closed-loop results using the $\underline{\text{nominal model}}$ $(A_0,\ B_0)$ under an abrupt actuator fault with a maximum magnitude of $f_1 = 0.8$.



(a) (b)

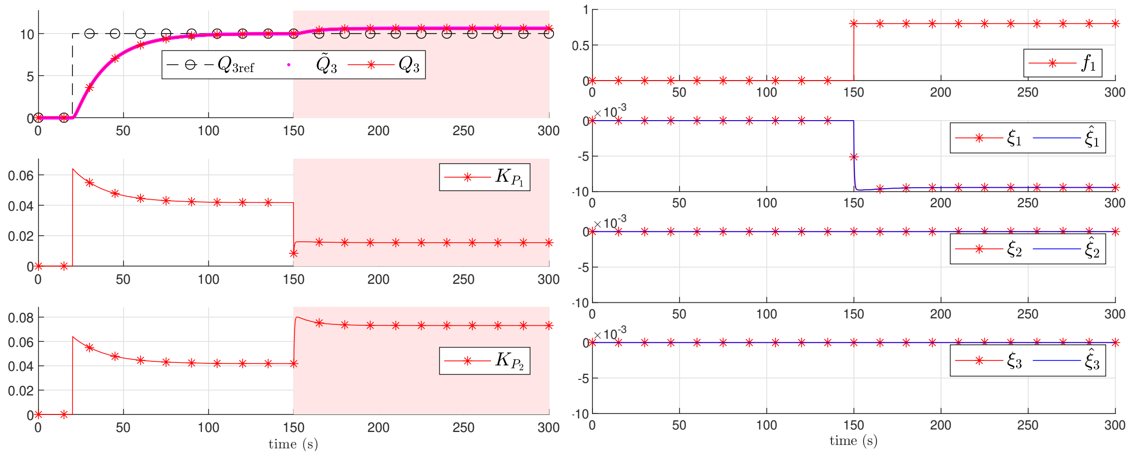
Figure 5.5: Closed-loop results using the $\underline{\text{exact model}}$ $(A(f_k),\ B(f_k))$ under an abrupt actuator fault with a maximum magnitude of $f_1 = 0.8$.

Table 5.1: Closed-loop performance metrics under an $\underline{\text{abrupt actuator fault}}$ (loss of effectiveness) initiated at around $t = 150$ s with a maximum magnitude of $f_1 = 0.8$.

Model	**RMSE$_{\text{post}}$ (cm^3/s)**	**SSE (cm^3/s)**	**Estimation Delay (s)**	**Recovery Time (s)**
Nominal	2.7285	3.1154	44.1	∞
Exact	0.0128	0.0061	0.4	0
Estimated	0.5761	0.6188	1.4	0

5.5 Plant Fault: Blocking in flow Q_{13}

Fault description: At approximately $t = 85$ s, an incipient blockage begins to develop in flow Q_{13}, progressively increasing until it reaches a 90% blockage ($f_6 = 0.9$). In practical



(a) (b)

Figure 5.6: Closed-loop results using the estimated model $\big(\hat{A}(f_k),\ \hat{B}(f_k)\big)$ under an abrupt actuator fault with a maximum magnitude of $f_1 = 0.8$.

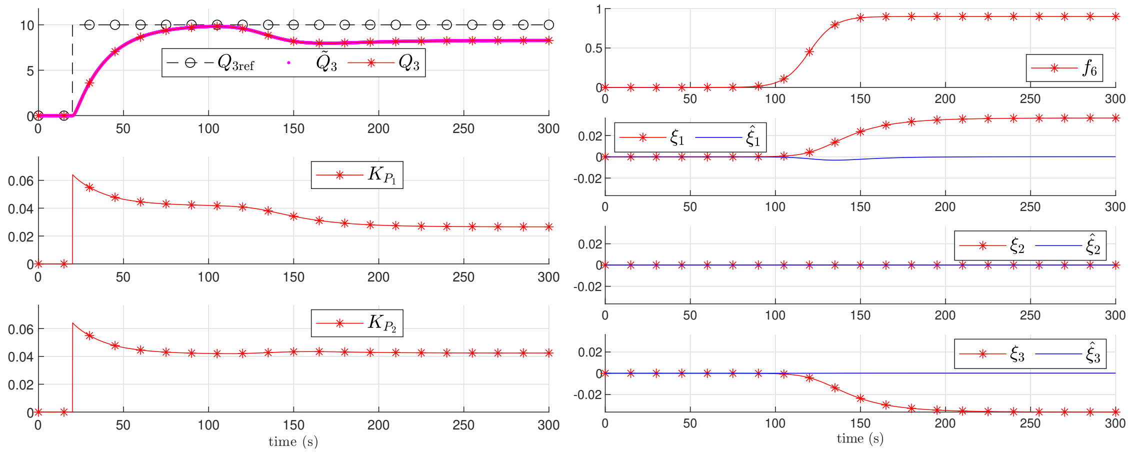
terms, this means that valve K_{13} is gradually closed, allowing only 10% of its nominal flow capacity to pass.

Results: The results are shown in Figures 5.7–5.9, and the corresponding performance metrics are summarized in Table 5.2.

- Nominal Model: unrecovered tracking; the fault estimate does not converge, resulting in an infinite estimation time.
- Exact Fault Model: immediate recovery with low tracking error and no actuator saturation.
- Estimated Model: significant performance degradation, characterized by high values of RMSE and SSE (close to those obtained with the nominal model), and infinite estimation time due to the lack of convergence in the fault estimate.

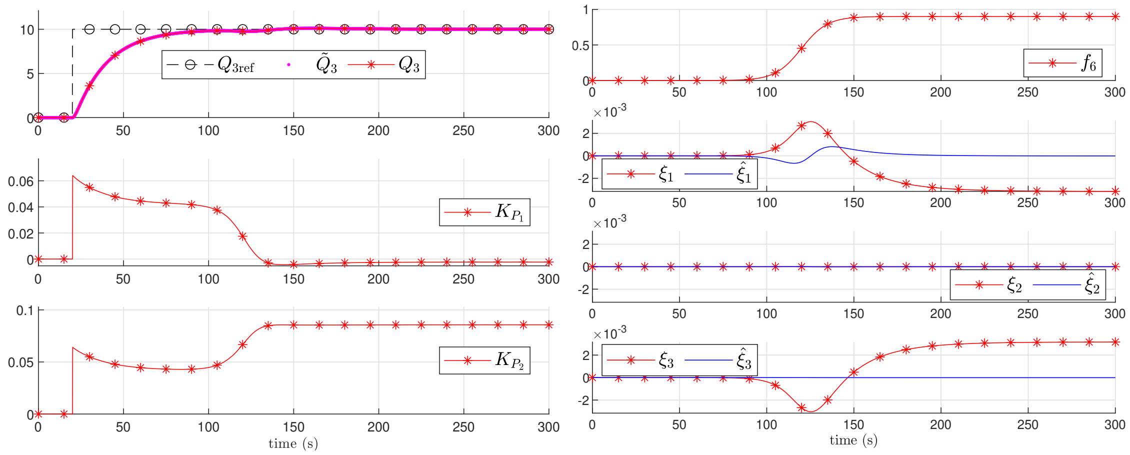
Table 5.2: Closed-loop performance metrics under an incipient plant fault (pipe blockage) initiated at approximately $t = 85$ s with a maximum magnitude of $f_6 = 0.9$.

Model	**RMSE$_{\text{post}}$ (cm^3/s)**	**SSE (cm^3/s)**	**Estimation Delay (s)**	**Recovery Time (s)**
Nominal	1.6084	1.8448	∞	∞
Exact	0.1105	0.0033	∞	0
Estimated	1.3879	1.7287	∞	∞



(a) (b)

Figure 5.7: Closed-loop results using the nominal model $(A_0,\ B_0)$ under an incipient plant fault with a maximum magnitude of $f_6 = 0.9$.



(a) (b)

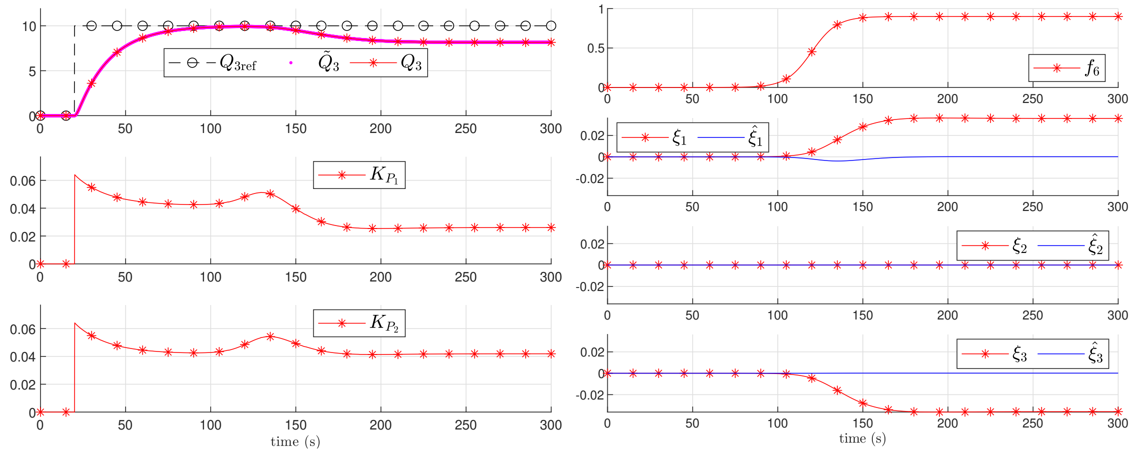
Figure 5.8: Closed-loop results using the exact model $(A(f_k),\ B(f_k))$ under an incipient plant fault with a maximum magnitude of $f_6 = 0.9$.

5.6 Plant Fault: Leakage in Tank T_2

Fault description: At approximately $t = 115$ s, an incipient leakage begins to develop in tank T_2. This fault is modeled by gradually opening valve K_2 until it is fully open, corresponding to $f_9 = 1$.

Results: The results are shown in Figures 5.10–5.12, and the corresponding performance metrics are summarized in Table 5.3.

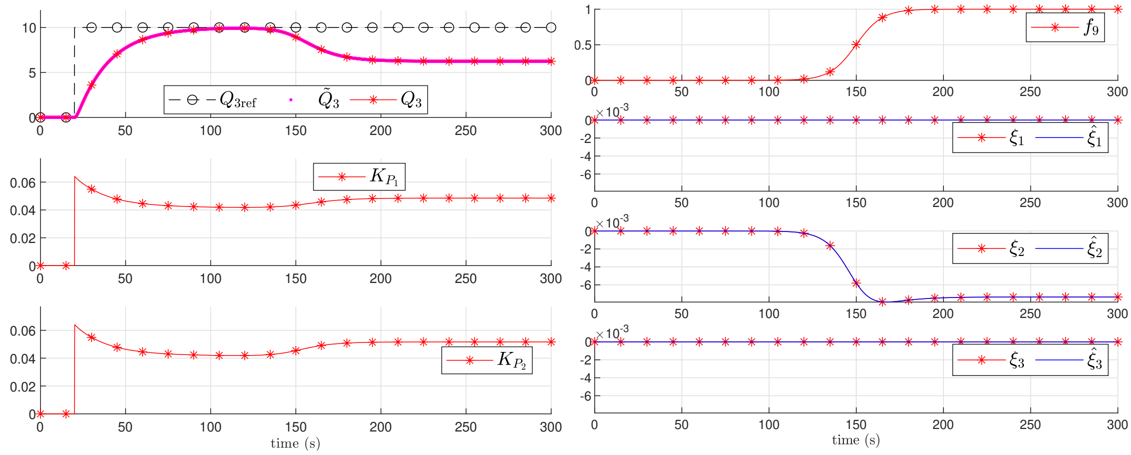
- Nominal Model: unrecovered tracking (infinite recovery time), short fault estimation time, and high tracking error values.
- Exact Fault Model: immediate recovery with negligible tracking error and no actuator saturation.



(a) (b)

Figure 5.9: Closed-loop results using the estimated model $\left(\hat{A}(f_k),\, \hat{B}(f_k)\right)$ under an incipient plant fault with a maximum magnitude of $f_6 = 0.9$.

- Estimated Model: slight performance degradation, characterized by a low RMSE and a small offset error ($\sim 0.75\,\text{cm}^3/\text{s}$), along with a short estimation delay ($\sim 6.4\,\text{s}$).

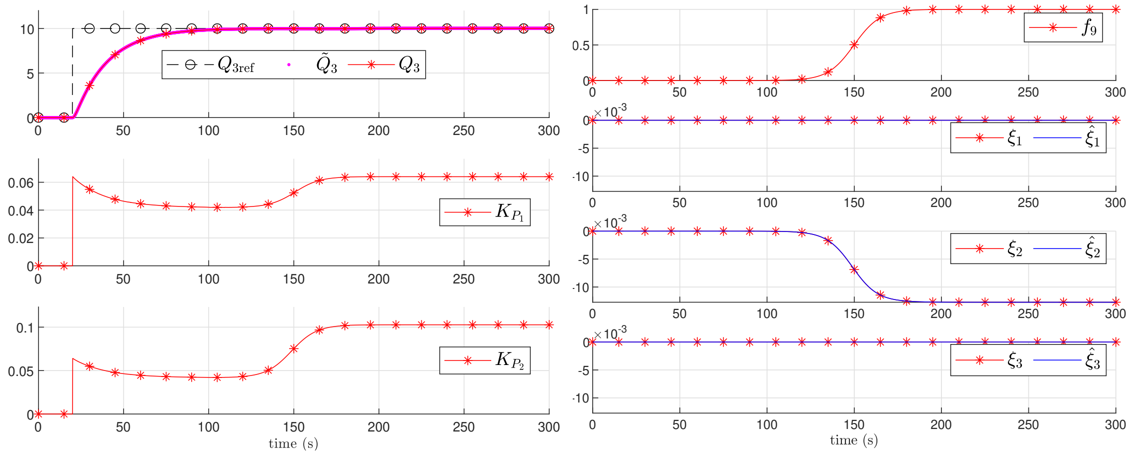


(a) (b)

Figure 5.10: Closed-loop results using the nominal model $\left(A_0,\, B_0\right)$ under an incipient plant fault with a maximum magnitude of $f_9 = 1$.

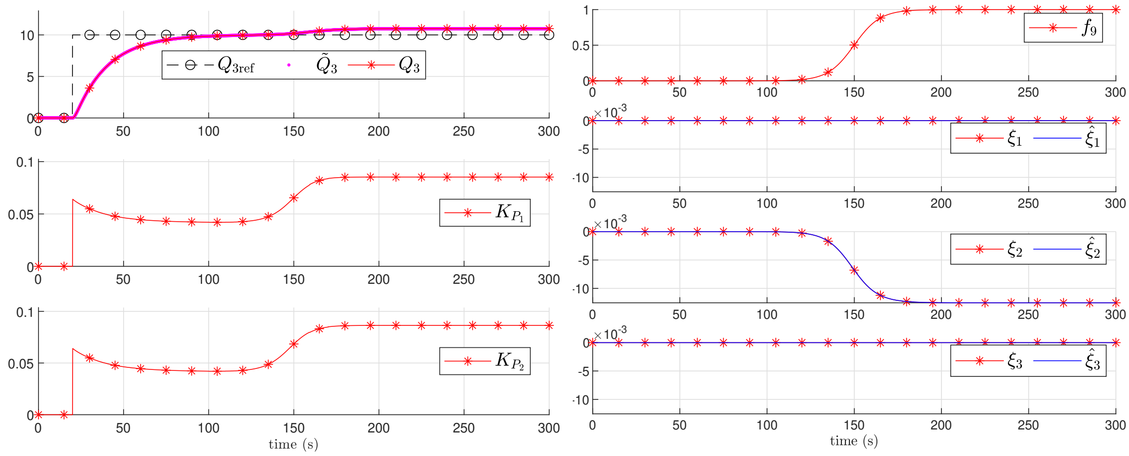
Table 5.3: Closed-loop performance metrics under an incipient plant fault (tank leakage) initiated at approximately $t = 115$ s with a maximum magnitude of $f_9 = 1$.

Model	**RMSE$_{\text{post}}$ (cm^3/s)**	**SSE (cm^3/s)**	**Estimation Delay (s)**	**Recovery Time (s)**
Nominal	3.1529	3.7551	4.6	∞
Exact	0.0206	0.0102	6.3	0
Estimated	0.6288	0.7507	6.4	∞



(a) (b)

Figure 5.11: Closed-loop results using the exact model $\big(A(f_k),\, B(f_k)\big)$ under an incipient plant fault with a maximum magnitude of $f_9 = 1$.



(a) (b)

Figure 5.12: Closed-loop results using the estimated model $\big(\hat{A}(f_k),\, \hat{B}(f_k)\big)$ under an incipient plant fault with a maximum magnitude of $f_9 = 1$.

Discussion: Overall, the results demonstrate that the proposed AFTC framework can effectively accommodate actuator and certain plant faults, restoring tracking performance close to the ideal case when accurate fault estimates are available. For abrupt actuator faults, the estimated-model configuration achieved recovery dynamics and steady-state accuracy comparable to the exact-model scenario, with only a minor delay due to fault estimation. In contrast, severe nonlinear plant faults, such as high-level blockages, posed greater challenges, limiting estimation convergence and preventing full recovery.

Chapter 6

Conclusions

This work proposed and evaluated numerically an integrated Active Fault-Tolerant Control (AFTC) framework that combines Model Predictive Control (MPC) with Moving Horizon Estimation (MHE) for real-time fault accommodation in constrained multi-variable systems. The approach was validated using the three-tank benchmark system under nominal conditions and in the presence of actuator and plant faults.

The proposed control architecture is composed of three main components:

- An MPC formulation with terminal set design and explicit handling of state and input constraints to ensure recursive feasibility and asymptotic stability.
- An MHE-based estimator capable of reconstructing both system states and fault residuals from recent input-output data over a sliding estimation horizon.
- A control reconfiguration scheme in which the internal prediction model of the MPC is updated online using the estimated fault residuals, yielding a Fault-Tolerant MPC (FTMPC) capable of compensating for altered system dynamics.

Simulation results demonstrated that, under nominal conditions, the proposed framework achieved precise setpoint tracking with negligible steady-state error while satisfying all operational constraints. In the case of abrupt actuator faults, the MPC–MHE integration provided fast and accurate fault estimation, enabling timely model reconfiguration and recovery of performance close to that obtained with perfect fault knowledge. Only a short transient degradation was observed due to estimation delay. For severe plant faults involving significant nonlinear effects, such as high-degree pipe blockages and tank leakages, the approach achieved partial recovery. In these scenarios, fault estimation convergence was limited, constraining the benefits of reconfiguration; nevertheless, the proposed method consistently outperformed the nominal-model configuration

In summary, the integration of real-time fault residual estimation and predictive control into a unified AFTC architecture has demonstrated effectiveness in enhancing resilience against faults in constrained multi-variable systems. The approach is particularly relevant for linear applications, where both performance preservation and constraint satisfaction are essential under fault conditions.

6.1 Future Work

Several research directions can be pursued to improve and extend the proposed framework:

- **Sensor fault accommodation:** extend the augmented estimator to include measurement equations for detecting and compensating sensor faults.
- **Robust and nonlinear estimation:** explore robust or nonlinear MHE formulations to improve estimation accuracy under disturbances and significant model mismatches.
- **Enhanced MPC robustness:** incorporate robustness margins in terminal set and cost design to maintain feasibility under late or imperfect fault estimation.
- **Nonlinear MPC integration:** replace the linear MPC with a nonlinear counterpart to better capture plant nonlinearities and improve fault accommodation.

The combination of advanced estimation and predictive control techniques remains a promising direction for developing fault-tolerant systems capable of maintaining high performance and safety in uncertain and fault-prone environments.

Bibliography

[1] M. Blanke, M. Kinnaert, J. Lunze, M. Staroswiecki, J. Schröder, Diagnosis and fault-tolerant control, Vol. 2, Springer, 2006.

[2] T. Steffen, Control reconfiguration of dynamical systems: linear approaches and structural tests, Vol. 320, Springer Science & Business Media, 2005.

[3] J. H. Richter, Reconfigurable control of nonlinear dynamical systems: a fault-hiding approach, Vol. 408, Springer, 2011.

[4] I. Bessa, V. Puig, R. M. Palhares, Reconfiguration blocks and fault hiding: Design, applications, and challenges, Annual Reviews in Control (2023) 100896.

[5] S. Hajshirmohamadi, F. Sheikholeslam, N. Meskin, Simultaneous actuator fault estimation and fault-tolerant tracking control for multi-agent systems: A sliding-mode observer-based approach, International Journal of Control 95 (2) (2022) 447–460.

[6] J. Lan, R. J. Patton, A new strategy for integration of fault estimation within fault-tolerant control, Automatica 69 (2016) 48–59.

[7] A. Oghbaee, B. Shafai, S. Nazari, Complete characterisation of disturbance estimation and fault detection for positive systems, IET Control Theory & Applications 12 (7) (2018) 883–891.

[8] S. Gao, G. Ma, Y. Guo, W. Zhang, Fast actuator and sensor fault estimation based on adaptive unknown input observer, ISA transactions 129 (2022) 305–323.

[9] T. Keijzer, R. M. Ferrari, Threshold design for fault detection with first order sliding mode observers, Automatica 146 (2022) 110600.

[10] M. Mousavi, M. Rahnavard, S. Haddad, Observer based fault reconstruction schemes using terminal sliding modes, International Journal of Control 93 (4) (2020) 881–888.

[11] K. Zhang, B. Jiang, X. Yan, C. Edwards, Interval sliding mode observer-based fault accommodation for non-minimum phase lpv systems with online control allocation, International Journal of Control 93 (11) (2020) 2675–2689.

[12] P. J. Prieto, C. Plata-Ante, R. Ramírez-Villalobos, Fuzzy extended state observer for the fault detection and identification, ISA transactions 128 (2022) 11–20.

[13] Z.-H. Liu, J. Nie, H.-L. Wei, L. Chen, F.-M. Wu, M.-Y. Lv, Second-order eso-based current sensor fault-tolerant strategy for sensorless control of pmsm with b-phase current, IEEE/ASME Transactions on Mechatronics 27 (6) (2022) 5427–5438.

[14] C. Sadhukhan, S. K. Mitra, M. K. Naskar, M. Sharifpur, Fault diagnosis of a nonlinear hybrid system using adaptive unscented kalman filter bank, Engineering with Computers 38 (3) (2022) 2717–2728.

[15] A. A. Sheydaeian Arani, M. Aliyari Shoorehdeli, A. Moarefianpour, M. Teshnehlab, Fault estimation based on ensemble unscented kalman filter for a class of nonlinear systems with multiplicative fault, International Journal of Systems Science 52 (10) (2021) 2082–2099.

[16] L. Sheng, S. Liu, M. Gao, W. Huai, D. Zhou, Moving horizon fault estimation for nonlinear stochastic systems with unknown noise covariance matrices, IEEE Transactions on Instrumentation and Measurement (2023).

[17] Y. Wan, T. Keviczky, Real-time fault-tolerant moving horizon air data estimation for the reconfigure benchmark, IEEE Transactions on Control Systems Technology 27 (3) (2018) 997–1011.

[18] E. C. Kerrigan, J. M. Maciejowski, Fault-tolerant control of a ship propulsion system using model predictive control, in: 1999 European Control Conference (ECC), IEEE, 1999, pp. 4602–4607.

[19] Z. Ruan, Q. Yang, S. S. Ge, Y. Sun, Performance-guaranteed fault-tolerant control for uncertain nonlinear systems via learning-based switching scheme, IEEE Transactions on Neural Networks and Learning Systems 32 (9) (2020) 4138–4150.

[20] K. Rudin, G. J. Ducard, R. Y. Siegwart, Active fault-tolerant control with imperfect fault detection information: Applications to uavs, IEEE Transactions on Aerospace and Electronic Systems 56 (4) (2019) 2792–2805.

[21] F. Valencia, J. D. López, A. Márquez, J. J. Espinosa, Moving horizon estimator for measurement delay compensation in model predictive control schemes, in: 2011 50th IEEE Conference on Decision and Control and European Control Conference, IEEE, 2011, pp. 6678–6683.

[22] J. Maciejowski, Discussion on: "reconfigurable fault-tolerant control: A tutorial introduction", European Journal of Control 14 (5) (2008) 387–389.

[23] J. Lunze, J. H. Richter, Reconfigurable fault-tolerant control: A tutorial introduction, European journal of control 14 (5) (2008) 359–386.

[24] A. A. Amin, M. S. Iqbal, M. H. Shahbaz, Development of intelligent fault-tolerant control systems with machine leaprning, deep learning, and transfer learning algorithms: A review, Expert Systems with Applications (2023) 121956.

[25] Y. Song, B. Zhang, C. Wen, D. Wang, G. Wei, Model predictive control for complicated dynamic systems: a survey, International Journal of Systems Science 56 (9) (2025) 2168–2193.

[26] M. M. Morato, J. E. Normey-Rico, O. Sename, Model predictive control design for linear parameter varying systems: A survey, Annual Reviews in Control 49 (2020) 64–80.

[27] F. Meng, X. Shen, H. R. Karimi, Emerging methodologies in stability and optimization problems of learning-based nonlinear model predictive control: a survey, International Journal of Circuit Theory and Applications 50 (11) (2022) 4146–4170.

[28] D. Q. Mayne, J. B. Rawlings, C. V. Rao, P. O. Scokaert, Constrained model predictive control: Stability and optimality, Automatica 36 (6) (2000) 789–814.

[29] H. S. Ganesh, K. Seo, H. E. Fritz, T. F. Edgar, A. Novoselac, M. Baldea, Indoor air quality and energy management in buildings using combined moving horizon estimation and model predictive control, Journal of Building Engineering 33 (2021) 101552.

[30] G. D. Patrón, L. Ricardez-Sandoval, An integrated real-time optimization, control, and estimation scheme for post-combustion co2 capture, Applied energy 308 (2022) 118302.

[31] J. W. Kim, N. Krausch, J. Aizpuru, T. Barz, S. Lucia, P. Neubauer, M. N. C. Bournazou, Model predictive control and moving horizon estimation for adaptive optimal bolus feeding in high-throughput cultivation of e. coli, Computers & Chemical Engineering 172 (2023) 108158.

[32] D. A. Copp, J. P. Hespanha, Simultaneous nonlinear model predictive control and state estimation, Automatica 77 (2017) 143–154.

[33] M. Diehl, H. J. Ferreau, N. Haverbeke, Efficient numerical methods for nonlinear mpc and moving horizon estimation, Nonlinear model predictive control: towards new challenging applications (2009) 391–417.

[34] J. B. Rawlings, D. Q. Mayne, M. Diehl, Model predictive control: theory, computation, and design, 2nd Edition, Nob Hill Publishing Madison, WI, 2017.

[35] H. K. Khalil, J. W. Grizzle, Nonlinear systems, Vol. 3, Prentice hall Upper Saddle River, NJ, 2002.

[36] D. Limon, T. Alamo, E. F. Camacho, Enlarging the domain of attraction of mpc controllers, Automatica 41 (4) (2005) 629–635.

[37] W.-H. Chen, D. J. Ballance, J. O'Reilly, Optimisation of attraction domains of nonlinear mpc via lmi methods, in: Proceedings of the 2001 American Control Conference.(Cat. No. 01CH37148), Vol. 4, IEEE, 2001, pp. 3067–3072.

[38] H. A. Izadi, Y. Zhang, B. W. Gordon, Fault tolerant model predictive control of quad-rotor helicopters with actuator fault estimation, IFAC Proceedings Volumes 44 (1) (2011) 6343–6348.

[39] A. Eltrabyly, D. Ichalal, S. Mammar, Quadcopter trajectory tracking in the presence of 4 faulty actuators: A nonlinear mhe and mpc approach, IEEE Control Systems Letters 6 (2021) 2024–2029.

[40] E. Alcalá, V. Puig, J. Quevedo, Ts-mpc for autonomous vehicles including a ts-mhe-uio estimator, IEEE Transactions on Vehicular Technology 68 (7) (2019) 6403–6413.

[41] E. Bernardi, M. M. Morato, P. R. Mendes, J. E. Normey-Rico, E. J. Adam, Fault-tolerant energy management for an industrial microgrid: A compact optimization method, International Journal of Electrical Power & Energy Systems 124 (2021) 106342.

[42] F. K. Pour, P. Segovia, E. Duviella, V. Puig, A two-layer control architecture for operational management and hydroelectricity production maximization in inland waterways using model predictive control, Control Engineering Practice 124 (2022) 105172.

[43] X. Feng, R. Patton, Active fault tolerant control of a wind turbine via fuzzy mpc and moving horizon estimation, IFAC Proceedings Volumes 47 (3) (2014) 3633–3638.

[44] H. Safaeipour, M. Forouzanfar, A. Casavola, A survey and classification of incipient fault diagnosis approaches, Journal of Process Control 97 (2021) 1–16.

[45] A. Aboelhassan, M. Abdelgeliel, E. E. Zakzouk, M. Galea, Design and implementation of model predictive control based pid controller for industrial applications, Energies 13 (24) (2020) 6594.

[46] M. Huo, H. Luo, C. Cheng, K. Li, S. Yin, O. Kaynak, J. Zhang, D. Tang, Subspace-aided sensor fault diagnosis and compensation for industrial systems, IEEE Transactions on Industrial Electronics 70 (9) (2022) 9474–9482.

[47] A. O. Farias, G. A. C. Queiroz, I. V. Bessa, R. L. P. Medeiros, L. C. Cordeiro, R. M. Palhares, Sim3tanks: A benchmark model simulator for process control and monitoring, IEEE Access 6 (2018) 62234–62254.

[48] E. R. Rohr, L. F. A. Pereira, D. F. Coutinho, Robustness analysis of nonlinear systems subject to state feedback linearization, Sba: Controle & Automação Sociedade Brasileira de Automatica 20 (2009) 482–489.

[49] S. P. Boyd, L. Vandenberghe, Convex optimization, Cambridge university press, 2004.

[50] Á. González, Measurement of areas on a sphere using fibonacci and latitude–longitude lattices, Mathematical geosciences 42 (1) (2010) 49–64.

[51] J. M. Bravo, D. Limón, T. Alamo, E. F. Camacho, On the computation of invariant sets for constrained nonlinear systems: An interval arithmetic approach, Automatica 41 (9) (2005) 1583–1589.

[52] T. M. Inc., MATLAB version: 9.11.0 (R2021b), Natick, Massachusetts, United States, https://www.mathworks.com (2021).

[53] J. Löfberg, Yalmip: A toolbox for modeling and optimization in matlab, in: 2004 IEEE international conference on robotics and automation (IEEE Cat. No. 04CH37508), IEEE, 2004, pp. 284–289, https://yalmip.github.io.

[54] J. F. Sturm, Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones, Optimization methods and software 11 (1-4) (1999) 625–653.

[55] M. Herceg, M. Kvasnica, C. N. Jones, M. Morari, Multi-parametric toolbox 3.0, in: 2013 European control conference (ECC), IEEE, 2013, pp. 502–510, https://www.mpt3.org/.