

UNIVERSITY OF TECHNOLOGY, SYDNEY
Faculty of Engineering and Information Technology

**On Data-driven Modelling and Terminal Sliding
Mode Control of Dynamic Systems with
Applications**

by

Ansu Man Singh

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

Doctor of Philosophy

Sydney, Australia

2018

Certificate of Authorship/Originality

I hereby certify that the work produced in this thesis has neither been previously submitted for a degree nor has it been submitted as a part of the requirements for other degree except as fully acknowledged within this text.

I also certify that this thesis has been written by myself. Any help that I have received in my research and in the preparation of the thesis itself has been fully acknowledged. In addition, I certify that all information sources and literature used are quoted in the thesis.

© Copyright 2018 Ansu Man Singh

ABSTRACT

On Data-driven Modelling and Terminal Sliding Mode Control of Dynamic Systems with Applications

by

Ansu Man Singh

This thesis addresses critical issues in system modelling and control with some applications to robotics and automation. The main content is divided into three parts, namely data-driven identification, fast terminal sliding mode control alongside underactuated crane control, and robotic pointing system for thermoelastic stress analysis (TSA).

The first part is devoted to system modelling. A dynamic model can be identified from data collected (input and output data from the plant). However, the data obtained is often affected by noise. Hence, such algorithms for modelling the plant should be robust enough to accurately predict dynamic behaviour of the system in the presence of noisy data. Taking this into account, this thesis focuses on subspace-based identification methods, and proposes an effective algorithm based on the Least-Square Support Vector Regression (LS-SVR). In the proposed algorithm, the system identification is formulated as a regression problem to be solved by applying multi-output LS-SVR.

The second part of the thesis deals with the control of underactuated systems which are subject to uncertainties including nonlinearities, parameter variations, and external disturbances. Among many control methodologies, Sliding Mode Control (SMC) is known for its strong robustness. Conventional SMC usually consists of linear sliding surfaces, which can only guarantee the asymptotic stability of the system, and hence, takes infinite time to reach the equilibrium. Requirements of finite-time stability can be fulfilled by adding the sliding function with a fractional

nonlinear term to achieve the Terminal Sliding Mode, and using another attractor can lead to a faster response, called the Fast Terminal Sliding Mode (FTSM). FTSM is theoretically promising but it has limited application in real-time systems. This thesis is devoted to bridging this practical gap by the developing a FTSM controller for underactuated mechanical systems.

The third part of this thesis presents the applications of the proposed LS-SVR based identification algorithm and FTSM control scheme. Here, theoretical developments are implemented on a laboratorial gantry crane and an optical pointing system, respectively. Performance of both LS-SVR identification and FTSM control is verified through extensive simulation and experimental results. Notably, the work for this thesis has been applied to the RobotEye, an industrial pointing system of Ocular Robotics Pty. Ltd., which consists of a mirror integrated with other sensors such as laser sensors and vision cameras for robotic navigation or structural health monitoring with TSA.

List of Publications

Journal Papers

- J-1. **A. M. Singh**, and Q. P. Ha “Fast Terminal Sliding Control Application for Second-Order Underactuated Systems,” *International Journal of Control, Automations and Systems*, 2018. (Submitted: 30 October, 2018)
- J-2. **A. M. Singh**, and Q. P. Ha “System Identification using Multiple Output Least-Square Support Vector Regression,” *International Journal of Systems Science*, 2018. (Submitted: 16 September, 2018)

Conference Papers

- C-1. **A. M. Singh**, V. T. Hoang and Q. P. Ha, “Fast terminal sliding mode control for gantry cranes, *Proc. 33rd International Symposium on Automation and Robotics in Construction*, pp. 437-443, July. 18-21, 2016.
- C-2. **A. M. Singh**, Q. P. Ha, D. Wood, M. Bishop “Low-latency Vision-based Fiducial Detection and Localisation for Object Tracking, *Proc. 34th International Symposium on Automation and Robotics in Construction*, pp. 706-711, June 28- July 1, 2017.
- C-3. **A. M. Singh**, Q. P. Ha, D. Wood, M. Bishop, Q. Ngugen, and A. Wong “RobotEye Technology for Thermal Target Tracking Using Predictive Control, *Proc. 35th International Symposium on Automation and Robotics in Construction*, pp. 1167-1173, July 20- 25, 2018.
- C-4. **A. M. Singh**, M. D. Phung, and Q. P. Ha “Modelling and Fast Terminal Sliding Mode Control for Mirror-based Pointing Systems, *Proc. 15th International Conference on Control Automation Robotics and Vision*, pp. 1151-1157, Nov. 18-21, 2018.

Contents

Certificate	ii
Abstract	iii
List of Publications	v
List of Figures	xi
List of Tables	xvi
List of Abbreviations	xviii
List of Notations	xx
Chapter 1: Introduction	1
1.1 Background	1
1.2 Research Objectives	3
1.3 Thesis Organization	3
Part I System Identification using Subspace Methods	6
Chapter 2: Literature Survey on Subspace Based Identification Methods	7
2.1 Introduction	7
2.2 Problem Description	8
2.3 Block Representation of Input and Output Sequences	9
2.4 Orthogonal and Oblique Projection	10
2.5 A generalised framework and steps in Subspace Identification	10
2.5.1 Estimation of Γ_h and \mathbf{X}_h from \mathcal{O}_h	13

2.5.2	Calculation of A , B , C , and D from Γ_h and \mathbf{X}_h	14
2.5.3	An alternative method to estimate \hat{A} , \hat{B} , \hat{C} , and \hat{D}	15
2.6	Numerical Algorithms for Subspace State Space System Identification (N4SID)	16
2.7	Multivariable Output-Error State Space Model Identification (MOESP)	17
2.8	Principal Component Analysis Approach to System Identification	18
2.8.1	Principal component analysis	19
2.9	Summary	20
Chapter 3:	Subspace based System Identification using LS-SVR.	21
3.1	Introduction	21
3.2	Least Square Support Vector Regression (LS-SVR)	21
3.3	LS-SVR for Multidimensional Systems	23
3.3.1	Extended feature space	24
3.3.2	Multi-output LS-SVR (MLS-SVR)	25
3.4	Subspace Identification Methods and Regression	25
3.5	Least Square Support Vector Regression approach for $\mathbf{Y}_f/\mathbf{U}_f^\perp$ and $\mathbf{W}_p/\mathbf{U}_f^\perp$	26
3.6	Numerical Example	32
3.6.1	Scenario 1	33
3.6.2	Scenario 2	36
3.6.3	Scenario 3	38
3.7	Summary	41

Part II	Fast Terminal Sliding Mode Control	42
Chapter 4: Literature Survey on Fast Terminal Sliding Mode		43
4.1	Introduction	43
4.2	Sliding Mode Control (SMC)	45
4.3	Terminal Sliding Mode (TSM)	47
4.4	Fast Terminal Sliding Mode (FTSM)	49
4.5	Comparison of TSM and FTSM	49
4.6	Summary	51
Chapter 5: Fast Terminal Sliding Mode and Applications		52
5.1	Introduction	52
5.2	Underactuated Mechanical Systems and Modelling	53
5.3	Hierarchical Sliding Mode Framework	55
5.3.1	Stability Analysis	55
5.4	HSM and FTSM based control system design for underactuated systems	57
5.4.1	Stability analysis	58
5.4.2	Stability for crane dynamics	60
5.5	Mirror-based Pointing Technology and modelling	61
5.5.1	Pointing System Construction and System Variables	61
5.5.2	System Modelling	62
5.5.3	Identification of System Model	64
5.6	FTSM Control for the Optical Pointing Sensor	65
5.6.1	FTSM based control formulation	65
5.6.2	Discrete-time FTSM control input synthesis	66
5.6.3	Stability Analysis	70

5.7	Summary	71
Part III Applications	73	
Chapter 6: Application of Mirror-based Technology in Tracking Thermal Targets	74	
6.1	Introduction	74
6.2	System overview	76
6.2.1	System configuration	78
6.2.2	System architecture	79
6.3	Marker Detection System	81
6.3.1	Fiducial design	81
6.3.2	Marker detection and pose estimation of target	82
6.3.3	Experiment with other marker detection systems	86
6.4	Target Motion Modeling and Identification	91
6.4.1	Experimental Setup and data gathering	91
6.4.2	Identification using random signals	92
6.4.3	Identification of sinusoidal input	95
6.5	Controllers and their Comparisons	97
6.5.1	Fast Terminal Sliding Mode (FTSM)	97
6.5.2	Terminal Sliding Mode (TSM)	100
6.5.3	Model Predictive Control (MPC)	102
6.5.4	Simulation and Comparison of TSM, FTSM, and MPC	106
6.5.5	Experimental results	110
6.6	Performance Evaluation of the Tracking System	110
6.7	Summary	114

Chapter 7: Application of FTSM on Gantry cranes	116
7.1 Introduction	116
7.2 Crane Dynamics	117
7.3 Control System	119
7.4 Simulation results	120
7.4.1 Gain Tuning using Genetic Algorithm	120
7.4.2 Control performance	122
7.4.3 Comparison with SMC and TSM	129
7.5 Experimental results	131
7.5.1 Control implementation	131
7.5.2 Test results	133
7.5.3 Trajectory tracking	138
7.6 Summary	145
Chapter 8: Conclusion and Future Work	146
8.1 Thesis Contribution	146
8.2 Future work	147

List of Figures

1.1	Structure of the thesis.	4
2.1	Geometric interpretation of a) orthogonal and b) oblique projection.	11
2.2	Geometric interpretation of \mathcal{O}_h for N4SID.	17
3.1	Singular values of the SVD of \mathcal{O}_h for various methods	33
3.2	Estimation of the poles located at a) $0.67 + j0.67$, b) $0.67 - j0.67$, c) $-0.67 + j0.67$, and d) $-0.67 - j0.67$.	35
3.3	Comparison of the RMSE for the Scenario 1	36
3.4	Estimation of the poles for scenario 2.	37
3.5	RMSE with respect to N for scenario 2.	38
3.6	Poles estimation of the system for scenario 3.	39
3.7	RMSE for the scenario 3.	40
3.8	Singular values for a) scenario 1, b) scenario 2.	41
4.1	Tower crane working under harsh weather condition for the construction of buildings.	43
4.2	Bridge inspection using multiple UAVs in a formation (Source: Hoang et al. (2018)).	44
4.3	Phase portrait of a system states in TSM manifold.	48
4.4	Comparison of FTSM and TSM dynamics at the sliding manifold.	50

5.1	Mirror-based pointing sensor from Ocular Robotics Pty. Ltd.	62
5.2	Responses of the identified models, i.e. H_{11} and H_{22}	66
5.3	Transient analysis of discrete-time sliding function $\sigma_2[n]$	69
6.1	Thermoelastic stress analysis (TSA) on a test coupon using a thermal camera (Source: Saux and Doudard (2017)).	75
6.2	Thermoelastic stress analysis (TSA) of aircraft components: a) TSA of center-barrel structure of a F/A-18 Hornet aircraft at Defence Science Technology Group (DSTG) (Source: Rajic et al. (2018)), and b) Stress analyses of a P-3C Orion wing leading edge rib (Source: Wong et al. (2014)).	76
6.3	Components of the prototype thermal target tracking system.	77
6.4	Arrangement of the vision camera and pointing sensor for tracking target.	79
6.5	System architecture for thermal target tracking system	80
6.6	The proposed concentric circle based fiducial for the marker detection system.	81
6.7	Flow chart for marker detection algorithm.	82
6.8	Image segmentation of the captured images into green and blue channel using HSV.	83
6.9	Inner circle detection using the blob detection algorithm.	84
6.10	Orientation and position estimation of the fiducial marker in 3D space.	86
6.11	Marker detection systems: a) Aruco marker b) AprilTag marker system.	87
6.12	Experimental setup for benchmarking the proposed marker detection system.	88

6.13 Performance evaluations of the marker systems in terms of detection rate.	89
6.14 Computation time for Aruco, AprilTag, and the proposed marker detection system.	90
6.15 Cantilever whose motion-model is under study.	91
6.16 Components in experiment: (a) Amplifier for motor, (b) 100 fps camera (c), National Instrument DAQ, and (d) experimental setup. .	92
6.17 Output of the cantilever system with random input.	93
6.18 AIC for different values of n_x	94
6.19 Predicted and actual output for (a) LS-SVR based method, (b) SMIPCA-E, c) MOESP, and d) CVA.	95
6.20 Predicted and actual output for (a) the proposed method, (b) SMIPCA-E , c) MOESP, and d) CVA.	96
6.21 Controller architecture for the pointing system using FTSM	98
6.22 Step reference signal tracking by elevation and azimuth angles.	99
6.23 σ_1 and σ_2 for elevation angle.	99
6.24 Sinusoidal reference signal tracking by the proposed FTSM controller.	100
6.25 Step signal tracking response for the TSM controller	101
6.26 Sinusoidal signal tracking response for the TSM controller.	102
6.27 Plot of the response of MPC controller for step reference signals.	105
6.28 Elevation and azimuth response of MPC control system for sinusoidal reference signals.	106
6.29 Comparison of the proposed method with TSM and MPC for sinusoidal reference signal.	107
6.30 Comparison of the proposed method with TSM and MPC for step reference signal.	108

6.31 Comparison of ISE for elevation angle.	108
6.32 Integral of Square Error for Azimuth angle.	109
6.33 Real time experiment for MPC.	110
6.34 Tracking errors in azimuth and elevation angles in real experiment. .	111
6.35 Experimental setup for tracking system.	112
6.36 An example of the PCA of target points showing major and minor axis.	114
7.1 Diagram of a gantry crane.	118
7.2 Plot of lk_1 and $k_2 \cos(\theta)$	120
7.3 Gantry crane laboratory testbed.	120
7.4 Flowchart for tuning of controller gains using Genetic Algorithm . .	122
7.5 Response of the trolley position.	124
7.6 Responses of the payload swing angle.	125
7.7 Responses of sliding functions s_1 , s_2 and S	125
7.8 Responses the control input u	126
7.9 Performance of the controller with respect to the variations in rope length.	127
7.10 Performance of the proposed control system with resect to varing payload mass.	128
7.11 Performance of the controller for different deadband.	129
7.12 Comparison of FTSM, TSM, and SMC responses for the cart position x , and payload swing angle θ	130
7.13 Block diagram for implementation.	132
7.14 Test results of the proposed FTSM based control scheme for cart position.	133

7.15	Test results of the proposed method for swing angle.	134
7.16	Comparison between simulation and experiment.	135
7.17	Feasibility and performance of the proposed control algorithm.	136
7.18	Experimental responses of the cart position and swing angle against parameter variations.	137
7.19	Comparison with TSM and SMC.	138
7.20	Position and velocity trajectories of the cart.	139
7.21	Position and velocity trajectories of the swinging load.	140
7.22	Sliding functions and control input during tracking.	141
7.23	Control performance of cart position with respect to varying rope length.	142
7.24	Control performance of payload swing angle due to varying rope length.	142
7.25	Response of the cart for the proposed FTSM controller, TSM, and SMC.	143
7.26	Comparison of the swing angle for the proposed FTSM, TSM, and SMC.	143
7.27	ISE and ITSE errors for a) cart's position and b) velocity tracking.	144

List of Tables

5.1	Identified parameters of the coupled-model of the RobotEye.	64
6.1	Specification of the vision camera, thermal camera, and the mirror-based pointing device for the tracking system.	78
6.2	Sizes of the markers during experiment.	87
6.3	MRSE results for random and sinusoidal input cases.	97
6.4	Summaries of the performance of the thermal tracking system with and without MPC.	113
7.1	System and control parameters	123
7.2	Controller parameters for TSM and SMC	131
7.3	Reference trajectory parameters	138

List of Algorithms

1	Pseudocode for subspace-based identification methods.	15
2	Alternative pseudocode to calculate A , B , C , and D matrices.	16
3	Pseudocode for LS-SVR based identification method.	32

List of Abbreviations

2-DOF - 2 Degree of Freedom

SMC - Sliding Mode Control

TSM - Terminal Sliding Mode

FTSM - Fast Terminal Sliding Mode

HSM - Hierarchical sliding mode

UMS - Underactuated Mechanical System

MPC - Model Predictive Control

SVR - Support Vector Regression

KKT - Karush Kuhn Tucker

LS-SVR - Least Square Support Vector Regression

MLS-SVR - Multi-output Least Square Support Vector Regression

SVD - Singular Value Decomposition

PCA - Principal Component Analysis

N4SID - Numerical Algorithm for Subspace State Space System Identification

MOESP - Multivariable Output-Error State Space Model Identification

CVA - Canonical Variable Analysis

SIMPCA - Subspace Identification Methods Principal Component Analysis

TSA - Thermoelastic Stress Analysis

RMSE - Root Mean Square Error

NRMSE - Normalised Root Mean Square Error

MRSE - Mean of Relative Square Error

MVAF - Mean Variance Accounted-for

ISE - Integral of Square Error

ITSE - Integral of Time Square Error

IoT - Internet of Things

AIC - Akaike Information Criterion

API - Application Programming Interface

GA - Genetic Algorithm

List of Notations

\mathbb{R} - Field of real numbers

\mathbb{R}^n - n -dimensional space

$\mathbb{R}^{n \times m}$ - Space of all matrices of $(n \times m)$ -dimension

A^\top - Transpose of matrix A

A^{-1} - Inverse of matrix A

I_n - Identity matrix of $(n \times n)$ -dimension.

$\mathbf{0}_{(n \times m)}$ - Zero matrix of $n \times m$ -dimension.

$\mathbf{1}_{(n \times m)}$ - One matrix of $n \times m$ -dimension.

$\text{rank}(A)$ - Rank of matrix A .

$\text{trace}(A)$ - Trace of matrix A .

A^\dagger - Moore-penrose pseudo inverse of matrix A .

$\|\cdot\|$ - Euclidean norm of a vector or spectral norm of a matrix.

\forall - For all.

Chapter 1

Introduction

1.1 Background

Due to the advancement of technology in big-data and internet of things (IoT), a huge amount of data is gathered from various sources and systems. The objective of collecting data from a system is to understand its physical structure and its interaction with the environment. This helps to make predictions and apply automation. This process of understanding the physical system is known as system modeling.

In system modeling, models of any physical system can be derived by using various theories such as Newton's law, Lagrangian, etc, Nevertheless, it can be a tedious and time-consuming process because it requires physical description and parametric determination of the system. As an alternative, a data-driven approach can also be applied, which is also known as system identification. This technique is gaining a lot of attention, especially in designing control systems for industrial processes. However, data gathered from systems are not free from noise. As a result, system identification algorithms should be robust enough to accurately predict their dynamics in the presence of noisy data. There is an increasing interest in academics to develop such algorithms. Motivated by the research works in the area, this thesis focuses on such algorithms. Particularly, it focuses on a class of identification methodologies known as subspace identification methods.

Furthermore, this thesis also focuses on developing robust control systems for underactuated systems with robotic applications. Specifically, it focuses on Sliding Mode Control (SMC) based control methodologies. SMC is well known for its ro-

bustness, but, conventionally it uses linear sliding surfaces which leads to asymptotic stability of the system. In other words, such controllers take infinite time to stabilize the system, hence are not suitable for high precision applications. Such shortcomings can be overcome by adding nonlinear terms into the sliding surfaces, and the resulting controllers are known as Fast Terminal Sliding Mode (FTSM). Over the past few decades, there has been a lot of research interest in the application of such controllers. However, they have been mostly applied to fully actuated systems such as robotic manipulators, and very few studies have reported their usage in underactuated systems such as cranes. There is a gap in the understanding of the theoretical and practical implementation of such controllers in such systems, which this thesis aims to fill by developing a FTSM-based robust controller for second-order UMS and demonstrating its application in gantry cranes.

In this thesis, the application of FTSM-based controllers is also presented in a type of robotic sensors, known as mirror-based optical pointing sensors, which consist of a mirror integrated with other sensors such as thermal cameras and vision cameras, etc. This type of sensor has been gaining a lot of attention in the past few years because of their lightweight design and faster response. They have a huge potential in robotic applications such as navigation, localization, and mapping. In addition, such sensors can also be applied to assess the health of structures such as aircraft frames by using a tool known as Thermoelastic Stress Analysis (TSA). Control systems for such applications should provide a faster response besides overcoming the disturbances. Considering the requirement of such applications, FTSM based controllers are studied and developed for mirror-based pointing systems.

Overall, this thesis is divided into three parts. The first part deals with the system modelling and develops a LS-SVR based identification algorithm. In the second part, a FTSM based control system is developed for UMS, and for a mirror-based pointing device. Finally, in the third part, applications of the proposed LS-

SVR based identification algorithm and the FTSM controllers are presented. The FTSM controller for UMS is studied on gantry cranes dynamics. Furthermore, application of the FTSM controller for optical pointing devices is presented for the development of a prototype system for the TSA of mechanical structures. The prototype also consists of a vision and thermal camera which is embedded inside the sensor. The purpose of the pointing device is to compensate the motion of the structure-under-test. Hence, the future motion trajectory of the test specimen is estimated, and the model required for the estimation is identified using the proposed LS-SVR based identification algorithm.

1.2 Research Objectives

The main objectives of this research are listed below:

- Development of a robust algorithm for the system identification
- Construction of a generalised robust and fast convergent controller based on Fast Terminal Sliding Mode (FTSM) for a class of underactuated systems
- Development of a discrete-time FTSM controller for an optical pointing systems

1.3 Thesis Organization

This thesis consists of two components on system identification and FTSM controllers, respectively. Therefore it is divided into three parts as follows:

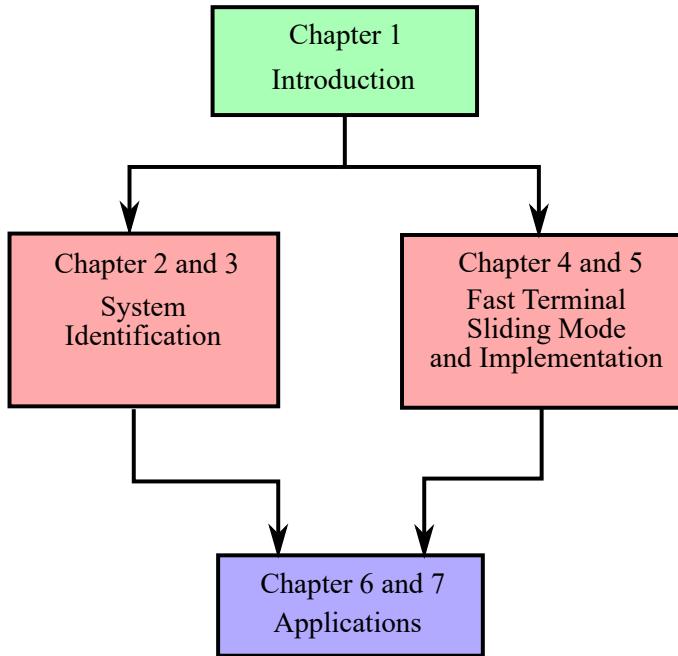


Figure 1.1 : Structure of the thesis.

- **Part I:** This part is devoted to system identification using subspace methods, and it consists of two chapters, i.e. Chapter 2 and Chapter 3. **Chapter 2** presents the literature survey on algorithms, terminologies, and framework used in subspace-based identification methods. Similarly, in **Chapter 3** a Least-Square Support Vector Regression (LS-SVR) based system identification method is proposed.
- **Part II:** This part presents the FTSM based controllers and its implementation. It comprises of two chapters, i.e. Chapter 4 and 5. **Chapter 4** presents a literature survey in Sliding Mode Control, Terminal Sliding Mode, and Fast Terminal Sliding Mode, whereas **Chapter 5** proposes FTSM based control methods for underactuated systems and optical pointing systems.
- **Part III:** The final part of this thesis demonstrates the applications of the proposed LS-SVR based identification method in Part I and the proposed FTSM controllers in Part II. There two chapters in this part, i.e. Chapter 6

and Chapter 7. In **Chapter 6**, application of the proposed FTSM controllers in underactuated gantry cranes is presented with simulation and experimental results. Similarly, application of the proposed LS-SVR based system identification algorithm in Chapter 3, and FTSM controller for optical pointing systems in Chapter 5 is presented in **Chapter 7**. This Chapter presents a thermal target tracking system for thermoelastic stress analysis (TSA) of mechanical structures.

The structure of the thesis can be summarised as depicted in the Fig. 1.1. Finally, conclusion and recommendations for future research is provided in Chapter 8.

Part I

System Identification using Subspace Methods

Chapter 2

Literature Survey on Subspace Based Identification Methods

2.1 Introduction

System identification is a data-driven approach to estimate a model of a plant. This approach has become quite feasible due to the advancement in technologies such as the internet of things (IoT). According to (Chen et al., 2014), systems consisting of IoTs for the environmental sensing and remote controlling have been reported. Furthermore, the advancement in the computational capabilities of computers has made the implementation of identification algorithms quite convenient.

Over the last few decades, the interest towards the development of identification algorithms has increased. Historically it can be traced back to the prediction error method (PEM) (Ljung, 1998) and instrumental variable methods (IVM)(Söderström and Stoica, 1989). However, as reported in (Qin, 2006) these algorithms suffer from parameterizations problem when the number of inputs and outputs are high. In addition, the computational complexity of the methods is also high. This led to the development of a different class of methods based on subspace, also known as subspace identification methods (SIM)(Kruger et al., 2006). SIM algorithms identify a system's parameters by utilizing the tools from linear algebra. These methods offer low computational complexity compared to PEM. Furthermore, the resultant models are in the state space form. As a result, it is convenient to design modern control systems, such as Linear Quadratic Regulator (LQR) and Model Predictive Control (MPC), and apply estimation and filter to such systems (Favoreel et al., 2000).

In this chapter, an introduction to the subspace-based identification methods is presented. Particularly, various terminologies, definitions, and algorithms that are used in such methods are covered in this chapter.

2.2 Problem Description

Consider a discrete time linear time invariant system of the form

$$\begin{aligned} \mathbf{x}[k+1] &= A\mathbf{x}[k] + B\mathbf{u}[k] + \mathbf{w}[k] \\ \mathbf{y}[k] &= C\mathbf{x}[k] + D\mathbf{u}[k] + \mathbf{v}[k], \end{aligned} \tag{2.1}$$

where $\mathbf{x} \in \mathbb{R}^{n_x}$, $\mathbf{u} \in \mathbb{R}^{n_u}$, and $\mathbf{y} \in \mathbb{R}^{n_y}$ are, respectively, the states, inputs and outputs of the system. Similarly $\mathbf{v} \in \mathbb{R}^{n_y}$ and $\mathbf{w} \in \mathbb{R}^{n_x}$ are the measurement and process noise vectors. In the identification methods, the following assumptions are made:

A1: \mathbf{v} and \mathbf{w} are normal distributed random signals, i.e. $\mathbf{v} \sim \mathcal{N}(0, \sigma_v^2)$ and $\mathbf{w} \sim \mathcal{N}(0, \sigma_w^2)$

A2: \mathbf{v} and \mathbf{w} are uncorrelated with \mathbf{u} and \mathbf{y} .

The subspace based identification methods estimate the system matrices, i.e. A , B , C , and D , within similarity transformation from the measurement sequences of the outputs ($\mathbf{y}[0], \mathbf{y}[1], \dots$) and the inputs ($\mathbf{u}[0], \mathbf{u}[1], \dots$). For that, the algorithms represent the input and output sequences into block Hankel matrix forms, which are defined in the following subsections.

2.3 Block Representation of Input and Output Sequences

As mentioned earlier, to proceed with subspace-based identification algorithms the following block Hankel matrices are defined for input and output sequences

$$\begin{aligned} \mathbf{U}_{0|2h-1} &\triangleq \begin{bmatrix} \mathbf{u}_0 & \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_{N-1} \\ \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 & \cdots & \mathbf{u}_N \\ \vdots & \vdots & \vdots & & \vdots \\ \mathbf{u}_{h-1} & \mathbf{u}_h & \mathbf{u}_{h+1} & \cdots & \mathbf{u}_{N+h-2} \\ \mathbf{u}_h & \mathbf{u}_{h+1} & \mathbf{u}_{h+2} & \cdots & \mathbf{u}_{N+h-1} \\ \vdots & \vdots & \vdots & & \vdots \\ \mathbf{u}_{2h-1} & \mathbf{u}_{2h} & \mathbf{u}_{2h+1} & \cdots & \mathbf{u}_{N+2h-2} \end{bmatrix}, \\ \mathbf{Y}_{0|2h-1} &\triangleq \begin{bmatrix} \mathbf{y}_0 & \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_{N-1} \\ \mathbf{y}_1 & \mathbf{y}_2 & \mathbf{y}_3 & \cdots & \mathbf{y}_N \\ \vdots & \vdots & \vdots & & \vdots \\ \mathbf{y}_{h-1} & \mathbf{y}_h & \mathbf{y}_{h+1} & \cdots & \mathbf{y}_{N+h-2} \\ \mathbf{y}_h & \mathbf{y}_{h+1} & \mathbf{y}_{h+2} & \cdots & \mathbf{y}_{N+h-1} \\ \vdots & \vdots & \vdots & & \vdots \\ \mathbf{y}_{2h-1} & \mathbf{y}_{2h} & \mathbf{y}_{2h+1} & \cdots & \mathbf{y}_{N+2h-2} \end{bmatrix}, \end{aligned}$$

where $\mathbf{U}_{0|2h-1} \in \mathbb{R}^{2n_u h \times N}$, $\mathbf{Y}_{0|2h-1} \in \mathbb{R}^{2n_y h \times N}$, h is the horizon and N is number of columns of the matrices. In addition, for subspace identification algorithms let us define also the following input and output block Hankel matrices from $\mathbf{U}_{0|2h-1}$ and $\mathbf{Y}_{0|2h-1}$, respectively, i.e.

$$\mathbf{U}_{0|2h-1} \triangleq \begin{bmatrix} \mathbf{U}_{0|h-1} \\ \mathbf{U}_{h|2h-1} \end{bmatrix} \triangleq \begin{bmatrix} \mathbf{U}_p \\ \mathbf{U}_f \end{bmatrix} = \begin{bmatrix} \mathbf{U}_{0|h} \\ \mathbf{U}_{h+1|2h-1} \end{bmatrix} \triangleq \begin{bmatrix} \mathbf{U}_p^+ \\ \mathbf{U}_f^- \end{bmatrix},$$

and

$$\mathbf{Y}_{0|2h-1} \triangleq \begin{bmatrix} \mathbf{Y}_{0|h-1} \\ \mathbf{Y}_{h|2h-1} \end{bmatrix} \triangleq \begin{bmatrix} \mathbf{Y}_p \\ \mathbf{Y}_f \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_{0|h} \\ \mathbf{Y}_{h+1|2h-1} \end{bmatrix} \triangleq \begin{bmatrix} \mathbf{Y}_p^+ \\ \mathbf{Y}_f^- \end{bmatrix}.$$

Similarly, following input-output block Hankel matrices are also defined:

$$W_p \triangleq \begin{bmatrix} \mathbf{U}_p \\ \mathbf{Y}_p \end{bmatrix}, W_p^+ \triangleq \begin{bmatrix} \mathbf{U}_p^+ \\ \mathbf{Y}_p^+ \end{bmatrix},$$

where $W_p \in \mathbb{R}^{(n_u+n_y)h \times N}$ and $W_p^+ \in \mathbb{R}^{(n_u+n_y)(h+2) \times N}$.

2.4 Orthogonal and Oblique Projection

Identification algorithms also utilize the orthogonal projection of matrices. For this, let us consider matrices \tilde{A} and \tilde{B} . The orthogonal projection of \tilde{A} onto the row space of \tilde{B} is defined as (Delgado et al., 2006; Van Overschee and De Moor, 1996)

$$\tilde{A}/\tilde{B} = \tilde{A}\tilde{B}^\dagger\tilde{B},$$

where \dagger denotes the Moore-penrose pseudo inverse of a matrix defined as $\tilde{B}^\dagger = \tilde{B}^\top (\tilde{B}\tilde{B}^\top)^{-1}$ and $\tilde{B}\tilde{B}^\dagger = I$. Similarly, orthogonal projection of \tilde{A} on the orthogonal complement of the row space of \tilde{B} is given by

$$\tilde{A}/\tilde{B}^\perp = \tilde{A} - \tilde{A}/\tilde{B}.$$

Similarly, the oblique projection of the row space of a matrix \mathcal{A} with respect to the matrix \mathcal{B} and \mathcal{C} is defined as

$$\begin{aligned} \tilde{A}/_{\tilde{B}}\tilde{C} &= \left(\tilde{A}/\tilde{B}^\perp\right) \left(\tilde{C}/\tilde{B}^\perp\right)^\dagger \tilde{C} \\ &= \tilde{A}/\tilde{B}^\perp + \left(\tilde{C}/\tilde{B}^\perp\right)^\dagger \tilde{C}, \end{aligned} \tag{2.2}$$

which represent the projection of \tilde{A} along \tilde{B} onto \tilde{C} . The geometric interpretations of the oblique and orthogonal projections are presented in Fig. 2.1.

2.5 A generalised framework and steps in Subspace Identification

For all SIMs, the following equations of block matrices are utilized

$$\mathbf{Y}_f = \Gamma_f \mathbf{X}_h + H_h \mathbf{U}_f + \mathbf{E}_f, \tag{2.3}$$

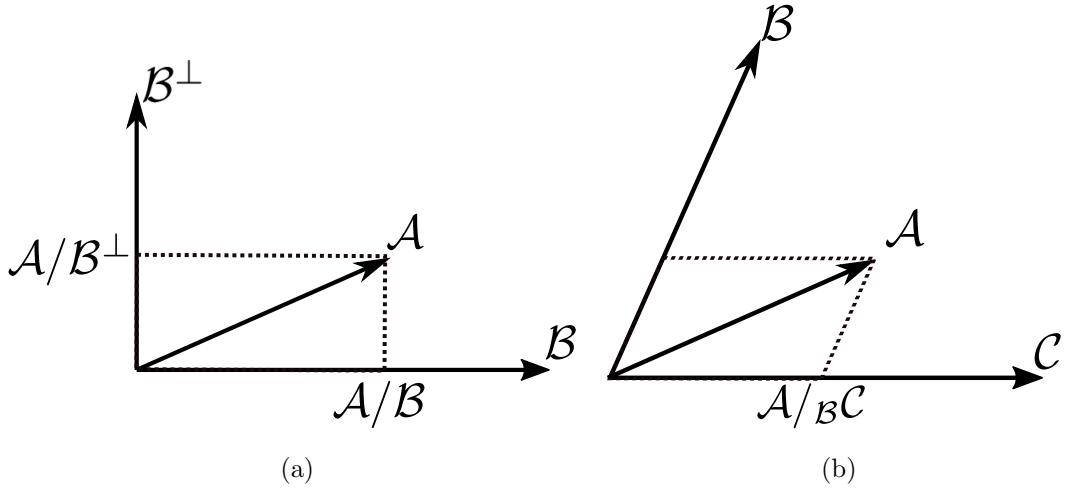


Figure 2.1 : Geometric interpretation of a) orthogonal and b) oblique projection.

where

$$\Gamma_h = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{h-1} \end{bmatrix} \in \mathbb{R}^{n_y h \times n_x}$$

is the observability matrix,

$$H_h = \begin{bmatrix} D & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ CB & D & \mathbf{0} & \cdots & \mathbf{0} \\ CAB & CB & D & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ CA^{h-2}B & CA^{h-3}B & CA^{h-4}B & \cdots & D \end{bmatrix} \in \mathbb{R}^{n_y h \times n_u h}$$

is the Toplitz matrix that relates the input to output, and

$$\mathbf{X}_h = [\mathbf{x}_h \ \mathbf{x}_{h+1} \ \dots \ \mathbf{x}_{h+N-1}] \in \mathbb{R}^{n_x \times N}$$

represents the block matrix representing states of the system (Favoreel et al., 2000).

Taking into account the process and measurement, we have

$$\mathbf{E}_f = \mathbf{G}_f \mathbf{W} + \mathbf{V}, \quad (2.4)$$

where $\mathbf{W} \in \mathbb{R}^{n_x h \times N}$ is the block matrix for process noise, $\mathbf{V} \in \mathbb{R}^{n_y h \times N}$ is the block matrix for measurement noise, and $\mathbf{G}_f \in \mathbb{R}^{n_y h \times n_x h}$ is the transformation from noise to the output.

Now, the orthogonal projection of the future output, i.e. \mathbf{Y}_f , onto the row space of orthogonal complement of \mathbf{U}_f is given by

$$\begin{aligned} \mathbf{Y}_f / \mathbf{U}_f^\perp &= \Gamma_f \mathbf{X}_h / \mathbf{U}_f^\perp + H_h \mathbf{U}_f / \mathbf{U}_f^\perp + \mathbf{E}_f / \mathbf{U}_f^\perp \\ &= \Gamma_f \mathbf{X}_h / \mathbf{U}_f^\perp + \mathbf{E}_f / \mathbf{U}_f^\perp. \end{aligned} \quad (2.5)$$

Here, \mathbf{U}_f is uncorrelated to \mathbf{E}_f , therefore, from the Kalman filter theory

$$\lim_{N \rightarrow \infty} \mathbf{E}_f \mathbf{U}_f^\top = 0, \quad (2.6)$$

which leads to

$$\mathbf{E}_f / \mathbf{U}_f^\perp = \mathbf{E}_f, \quad (2.7)$$

as shown in (Qin, 2006). Hence,

$$\mathbf{Y}_f / \mathbf{U}_f^\perp = \Gamma_f \mathbf{X}_h / \mathbf{U}_f^\perp + \mathbf{E}_f. \quad (2.8)$$

In (Van Overschee and De Moor, 1995), Overschee and De Moor provided a method to unify the subspace based identification methods such as N4SID, MOESP, and CVA. According to the unified framework, the orthogonal projection $\mathbf{Y}_f / \mathbf{U}_f^\perp$ is multiplied on the left and right by weight matrices W_1 and W_2 , respectively, which leads to

$$W_1 \mathbf{Y}_f / \mathbf{U}_f^\perp W_2 = W_1 \mathbf{X}_h / \mathbf{U}_f^\perp W_2 + W_1 \mathbf{G}_h \mathbf{E}_h W_2 = \mathcal{O}_h. \quad (2.9)$$

The weight matrices W_1 and W_2 should satisfy the following conditions:

- (i) $\text{rank}(W_1\Gamma_h) = \text{rank}\Gamma_h$
- (ii) $\text{rank}(\mathbf{X}_h/\mathbf{U}_f^\perp W_2) = \text{rank}\mathbf{X}_h$
- (iii) $W_1\mathbf{G}_h\mathbf{E}_h W_2 = 0.$

It should be noted that by applying the condition (iii) into Equation (2.9), the oblique projection is obtained as,

$$\mathcal{O}_h = W_1\mathbf{Y}_f/\mathbf{U}_f^\perp W_2 = W_1\Gamma_h\mathbf{X}_h/\mathbf{U}_f^\perp W_2 . \quad (2.10)$$

Here, different choices of W_1 and W_2 lead to different algorithms mentioned previously. Details on those algorithms will be discussed in the following subsections. After the calculation of the \mathcal{O}_h , subspace identification algorithms apply the following two steps to calculate the parameters of the state space model

1. Estimation of Γ_h and \mathbf{X}_h from \mathcal{O}_h
2. Calculation of A , B , C , and D from Γ_h and \mathbf{X}_h .

These steps will be discussed in details in the following subsections.

2.5.1 Estimation of Γ_h and \mathbf{X}_h from \mathcal{O}_h

From the conditions (i), (ii) and Equation (2.10), one can reach the following conclusions

$$\begin{aligned} \text{rank}(\mathcal{O}_h) &= n_x \\ \text{row space}(\mathcal{O}_h) &= \text{row space}(\mathbf{X}_h) \\ \text{column space}(\mathcal{O}_h) &= \text{row space}(\Gamma_h) . \end{aligned} \quad (2.11)$$

Interested readers can find details about (2.11) and its proof in (Van Overschee and De Moor, 1996). Furthermore, in SIMs, the singular value decomposition (SVD) of \mathcal{O}_h , is derived, i.e.

$$\mathcal{O}_h = QSV^\top, \quad (2.12)$$

where $Q = [Q_1 \ Q_2]$ is left singular vectors,

$$S = \begin{bmatrix} S_1 & 0 \\ 0 & 0 \end{bmatrix},$$

S_1 is the singular values, and $V = [V_1 \ V_2]^\top$ is the right singular vectors. It should be noted that $\text{rank}(S) = \text{rank}(S_1) = n_x$. In SIMs, Γ_h is obtained as $\Gamma_h = Q_1 S_1^{1/2}$ (Delgado et al., 2006).

In the identification algorithms \mathbf{X}_h can be calculated as

$$\hat{\mathbf{X}}_h = \Gamma_h^\dagger \mathcal{O}_h. \quad (2.13)$$

In addition, the algorithms also calculates $\hat{\mathbf{X}}_{h+1}$ as

$$\hat{\mathbf{X}}_{h+1} = \Gamma_{h-1}^\dagger \mathcal{O}_{h-1}, \quad (2.14)$$

where Γ_{h-1} is obtained from Γ_h by removing last n_y columns. Similarly

$$\mathcal{O}_{h-1} = \bar{W}_1 \mathbf{Y}_f^- / \mathbf{U}_f^{-\perp} \bar{W}_2, \quad (2.15)$$

where

$$\bar{W}_1 = I \in \mathbb{R}^{n_y(h-1) \times N}$$

$$\bar{W}_2 = (W_p^+ / \mathbf{U}_f^\perp)^\dagger (W_p^+ / \mathbf{U}_f^\perp).$$

It should be noted that $\hat{\mathbf{X}}_h$ and $\hat{\mathbf{X}}_{h+1}$ are also known as Kalman filter estimates of \mathbf{X}_h and \mathbf{X}_{h+1} , respectively.

2.5.2 Calculation of A , B , C , and D from Γ_h and \mathbf{X}_h

Once the $\hat{\mathbf{X}}_h$ and $\hat{\mathbf{X}}_{h+1}$ are obtained, one can represent system matrices A , B , C , and D by the following equation

$$\begin{aligned} \hat{\mathbf{X}}_{h+1} &= \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \hat{\mathbf{X}}_h \\ \mathbf{U}_{h|h} \end{bmatrix} + \begin{bmatrix} \mathbf{w} \\ \mathbf{v} \end{bmatrix}, \\ \mathbf{Y}_{h|h} &= \dots \end{aligned} \quad (2.16)$$

where $\mathbf{Y}_{h|h} \in \mathbb{R}^{n_y \times N}$ and $\mathbf{U}_{h|h} \in \mathbb{R}^{n_u \times N}$ are the h^{th} row of $\mathbf{Y}_{0|2h-1}$ and $\mathbf{U}_{0|2h-1}$, respectively. Therefore, the system matrices A , B , C , and D can be calculated upto the similarity transformation in the least square sense, i.e.

$$\begin{bmatrix} \hat{A} & \hat{B} \\ \hat{C} & \hat{D} \end{bmatrix} = \arg \min \left\{ \left\| \begin{pmatrix} \hat{\mathbf{X}}_{h+1} \\ \mathbf{Y}_{h|h} \end{pmatrix} - \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \hat{\mathbf{X}} \\ \mathbf{U}_{h|h} \end{bmatrix} \right\|^2 \right\}. \quad (2.17)$$

It is noted that covariance matrices of the noise \mathbf{w} and \mathbf{v} can be calcuated as the residue of (2.17). This whole process of calculating system matrices can be summarised in the following pseudocode (Algorithm 1).

Algorithm 1: Pseudocode for subspace-based identification methods.

- 1 Form the block Hankel matrices \mathbf{Y}_f , \mathbf{U}_f , \mathbf{W}_p , and \mathbf{W}_p^+ from the input sequences $(\mathbf{u}[0], \mathbf{u}[1], \dots)$ and output sequences $(\mathbf{y}[0], \mathbf{y}[1], \dots)$.
 - 2 Calculate \mathcal{O}_h and \mathcal{O}_{h-1} using (2.9) and (2.15).
 - 3 Apply SVD on \mathcal{O}_h to calculate Γ_h and Γ_{h-1} using (2.12).
 - 4 Calculate $\hat{\mathbf{X}}_h$ and $\hat{\mathbf{X}}_{h+1}$ from Γ_h and Γ_{h-1} using (2.13) and (2.14), respectively.
 - 5 Obtain \hat{A} , \hat{B} , \hat{C} , \hat{D} from $\hat{\mathbf{X}}_h$, $\hat{\mathbf{X}}_{h+1}$, $\mathbf{Y}_{h|h}$, and $\mathbf{U}_{h|h}$, respectively in least-square sense, i.e (2.17).
-

2.5.3 An alternative method to estimate \hat{A} , \hat{B} , \hat{C} , and \hat{D}

In this approach, \hat{A} and \hat{C} are calculated directly from the observability matrix (Γ_h) . By utilizing the definition of Γ_h in (2.3), one can directly extract \hat{C} from the first n_y rows of the estimated observability matrix. On the other hand, \hat{A} can be derived as

$$\hat{A} = \bar{\Gamma}_h \underline{\Gamma}_h^\dagger, \quad (2.18)$$

where $\bar{\Gamma}_h$ and $\underline{\Gamma}_h$ are extracted from Γ_h by removing bottommost and topmost n_y rows, respectively.

Estimation of \hat{B} and \hat{D} requires the estimation of Toeplitz matrix, i.e. \hat{H}_h . For that, one approach is to multiply both sides of (2.3) by Γ_h^\perp and \mathbf{U}_f^\dagger , respectively, which leads to

$$\Gamma_h^\perp \mathbf{Y}_f \mathbf{U}_f^\dagger = \Gamma_h^\perp H_h. \quad (2.19)$$

Here, \hat{H}_h can be estimated in the least square sense, i.e.

$$\hat{H} = \arg \min \left\{ \left\| \Gamma_h^\perp \mathbf{Y}_f \mathbf{U}_f^\dagger - \Gamma_h^\perp H_h \right\|^2 \right\}. \quad (2.20)$$

Once, \hat{H}_h is estimated, \hat{D} is extracted from the first n_y rows and n_u columns of \hat{H}_h .

Finally, B is estimated as

$$\hat{B} = \hat{H}_h \underline{\Gamma}_h^\dagger. \quad (2.21)$$

This method can be summarised by Algorithm 2.

Algorithm 2: Alternative pseudocode to calculate A , B , C , and D matrices.

1 Extract 1st n_y rows from Γ_h for \hat{C}

2 Calculate \hat{A} from (2.18).

3 Estimate Toeplitz matrix (\hat{H}_h) as

$$\hat{H} = \arg \min \left\{ \left\| \Gamma_h^\perp \mathbf{Y}_f \mathbf{U}_f^\dagger - \Gamma_h^\perp H_h \right\|^2 \right\}.$$

4 Extract first n_y rows and n_u columns of \hat{H}_h for \hat{D}

5 Calculate \hat{B} as $\hat{B} = \hat{H}_h \underline{\Gamma}_h^\dagger$.

2.6 Numerical Algorithms for Subspace State Space System Identification (N4SID)

N4SID algorithm was initially proposed by Van Overshee in (Van Overshee and De Moor, 1994, 1996). The acronym for N4SID is Numerical algorithms for

Subspace State Space System IDentification. In this method, the following weight matrices are used (Favoreel et al., 2000):

$$\begin{aligned} W_1 &= I \\ W_2 &= (W_p/\mathbf{U}_f)^\dagger W_p, \end{aligned} \tag{2.22}$$

which leads to

$$\mathcal{O}_h = (\mathbf{Y}_f/\mathbf{U}_f) (\mathbf{W}_p/\mathbf{U}_f)^\dagger \mathbf{W}_p. \tag{2.23}$$

It is clear from (2.23) that \mathcal{O}_h is the oblique projection of the row space of \mathbf{Y}_f along the row space of \mathbf{U}_f on \mathbf{W}_p , i.e.

$$\mathcal{O}_h = \mathbf{Y}_f/\mathbf{U}_f \mathbf{W}_p. \tag{2.24}$$

The geometric interpretation of \mathcal{O}_h is depicted in Fig. 2.2.

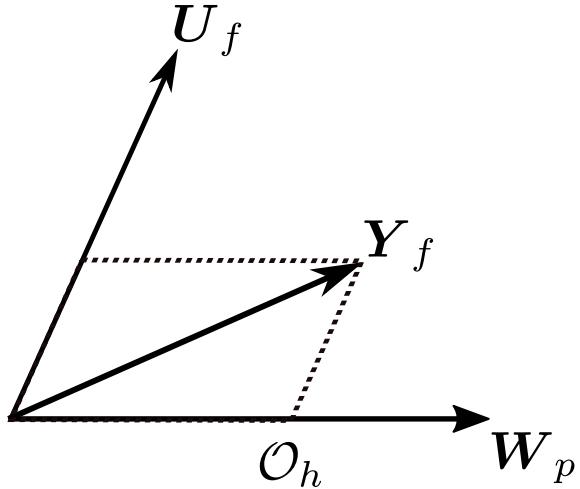


Figure 2.2 : Geometric interpretation of \mathcal{O}_h for N4SID.

2.7 Multivariable Output-Error State Space Model Identification (MOESP)

(MOESP) stands for **M**ultivariable **O**utput-**E**rror **S**tate **sPace** model identification. In this method, LQ decomposition on the block matrix consisting of \mathbf{Y}_f , \mathbf{U}_f ,

and \mathbf{W}_p is applied (Verhaegen and Dewilde, 1992). In other words

$$\begin{bmatrix} \mathbf{U}_f \\ \mathbf{W}_p \\ \mathbf{Y}_f \end{bmatrix} = \begin{bmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \begin{bmatrix} \mathcal{Q}_1^\top \\ \mathcal{Q}_2^\top \\ \mathcal{Q}_3^\top \end{bmatrix}. \quad (2.25)$$

In (Verhaegen and Dewilde, 1992), it has been reported that one can obtain Γ_h by applying SVD on L_{32} , i.e.

$$L_{32} = [Q_1^m \ Q_2^m] \begin{bmatrix} S_1^m & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^\top \\ V_2^\top \end{bmatrix}, \quad (2.26)$$

and

$$\Gamma_h = Q_1^m (S_1^m)^{1/2}. \quad (2.27)$$

Furthermore, as shown in (Van Overschee and De Moor, 1995), MOESP can be represented in the framework presented in the subsection 2.5 (Favoreel et al., 2000) by applying the following weight matrices:

$$\begin{aligned} W_1 &= I \\ W_2 &= (\mathbf{W}_p / \mathbf{U}_f^\perp)^\dagger (\mathbf{W}_p / \mathbf{U}_f^\perp). \end{aligned} \quad (2.28)$$

This leads to following equation:

$$\begin{aligned} \mathcal{O}_h &= (\mathbf{Y}_f / \mathbf{U}_f) (\mathbf{W}_p / \mathbf{U}_f^\perp)^\dagger (\mathbf{W}_p / \mathbf{U}_f^\perp) \\ &= \mathbf{Y}_f / \mathbf{U}_f \mathbf{W}_p \Pi_{\mathbf{U}_f}^\perp, \end{aligned} \quad (2.29)$$

where $\Pi_{\mathbf{U}_f}^\perp = I - \mathbf{U}_f^\dagger \mathbf{U}_f$.

2.8 Principal Component Analysis Approach to System Identification

In recent years, there has been an increasing interest in using Principal Component Analysis (PCA) in SIMs. For instance, in (Wang and Qin, 2002), Jin Wang and Joe Qin presented an approach to replace SVD in (2.12) by PCA. Such methodologies are also known as SIMPCA. Before digging dip into the method, let us discuss briefly about PCA of a matrix.

2.8.1 Principal component analysis

PCA decomposes a matrix \mathbf{Z}_f into the following matrices (Ku et al., 1995)

$$\mathbf{Z}_f^\top = \mathbf{T}\mathbf{P}^\top + \tilde{\mathbf{T}}\tilde{\mathbf{P}}^\top \quad (2.30)$$

where \mathbf{T} and \mathbf{P} are, respectively, known as the score and loading matrix of the principal components of \mathbf{Z}_f , whereas, $\tilde{\mathbf{T}}$ and $\tilde{\mathbf{P}}$ represent the score and loading matrix for the residue (Ku et al., 1995). PCA decomposition minimises the norm of the $\tilde{\mathbf{T}}\tilde{\mathbf{P}}$ matrix.

Let us reconsider (2.3) and apply some modifications which leads to the following equation

$$[\mathbf{I} - \mathbf{H}_f] \mathbf{Z}_f = \Gamma_h \mathbf{X}_h - \mathbf{E}_f, \quad (2.31)$$

where

$$\mathbf{Z}_f = \begin{bmatrix} \mathbf{Y}_f \\ \mathbf{U}_f \end{bmatrix} \in \mathbb{R}^{(n_y+n_u)h \times N}.$$

Now, multiplying both sides of (2.31) by an instrumental variable \mathbf{Z}_p^\top , such that the following condition

$$\lim_{N \rightarrow \infty} \mathbf{E}_f \mathbf{Z}_p^\top = 0 \quad (2.32)$$

is satisfied, then \mathbf{E}_f can be removed from (2.31). There are various choices of \mathbf{Z}_p . However, in (Wang and Qin, 2002), the following instrumental variable is presented

$$\mathbf{Z}_p = \begin{bmatrix} \mathbf{Y}_p \\ \mathbf{U}_p \end{bmatrix} \in \mathbb{R}^{(n_y+n_u)h \times N}$$

Furthermore, it has been shown that when $N \rightarrow \infty$, then the following condition is satisfied, i.e.

$$\begin{aligned} \begin{bmatrix} \Gamma_f^\perp \\ -\mathbf{H}_f^\top \Gamma_f^\perp \end{bmatrix} &= \tilde{\mathbf{P}} \mathbf{M} \\ &= \begin{bmatrix} \tilde{\mathbf{P}}_y \\ \tilde{\mathbf{P}}_u \end{bmatrix} \mathbf{M}, \end{aligned} \quad (2.33)$$

where $\tilde{\mathbf{P}}$ is the loading matrix of the residue of the PCA of $\frac{1}{N} \mathbf{Z}_f \mathbf{Z}_p^\top$, i.e.

$$\frac{1}{N} \mathbf{Z}_f \mathbf{Z}_p^\top = \mathbf{P} \mathbf{T}^\top + \tilde{\mathbf{P}} \tilde{\mathbf{T}}^\top, \quad (2.34)$$

where $\mathbf{M} \in \mathbb{R}^{(n_y h - n_x) \times (n_y h - n_x)}$ is a non-singular matrix. By letting $\mathbf{M} = I$, one obtains

$$\Gamma_f^\top = \tilde{\mathbf{P}}_y, \quad (2.35)$$

and

$$-\tilde{\mathbf{P}}_y^\top H_f = \tilde{\mathbf{P}}_u$$

or,

$$\Phi H_f = \Omega, \quad (2.36)$$

where $\Phi = -\tilde{\mathbf{P}}_y^\top$ and $\Omega = \tilde{\mathbf{P}}_u$. Here, by using (2.36), H_f can be calculated in the least square sense, i.e.

$$\hat{H}_f = \arg \min \|\Phi H_f - \Omega\|^2. \quad (2.37)$$

Similarly, Γ_f can be obtained from Γ_f^\perp , since $(\Gamma_f^\perp)^\top \Gamma_f = 0$. Therefore, system matrices \hat{A} , \hat{B} , \hat{C} , and \hat{D} are estimated from Γ_f and H_f . Interested reader can refer to (Wang and Qin, 2002) for detail.

2.9 Summary

Subspace-based identification methods have been presented in this chapter. The methodologies can be generalized by a process, where, at first, block Hankel matrices are formed from the input and output data sequences of a plant, followed by the multiplication by weight matrices. Then, SVD on the resultant matrix is applied to calculate the observability matrix for the system. Finally, system matrices are calculated from the estimated observability matrix. In addition, a new trend in SIMs that utilizes PCA has also been presented in this chapter.

Chapter 3

Subspace based System Identification using LS-SVR

3.1 Introduction

In the previous chapter, various SIM algorithms were presented, which were represented in a generalized framework as the Singular Value Decomposition (SVD) of orthogonal projection of block Hankel matrices formed from input and output data sequences. In all the algorithms, calculation of the projection involves the least-square regression from the input and output block matrices. However, as reported in the literature (Thissen et al., 2004; Balabin and Lomakina, 2011; Xu and Liu, 2012), least square regressions are less robust to noisy data, particularly when it consists of outliers. In order to overcome the issue, this Chapter presents a novel method to calculate the projection matrices by Least Square Support Vector Regression (LS-SVR). Here the LS-SVR projection is formulated as a primal optimization problem, which is solved by applying Lagrangian and Karush–Kuhn–Tucker (KKT) conditions (Vapnik, 2013). Since the regression involving the block matrices has several outputs, the LS-SVR primal equations are formulated for the multiple output case.

3.2 Least Square Support Vector Regression (LS-SVR)

Support Vector Regression (SVR) is a widely used method for regression problems, which basically estimates the mapping from the input space $\mathbf{x} \in \mathbb{R}^{n_x}$ to the

output variable $y \in \mathbb{R}$ using the regression model as

$$f(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + b, \quad (3.1)$$

where $\mathbf{w} \in \mathbb{R}^m$ is the weight, $\phi(\mathbf{x}) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^m$ is a function that maps the input space to higher dimensional space, and $b \in \mathbb{R}$ is the bias. SVR solves the problem by formulation it in the following quadratic optimization problem (Vapnik, 2013):

$$\begin{aligned} & \text{minimize} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=0}^{N-1} (\xi_i + \xi_i^*) \right\} \\ & \text{subject to} \begin{cases} y_i - \mathbf{w}^\top \phi(\mathbf{x}_i) - b \leq \varepsilon + \xi_i \\ \mathbf{w}^\top \phi(\mathbf{x}_i) + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0, \end{cases} \end{aligned} \quad (3.2)$$

where ξ_i and ξ_i^* are the slack variables, $i = (0, \dots, N-1)$, and ε is known as the soft margin. The optimization problem can be solved using quadratic programming algorithms such as inferior point. However, the computational complexity of such algorithms is very high. There are various ways to reduce the complexity, but replacing the inequality in (3.2) by the equality is promising. This changes lead to the a new set of optimization problems known as Least-Square Support Vector Regression (LS-SVR). The optimization problem for the single output case is defined as

$$\begin{aligned} & \text{minimize } J(\mathbf{w}, \boldsymbol{\xi}) = \text{minimize} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 + \frac{\gamma}{2} \sum_{i=0}^{N-1} \xi_i^2 \right\} \\ & \text{subject to } y_i = \mathbf{w}^\top \phi(\mathbf{x}_i) + b + \xi_i, \end{aligned} \quad (3.3)$$

where γ is a regularization variable, and $\boldsymbol{\xi} = [\xi_0, \xi_1, \xi_2, \dots, \xi_{N-1}]^\top \in \mathbb{R}^N$ is a vector of slack variables. The Lagrangian for (3.3) is defined as

$$L(\mathbf{w}, b, \boldsymbol{\xi}) = J(\mathbf{w}, \boldsymbol{\xi}) - \sum_{i=0}^{N-1} \alpha_i^\top (\mathbf{w}^\top \phi(\mathbf{x}_i) + b + \xi_i - y_i), \quad (3.4)$$

where $\alpha_{(i=0, \dots, N-1)}$ are the Lagrangian multipliers. The Karush-Kuhn-Tucker (KKT) conditions for the optimality lead to the following set of equations:

$$\begin{cases} \frac{\partial L}{\partial \mathbf{w}} = 0 \rightarrow \mathbf{w} = \mathbf{X}\boldsymbol{\xi} \\ \frac{\partial L}{\partial b} = 0 \rightarrow \mathbf{1}_{1 \times l}\boldsymbol{\xi} = 0 \\ \frac{\partial L}{\partial \xi_i} = 0 \rightarrow \xi_i = \frac{1}{\gamma}\alpha_i \\ \frac{\partial L}{\partial \alpha_i} = 0 \rightarrow \mathbf{w}^\top \mathbf{x}_i + b + \xi_i - y_i = 0. \end{cases} \quad (3.5)$$

Now, eliminating \mathbf{w} and $\xi_{(i=0 \dots N-1)}$ in the equation 3.5 one obtains following set of linear equations

$$\begin{bmatrix} 0 & \mathbf{1}_{1 \times N} \\ \mathbf{1}_{l \times 1} & \mathbf{X}^\top \mathbf{X} + \frac{1}{\gamma} I \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{y}, \end{bmatrix}, \quad (3.6)$$

where $\mathbf{y} = [y_0, y_1, y_2, \dots, y_{N-1}]^\top \in \mathbb{R}^N$ and $\mathbf{X} = [\phi(\mathbf{x}_0) \phi(\mathbf{x}_1) \phi(\mathbf{x}_2) \dots \phi(\mathbf{x}_{N-1})] \in \mathbb{R}^{m \times N}$. Hence by solving (3.6), one can solve for \mathbf{w} and b in (3.1).

3.3 LS-SVR for Multidimensional Systems

The solution provided by (3.6) is for the single output case, but it does not solve the regression for multiple output variables, i.e. $\mathbf{y} = (y_0 \ y_1 \ \dots \ y_{n_y-1})^\top \in \mathbb{R}^{n_y}$. There studies are limited in this area even though multi-output regression has wide applications. One of the conventional techniques for solving multi-output LS-SVR is by solving each variable independently and then combine them all together. In other words, it calculates the model for each variable, i.e.

$$f_j(\mathbf{x}) = \mathbf{w}_j^\top \phi(\mathbf{x}) + b_j, \forall j = 0 \dots n_y - 1. \quad (3.7)$$

Then, the estimates $f_{j=0 \dots n_y-1}(\mathbf{x})$ are combined at the end. However, in this approach, each function are assumed to be independent of each other. Therefore, it does not provide good approximation when there are correlations between output variables (Borchani et al., 2015). Many literatures have reported two methods to overcome the shortcomings in the conventional approach, which are listed below:

- Extended feature space method

- Multi-output LS-SVR (MLS-SVR)

A brief introduction on the methods are presented in the following subsections.

3.3.1 Extended feature space

This method was first proposed by Zhang et al. in (Zhang et al., 2012), which extends a multi-dimensional output variable by the help of binary variables. In other words, consider a sample $(\mathbf{x}_i, \mathbf{y}_i)$, where $\mathbf{x}_i \in \mathbb{R}^{n_x}$ and $\mathbf{y}_i \in \mathbb{R}^{n_y}$ are the input and output, then the sample is extended to n_y samples, i.e.

$$\begin{aligned} & (I_0, \mathbf{x}_i, y_{i0}) \\ & \vdots \\ & (\mathbf{x}_i, \mathbf{y}_i) \rightarrow (I_p, \mathbf{x}_i, y_{ip}) \\ & \vdots \\ & (I_{n_y-1}, \mathbf{x}_i, y_{i(n_y-1)}), \end{aligned}$$

where $I_{p(p=0,\dots,n_y-1)}$ represents a binary vector whose elements except p^{th} element is 0, and $y_{ip(p=0,\dots,n_y-1)}$ is the p^{th} element of the output vector. The regression function $f_p : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ for this extended feature space is defined as

$$f_p = \mathbf{w}_p^\top \phi(I_p, \mathbf{x}) + I_p^\top \mathbf{b}, \quad (3.8)$$

where $\mathbf{w} \in \mathbb{R}^m$ is the weight, $\phi(I_p, \mathbf{x}) : \mathbb{R}^{n_y \times n_x} \rightarrow \mathbb{R}^m$ is the extended feature function, and $\mathbf{b} = (b_0, b_1, \dots, b_{n_y-1})^\top \in \mathbb{R}^{n_y}$. This function is solved by formulating into the following quadratic optimization problem:

$$\begin{aligned} \text{minimize } J(\mathbf{w}, \boldsymbol{\xi}) &= \text{minimize} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 + \frac{C}{2} \sum_{i=0}^{N-1} \sum_{p=0}^{n_y-1} e_{ip}^2 \right\} \\ \text{subject to } y_{ip} &= \mathbf{w}_p^\top \phi(I_p, \mathbf{x}_i) + \mathbf{b} + e_{ip} \quad \forall i = 0 \dots N-1, \forall p = 0 \dots n_y-1, \end{aligned} \quad (3.9)$$

where C is known as tradeoff factor, and $\mathbf{w} = (\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{n_y}) \in \mathbb{R}^{m \times n_y}$.

3.3.2 Multi-output LS-SVR (MLS-SVR)

This method uses the concepts from the multi-task learning methods based on hierarchical Bayesian methods (Bakker and Heskes, 2003). It was first presented by Xu et al. in (Xu et al., 2013), and in this method, the weight matrix $\mathbf{W} \in \mathbb{R}^{m \times N}$ consists of vectors that carries commonality and speciality information is defined. In other words, $\mathbf{W} = [\hat{\mathbf{w}} + \mathbf{v}_0 \ \hat{\mathbf{w}} + \mathbf{v}_1 \ \dots \ \hat{\mathbf{w}} + \mathbf{v}_{N-1}]$ where $\hat{\mathbf{w}}$ represents the commonality information and $\mathbf{v}_{i(i=0\dots N-1)}$ represent the speciality information. The optimization problem formulated in this method can be represented as:

$$\begin{aligned} \text{minimize } J(\mathbf{W}, \mathbf{b}) &= \frac{1}{2} \{ \hat{\mathbf{w}}^\top \hat{\mathbf{w}} \} + \frac{1}{2} \frac{\lambda}{n_y} \text{trace} \{ \mathbf{V}^\top \mathbf{V} \} + \frac{\gamma}{2} \text{trace} \{ \boldsymbol{\Xi}^\top \boldsymbol{\Xi} \} \\ \text{subject to } \mathbf{Y}^\top &= \mathbf{Z}^\top \mathbf{W} + \mathbf{B} + \boldsymbol{\Xi}, \end{aligned} \quad (3.10)$$

where $\mathbf{Y} = [\mathbf{y}_0 \ \mathbf{y}_1 \ \dots \ \mathbf{y}_{N-1}]^\top \in \mathbb{R}^{n_y \times N}$, $\mathbf{V} = [\mathbf{v}_0 \ \mathbf{v}_1 \ \dots \ \mathbf{v}_{N-1}] \in \mathbb{R}^{m \times N}$, $\tilde{\mathbf{B}} = [\mathbf{b} \ \mathbf{b} \ \dots \ \mathbf{b}]^\top \in \mathbb{R}^{N \times n_y}$ is the matrix of biased term $\mathbf{b} \in \mathbb{R}^{n_y}$, and $\boldsymbol{\Xi} = [\xi_0 \ \xi_1 \ \dots \ \xi_{n_y-1}] \in \mathbb{R}^{N \times n_y}$ is the matrix of slack variables. Similarly, λ and γ are the positive real regularized parameters.

3.4 Subspace Identification Methods and Regression

In Chapter 2, subspace-based identification methods were covered, and the algorithms were generalized and summarised in Algorithm 1. The methods form block matrices \mathbf{Y}_f , \mathbf{Y}_p , \mathbf{U}_f , and \mathbf{U}_p from the input and output data sequences. Then oblique projections \mathcal{O}_h and \mathcal{O}_{h-1} are calculated using (2.10) and (2.15), which requires the calculation of orthogonal projections Y_f/U_f^\perp and W_p/U_f^\perp . The projections can be represented in terms of regressions, which will be discussed in the following section.

3.5 Least Square Support Vector Regression approach for

$$\mathbf{Y}_f/\mathbf{U}_f^\perp \text{ and } \mathbf{W}_p/\mathbf{U}_f^\perp$$

From the definition of orthogonal projection, $\mathbf{Y}_f/\mathbf{U}_f^\perp$ can be represented as

$$\begin{aligned} \mathbf{Y}_f/\mathbf{U}_f^\perp &= \mathbf{Y}_f - \mathbf{Y}_f \mathbf{U}_f^\top (\mathbf{U}_f \mathbf{U}_f^\top)^{-1} \mathbf{U}_f \\ &= \mathbf{Y}_f - \hat{\mathbf{H}}_f \mathbf{U}_f, \end{aligned} \quad (3.11)$$

where $\hat{\mathbf{H}}_f$ is the least square solution of $\mathbf{Y}_f = \mathbf{H}_f \mathbf{U}_f$. In other words

$$\hat{\mathbf{H}}_f = \arg \min_{\mathbf{H}_f} \|\mathbf{Y}_f - \mathbf{H}_f \mathbf{U}_f\|_2^2. \quad (3.12)$$

Therefore, Equation (3.11) can be represented as

$$\mathbf{Y}_f/\mathbf{U}_f^\perp = \mathbf{Y}_f - \hat{\mathbf{Y}}_f, \quad (3.13)$$

where $\hat{\mathbf{Y}}_f$ is the least square estimate of $\mathbf{Y}_f = \mathbf{H}_f \mathbf{U}_f$.

In this chapter, $\hat{\mathbf{Y}}_f$ is proposed to be estimated using LS-SVR, i.e.

$$\hat{\mathbf{y}}_k = \mathbf{W}^\top \phi(\tilde{\mathbf{u}}_k) + \boldsymbol{\delta}, \quad \forall k = 0 \dots N-1 \quad (3.14)$$

where $\hat{\mathbf{y}}_{k(k=0\dots N-1)} \in \mathbb{R}^{n_y h}$ is the k^{th} column of the estimate of $\hat{\mathbf{Y}}_f$, i.e. $\hat{\mathbf{Y}}_f = [\hat{\mathbf{y}}_0 \hat{\mathbf{y}}_1 \hat{\mathbf{y}}_2 \dots \hat{\mathbf{y}}_k \dots \hat{\mathbf{y}}_{N-1}]$, $\tilde{\mathbf{u}}_k \in \mathbb{R}^{n_u h}$ is the k^{th} column of \mathbf{U}_f , $\phi(\mathbf{u}_k) : \mathbb{R}^{n_u h} \rightarrow \mathbb{R}^M$ maps the input space into a higher dimensional space, and M is the dimension of the space. Similarly, $\mathbf{W} \in \mathbb{R}^{M \times n_y h}$ is the weight matrix, and $\boldsymbol{\delta} \in \mathbb{R}^{n_y h}$ is the bias compensating term.

It should be noted that in (3.14) the output variable $\hat{\mathbf{y}}_{k(k=0\dots N-1)}$ is a multi-dimensional variable. Hence, the standard formulation of LS-SVR, i.e. (3.2), cannot be applied in this case. To address the issue, a LS-SVR for multi-output should be developed. For that, MLS-SVR (Section 3.3.2) is considered, which leads to the following multi-output objective function:

$$\text{minimize } J(\mathbf{W}, \boldsymbol{\xi}_p) = \frac{1}{2} \text{trace} \{ \mathbf{W}^\top \mathbf{W} \} + \frac{\gamma}{2} \sum_{i=0}^{N-1} \boldsymbol{\xi}_i^2 \quad (3.15)$$

$$\text{subject to } \tilde{\mathbf{y}}_i = \mathbf{W}^\top \phi(\mathbf{u}_i) + \boldsymbol{\delta} + \boldsymbol{\xi}_i, \quad \forall i = 0 \dots N-1,$$

where $\tilde{\mathbf{y}}_i$ is the i^{th} column of \mathbf{Y}_f , i.e. $\mathbf{Y}_f = [\tilde{\mathbf{y}}_0 \tilde{\mathbf{y}}_1 \tilde{\mathbf{y}}_2 \cdots \tilde{\mathbf{y}}_{N-1}]$ and $\boldsymbol{\xi}_k \in \mathbb{R}^{hn_y}$ is the error vector which is assumed to be randomly distributed and uncorrelated with $\tilde{\mathbf{u}}_k$.

It should be noted that the objective function defined in Section 3.3.2 consists of a biased term in \mathbf{W} , i.e. $\mathbf{W} = [\hat{\mathbf{w}} + \mathbf{v}_0 \hat{\mathbf{w}} + \mathbf{v}_1 \dots \hat{\mathbf{w}} + \mathbf{v}_{N-1}]$ with $\hat{\mathbf{w}}$ known as bias. However, to simplify (3.15), terms are omitted. The solution for the primal problem is presented in the following lemma.

Lemma 1: Consider the primal problem (3.15) with the Lagrangian defined as

$$\mathcal{L} = J(\mathbf{W}, \boldsymbol{\xi}_p) - \sum_{i=0}^{N-1} \boldsymbol{\alpha}_i^\top \{ \mathbf{W}^\top \boldsymbol{\phi}(\mathbf{u}_i) + \boldsymbol{\delta} + \boldsymbol{\xi}_i - \tilde{\mathbf{y}}_i \} \quad (3.16)$$

where $\boldsymbol{\alpha}_{i(i=0\dots N-1)} \in \mathbb{R}^{n_y h}$ are the Lagrangian multipliers. The solution to the primal problem is given by

$$\begin{bmatrix} 0 & \mathbf{1}_{1 \times N} \\ \mathbf{1}_{N \times 1} & \boldsymbol{\kappa} + \frac{1}{\gamma} I_{N \times N} \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta}_i \\ \tilde{\boldsymbol{\alpha}}_j \end{bmatrix} = \begin{bmatrix} 0 \\ \bar{\mathbf{y}}_j \end{bmatrix}, \quad \forall j = 0 \dots n_y h - 1, \quad (3.17)$$

where $\boldsymbol{\kappa}$ is called the Mercer kernel function with $\kappa_{ij} = \boldsymbol{\phi}^\top(\mathbf{u}_i) \boldsymbol{\phi}(\mathbf{u}_j)$, $\tilde{\boldsymbol{\alpha}}_j = [\alpha_{0j} \alpha_{1j} \alpha_{2j} \alpha_{(N-1)j}]^\top \in \mathbb{R}^N$, $\bar{\mathbf{y}}_j \in \mathbb{R}^N$ is the j^{th} row of the block matrix \mathbf{Y}_f , i.e. $\mathbf{Y}_f = [\bar{\mathbf{y}}_0 \bar{\mathbf{y}}_1 \dots \bar{\mathbf{y}}_{N-1}]^\top \in \mathbb{R}^{n_y h \times N}$.

Proof: By applying the KKT condition for optimality, one obtains:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\delta}} = 0 \rightarrow \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial \delta_0} \\ \frac{\partial \mathcal{L}}{\partial \delta_1} \\ \vdots \\ \frac{\partial \mathcal{L}}{\partial \delta_{n_y h - 1}} \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{N-1} \alpha_{i0} \\ \sum_{i=0}^{N-1} \alpha_{i1} \\ \vdots \\ \sum_{i=0}^{N-1} \alpha_{in_y h - 1} \end{bmatrix} = 0, \quad (3.18)$$

or

$$[0 \mathbf{1}_{1 \times N}] \begin{bmatrix} \boldsymbol{\delta}_i \\ \tilde{\boldsymbol{\alpha}}_i \end{bmatrix} = \mathbf{0}_{(N+1) \times 1}, \quad \forall i = 0 \dots N - 1. \quad (3.19)$$

Similarly,

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\xi}_i} = 0 \rightarrow \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial \xi_{i1}} \\ \frac{\partial \mathcal{L}}{\partial \xi_{i2}} \\ \vdots \\ \frac{\partial \mathcal{L}}{\partial \xi_{iph}} \end{bmatrix} = \begin{bmatrix} \alpha_{i1}/\gamma \\ \alpha_{i2}/\gamma \\ \vdots \\ \alpha_{iph}/\gamma \end{bmatrix} = 0, \quad (3.20)$$

or,

$$\boldsymbol{\xi}_i = \frac{1}{\gamma} \boldsymbol{\alpha}_i, \quad \forall i = 0 \dots N-1. \quad (3.21)$$

Furthermore, by applying $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\alpha}_i} = 0$, we have

$$\mathbf{W}^\top \boldsymbol{\phi}(\mathbf{u}_i) + \boldsymbol{\delta} + \boldsymbol{\xi}_i - \mathbf{y}_i = 0, \quad \forall i = 0 \dots N-1. \quad (3.22)$$

Finally, condition $\frac{\partial \mathcal{L}}{\partial \mathbf{W}} = 0$ leads to

$$\mathbf{W}^\top = [\boldsymbol{\alpha}_0 \boldsymbol{\alpha}_1 \boldsymbol{\alpha}_2 \dots \boldsymbol{\alpha}_{N-1}] \begin{bmatrix} \boldsymbol{\phi}^\top(\mathbf{u}_0) \\ \boldsymbol{\phi}^\top(\mathbf{u}_1) \\ \vdots \\ \boldsymbol{\phi}^\top(\mathbf{u}_{N-1}) \end{bmatrix}. \quad (3.23)$$

Now, substituting (3.21) and (3.23) into (3.22), one obtains:

$$[\boldsymbol{\alpha}_0 \boldsymbol{\alpha}_1 \boldsymbol{\alpha}_2 \dots \boldsymbol{\alpha}_{N-1}] \begin{bmatrix} \boldsymbol{\phi}^\top(\mathbf{u}_0) \\ \boldsymbol{\phi}^\top(\mathbf{u}_1) \\ \vdots \\ \boldsymbol{\phi}^\top(\mathbf{u}_{N-1}) \end{bmatrix} \boldsymbol{\phi}(\mathbf{u}_i) + \boldsymbol{\delta} + \frac{1}{\gamma} \boldsymbol{\alpha}_i = \hat{\mathbf{y}}_i,$$

or

$$[\boldsymbol{\alpha}_0 \boldsymbol{\alpha}_1 \boldsymbol{\alpha}_2 \dots \boldsymbol{\alpha}_{N-1}] \begin{bmatrix} \boldsymbol{\phi}^\top(\mathbf{u}_0)\boldsymbol{\phi}(\mathbf{u}_i) \\ \boldsymbol{\phi}^\top(\mathbf{u}_1)\boldsymbol{\phi}(\mathbf{u}_i) \\ \vdots \\ \boldsymbol{\phi}^\top(\mathbf{u}_{N-1})\boldsymbol{\phi}(\mathbf{u}_i) \end{bmatrix} + \boldsymbol{\delta} + \frac{1}{\gamma} \boldsymbol{\alpha}_i = \hat{\mathbf{y}}_i \quad \forall i = 0 \dots N-1 \quad (3.24)$$

By combining all the equations for $i = 0 \dots N-1$ from (3.24) to form one equation of block matrices, we have

$$[\boldsymbol{\alpha}_0 \boldsymbol{\alpha}_1 \boldsymbol{\alpha}_2 \dots \boldsymbol{\alpha}_{N-1}] \boldsymbol{\kappa} + \boldsymbol{\Xi} + \frac{1}{\gamma} [\boldsymbol{\alpha}_0 \boldsymbol{\alpha}_1 \boldsymbol{\alpha}_2 \dots \boldsymbol{\alpha}_{N-1}] I_{N \times N} = \mathbf{Y}_f, \quad (3.25)$$

where

$$\boldsymbol{\kappa} = \begin{bmatrix} \boldsymbol{\phi}^\top(\mathbf{u}_0)\boldsymbol{\phi}(\mathbf{u}_0) & \boldsymbol{\phi}^\top(\mathbf{u}_0)\boldsymbol{\phi}(\mathbf{u}_1) & \dots & \boldsymbol{\phi}^\top(\mathbf{u}_0)\boldsymbol{\phi}(\mathbf{u}_{N-1}) \\ \boldsymbol{\phi}^\top(\mathbf{u}_1)\boldsymbol{\phi}(\mathbf{u}_0) & \boldsymbol{\phi}^\top(\mathbf{u}_1)\boldsymbol{\phi}(\mathbf{u}_1) & \dots & \boldsymbol{\phi}^\top(\mathbf{u}_1)\boldsymbol{\phi}(\mathbf{u}_{N-1}) \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\phi}^\top(\mathbf{u}_{N-1})\boldsymbol{\phi}(\mathbf{u}_0) & \boldsymbol{\phi}^\top(\mathbf{u}_{N-1})\boldsymbol{\phi}(\mathbf{u}_1) & \dots & \boldsymbol{\phi}^\top(\mathbf{u}_{N-1})\boldsymbol{\phi}(\mathbf{u}_{N-1}) \end{bmatrix},$$

$$\Xi = [\boldsymbol{\delta} \ \boldsymbol{\delta} \ \dots \ \boldsymbol{\delta}],$$

and

$$\mathbf{Y}_f = [\tilde{\mathbf{y}}_0 \ \tilde{\mathbf{y}}_1 \ \tilde{\mathbf{y}}_2 \ \dots \ \tilde{\mathbf{y}}_{N-1}].$$

Now, taking the transpose on both sides of Equation (3.25) one obtains

$$(\boldsymbol{\kappa} + \frac{1}{\gamma}I) [\tilde{\boldsymbol{\alpha}}_0 \ \tilde{\boldsymbol{\alpha}}_1 \ \tilde{\boldsymbol{\alpha}}_2 \ \dots \ \tilde{\boldsymbol{\alpha}}_{n_y h - 1}] + \Xi^\top = \mathbf{Y}_f^\top,$$

which can also be rewritten as

$$\begin{bmatrix} \mathbf{1}_{N \times 1} & \boldsymbol{\kappa} + \frac{1}{\gamma}I \end{bmatrix} \begin{bmatrix} \delta_i \\ \tilde{\boldsymbol{\alpha}}_i \end{bmatrix} = \bar{\mathbf{y}}_i, \quad i = 0 \dots n_y h - 1, \quad (3.26)$$

where $\mathbf{1}_{N \times 1} = [1 \ 1 \ \dots \ 1]^\top \in \mathbb{R}^{N \times 1}$. Therefore, by combining (3.19) and (3.26), one can finally have

$$\begin{bmatrix} 0 & \mathbf{1}_{1 \times N} \\ \mathbf{1}_{N \times 1} & \boldsymbol{\kappa} + \frac{1}{\gamma}I_{N \times N} \end{bmatrix} \begin{bmatrix} \delta_i \\ \tilde{\boldsymbol{\alpha}}_j \end{bmatrix} = \begin{bmatrix} 0 \\ \bar{\mathbf{y}}_j \end{bmatrix}, \quad \forall j = 1 \dots n_y h - 1.$$

This completes the proof.

Now the proposed method of calculating orthogonal projection $\mathbf{Y}_f/\mathbf{U}_f^\perp$ using LS-SVR can be formulated in the following theorem.

Theorem 1: Given block Hankel matrices \mathbf{Y}_f and \mathbf{U}_f , which are formed from the input and output sequences, and consider the orthogonal projection $\mathbf{Y}_f/\mathbf{U}_f^\perp$ expressed in the form (3.13), i.e.

$$\mathbf{Y}_f/\mathbf{U}_f^\perp = \mathbf{Y}_f - \hat{\mathbf{Y}}_f,$$

where $\hat{\mathbf{Y}}_f$ represents the projection $\mathbf{Y}_f/\mathbf{U}_f$ or a least square estimate of $\mathbf{H}_f\mathbf{U}_f$. Then, the block matrix $\hat{\mathbf{Y}}_f$ can be obtained as a solution of LS-SVR regression, i.e.

$$\hat{\mathbf{Y}}_f = \Psi \kappa + \Xi, \quad (3.27)$$

where κ is a matrix of kernel functions and $\Xi = [\boldsymbol{\delta} \boldsymbol{\delta} \boldsymbol{\delta} \cdots \boldsymbol{\delta}] \in \mathbb{R}^{n_y h \times N}$ is a matrix of the bias vector $\boldsymbol{\delta}$. Hence, $\mathbf{Y}_f/\mathbf{U}_f^\perp$ can be estimated in the LS-SVR sense.

Proof: Consider that each columns of \mathbf{Y}_f matrix is represented as (3.14), i.e.

$$\hat{\mathbf{y}}_i = \mathbf{W}^\top \phi(\tilde{\mathbf{u}}_i) + \boldsymbol{\delta},$$

where the terms \mathbf{W}^\top , $\phi(\tilde{\mathbf{u}}_i)$ and $\boldsymbol{\delta}$ have their usual meanings which are already defined. Now, to solve for \mathbf{W} and $\boldsymbol{\delta}$ in (3.14), one can formulate the optimal primal problem (3.15). By using Lemma 1 the optimization problem can be solved by solving (3.17), i.e.

$$\begin{bmatrix} 0 & \mathbf{1}_{1 \times N} \\ \mathbf{1}_{N \times 1} & \kappa + \frac{1}{\gamma} I_{N \times N} \end{bmatrix} \begin{bmatrix} \delta_i \\ \tilde{\boldsymbol{\alpha}}_j \end{bmatrix} = \begin{bmatrix} 0 \\ \bar{\mathbf{y}}_j \end{bmatrix}, \quad \forall j = 1 \dots n_y h - 1.$$

Now, by using the kernel trick (Pelckmans et al., 2005) one can replace $\mathbf{W}^\top \phi(\tilde{\mathbf{u}}_i)$ in (3.14) by $\Psi \kappa_i$ which leads to

$$\hat{\mathbf{y}}_i = \Psi \kappa_i + \boldsymbol{\delta}, \quad \forall i = 0 \dots N - 1 \quad (3.28)$$

where κ_i is the i^{th} column of κ matrix, and $\Psi = [\boldsymbol{\alpha}_0 \boldsymbol{\alpha}_1 \boldsymbol{\alpha}_2 \dots \boldsymbol{\alpha}_{N-1}]$. It is noted that (3.28) removes the necessity to calculate \mathbf{W} and ϕ (Pelckmans et al., 2005). Now, by combining all the vectors of $\hat{\mathbf{y}}_i$ into a block matrix, one obtains (3.27), i.e.

$$\hat{\mathbf{Y}}_f = \Psi \kappa + \Xi.$$

Remark 1: There are various types of kernel functions, such as

- linear kernel $\kappa(\tilde{\mathbf{u}}_i, \tilde{\mathbf{u}}_j) = \tilde{\mathbf{u}}_i^\top \tilde{\mathbf{u}}_j$

- polynomial kernel function $\kappa(\tilde{\mathbf{u}}_i, \tilde{\mathbf{u}}_j) = (\tau + \tilde{\mathbf{u}}_i^\top \tilde{\mathbf{u}}_j)^{n_d}$ where n_d is the order of polynomial
- RBF kernel function $\kappa(\tilde{\mathbf{u}}_i, \tilde{\mathbf{u}}_j) = \exp(-\|\tilde{\mathbf{u}}_i - \tilde{\mathbf{u}}_j\|_2^2/\sigma)$ where σ is known as the bandwidth.

Here, the linear kernel function is considered for simplicity.

Finally, the orthogonal projections $\mathbf{W}_p/\mathbf{U}_f^\perp$, $\mathbf{Y}_f^-/\mathbf{U}_f^{-\perp}$, and $\mathbf{W}_p^+/\mathbf{U}_f^\perp$ that are necessary in the subspace-based identification methods can be calculated in a similar fashion, i.e.

$$\begin{aligned}\mathbf{W}_p/\mathbf{U}_f^\perp &= \mathbf{W}_p - \hat{\mathbf{W}}_p, \\ \mathbf{Y}_f^-/\mathbf{U}_f^{-\perp} &= \mathbf{Y}_f^- - \hat{\mathbf{Y}}_f^-, \\ \mathbf{W}_p^+/\mathbf{U}_f^\perp &= \mathbf{W}_p^+ - \hat{\mathbf{W}}_p^+, \end{aligned}\tag{3.29}$$

where $\hat{\mathbf{W}}_p$, $\hat{\mathbf{Y}}_f^-$, and $\hat{\mathbf{W}}_p^+$ are the LS-SVR estimations of $\mathbf{W}_p/\mathbf{U}_f$, $\mathbf{Y}_f^-/\mathbf{U}_f$, and $\mathbf{W}_p^+/\mathbf{U}_f$, respectively. In other words,

$$\begin{aligned}\hat{\mathbf{W}}_p &= \Psi_{\mathbf{W}_p} \kappa_{\mathbf{W}_p} + \Xi_{\mathbf{W}_p}, \\ \hat{\mathbf{Y}}_f^- &= \Psi_{\mathbf{Y}_f^-} \kappa_{\mathbf{Y}_f^-} + \Xi_{\mathbf{Y}_f^-}, \\ \hat{\mathbf{W}}_p^+ &= \Psi_{\mathbf{W}_p^+} \kappa_{\mathbf{W}_p^+} + \Xi_{\mathbf{W}_p^+}, \end{aligned}\tag{3.30}$$

where $\kappa_{\mathbf{W}_p} \in \mathbb{R}^{N \times N}$, $\kappa_{\mathbf{Y}_f^-} \in \mathbb{R}^{N \times N}$, $\kappa_{\mathbf{W}_p^+} \in \mathbb{R}^{N \times N}$, $\Psi_{\mathbf{W}_p} \in \mathbb{R}^{(n_y+n_u)h \times N}$, $\Psi_{\mathbf{Y}_f^-} \in \mathbb{R}^{n_y(h-1) \times N}$, $\Psi_{\mathbf{W}_p^+} \in \mathbb{R}^{(n_y+n_u)(h+2) \times N}$, $\Xi_{\mathbf{W}_p} \in \mathbb{R}^{(n_y+n_u)h \times N}$, $\Xi_{\mathbf{Y}_f^-} \in \mathbb{R}^{n_y(h-1) \times N}$, and $\Xi_{\mathbf{W}_p^+} \in \mathbb{R}^{(n_y+n_u)(h+2) \times N}$.

The proposed method can now be summarized into the following step-by-step Algorithm.

Algorithm 3: Pseudocode for LS-SVR based identification method.

- 1 Form the block Hankel matrices \mathbf{Y}_f , \mathbf{U}_f , \mathbf{W}_p , and \mathbf{W}_p^+ from the input sequence $(\mathbf{u}[0], \mathbf{u}[1], \dots)$ and output sequence $(\mathbf{y}[0], \mathbf{y}[1], \dots)$.
 - 2 Estimate $\hat{\mathbf{Y}}_f$, $\hat{\mathbf{W}}_p$, $\hat{\mathbf{Y}}_f^-$, and $\hat{\mathbf{W}}_p^+$ using the LS-SVR, i.e. Equation (3.27) and (3.30) respectively.
 - 3 Calculate $\mathbf{Y}_f/\mathbf{U}_f^\perp$, $\mathbf{W}_p/\mathbf{U}_f$, $\mathbf{Y}_f^-/\mathbf{U}_f$, and $\mathbf{W}_p^+/\mathbf{U}_f$, respectively, using (3.13) and (3.29).
 - 4 Calculate \mathcal{O}_h and \mathcal{O}_{h-1} using (2.10) and (2.15), respectively.
 - 5 Apply SVD on \mathcal{O}_h and \mathcal{O}_{h-1} and then calculate Γ_h , \mathbf{X}_h , Γ_{h-1} , and \mathbf{X}_{h+1} .
 - 6 Estimate \hat{A} , \hat{B} , \hat{C} , and \hat{D} in a least-square sense from $\mathbf{Y}_{h|h}$, $\mathbf{U}_{h|h}$, \mathbf{X}_h , and \mathbf{X}_{h+1} , respectively (Equation (2.17)).
-

3.6 Numerical Example

For numerical verification of our proposed method, let us consider a state space model (Wu et al., 2015) with following system matrices:

$$A = \begin{bmatrix} 0.67 & 0.67 & 0 & 0 \\ -0.67 & 0.67 & 0 & 0 \\ 0 & 0 & -0.67 & -0.67 \\ 0 & 0 & 0.67 & -0.67 \end{bmatrix},$$

$$B = \begin{bmatrix} 0.6598 & -0.5256 \\ 1.9698 & 0.4845 \\ 4.3171 & -0.4879 \\ -2.6436 & -0.3416 \end{bmatrix},$$

$$C = \begin{bmatrix} -0.5749 & -1.0751 & -0.5225 & 0.1830 \\ -0.2977 & -0.1543 & 0.1159 & 0.0982 \end{bmatrix},$$

and

$$D = \begin{bmatrix} -0.7139 & -0.1174 \\ 0.3131 & -0.2876 \end{bmatrix}.$$

In addition, let us compare the performance of our proposed method (Algorithm (3)) with MOESP and CVA. Also Subspace Identification Methods using Principal Component Analysis and Error estimation (SIMPCA-E) (Wu et al., 2015) is considered for comparison which is a varient of SIMPCA that was presented in Chapter 2. Furthermore, to better evaluate the algorithms are compared in the following three scenarios.

3.6.1 Scenario 1

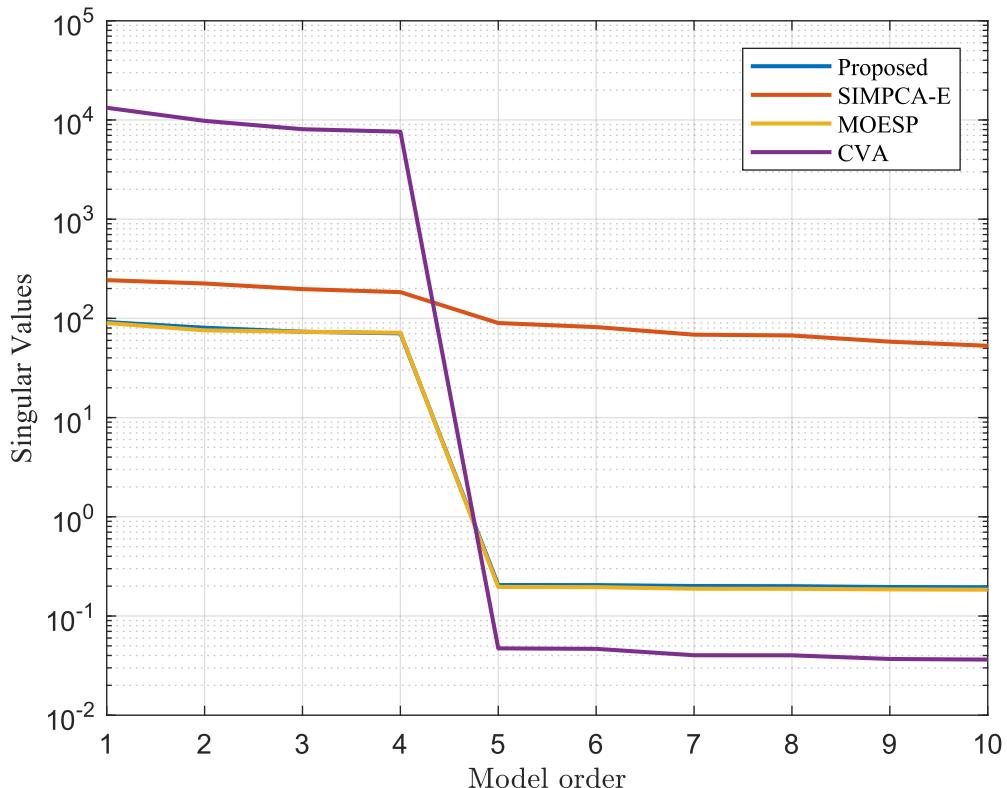


Figure 3.1 : Singular values of the SVD of \mathcal{O}_h for various methods

In this scenario, a Gaussian distributed random signal is applied to the system with the standard deviation of 1. Measurement noise considered for this case is low, i.e. $\sigma_v = 0.01$. Here, the horizon h is set to 10 and range of N is between 600 and 700. In addition, five experiments are conducted for each N .

Figure 3.1 shows the singular values of \mathcal{O}_h obtained from all the considered methods using (2.12). The values provide important information which helps to determine the order of the system. For instance, there is a significant change in the singular values, when the order of the system changes from 4 to 5. Therefore, as identified in (Wu et al., 2015), we assume that the order of the system is 4 and use the value to calculate the system's matrices A , B , C , and D .

Experimental results are presented in Fig. 3.2 which shows all the estimated poles. In the figure, the actual poles are represented by a cross located at $0.67 \pm j0.67$, and $-0.67 \pm j0.67$, respectively. From the figures, it can be observed that the estimated poles are quite close to the actual poles. For example, the estimated pole for $0.67 + j0.67$ lies within the bounded rectangle of $0.66994 + j0.67006$ and $0.67008 + j0.66994$.

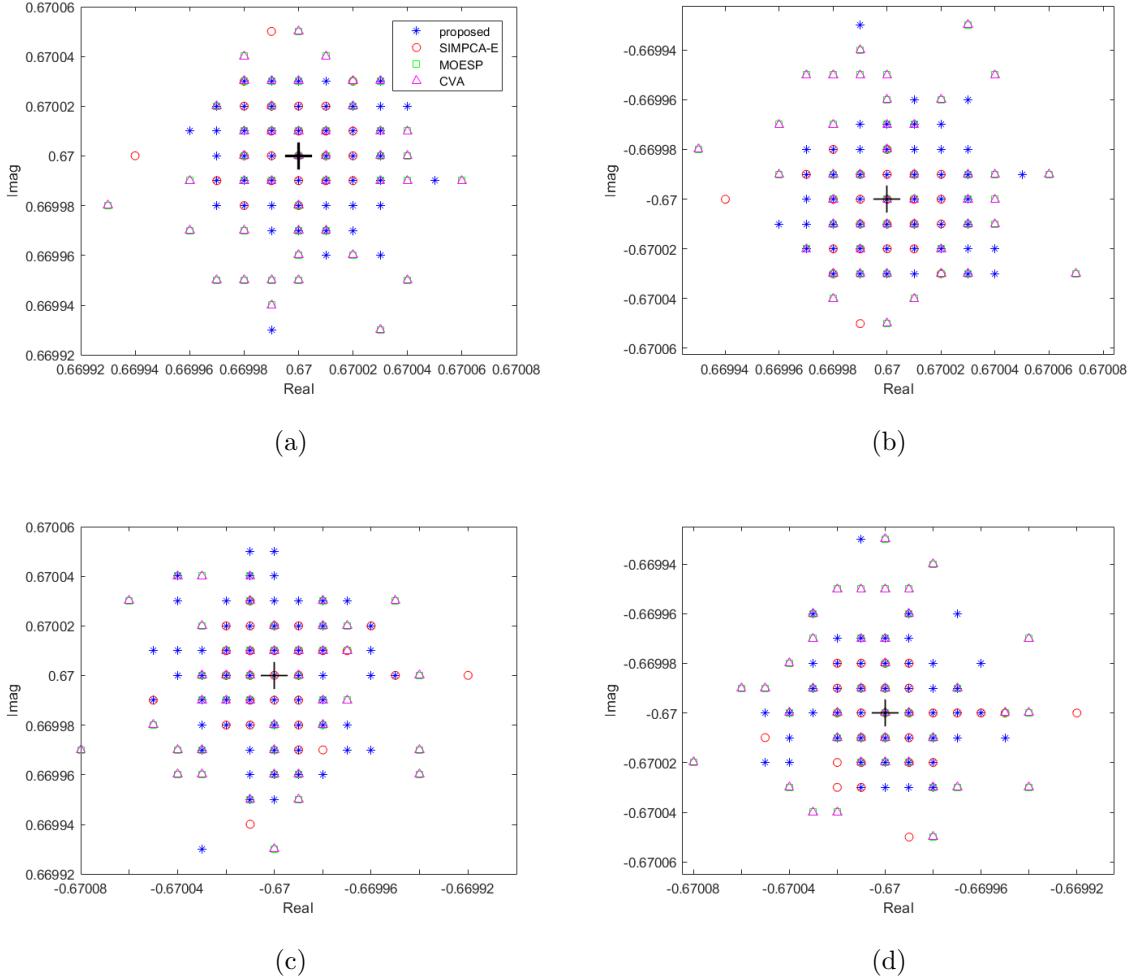


Figure 3.2 : Estimation of the poles located at a) $0.67 + j0.67$, b) $0.67 - j0.67$, c) $-0.67 + j0.67$, and d) $-0.67 - j0.67$.

To quantify the performance, let us consider the Root Mean Square Error (RMSE) which is defined as

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n_x} |\lambda_i - \lambda_i^*|^2}{n_x}}, \quad (3.31)$$

where $\lambda_{i(i=1 \dots n_x)}$ and $\lambda_{i(i=1 \dots n_x)}^*$ are respectively the estimated and actual poles. Figure 3.3 shows the mean and standard deviation of RMSE for $40 \leq N \leq 500$ and $h = 10$. From the figure, it is clear that RMSE is less than 1×10^{-4} for all methods. Nevertheless, the performance index clearly indicates the superiority of the

proposed algorithm compared to MOESP, CVA, and SIMPCA-E. For instance, for $N \geq 300$, the average RMSE for Algorithm 3 is less than 1×10^{-5} , whereas in case of the MOESP, CVA and SIMPCA-E, the RMSE values are 3×10^{-5} , 2×10^{-5} , and 5×10^{-5} , respectively.

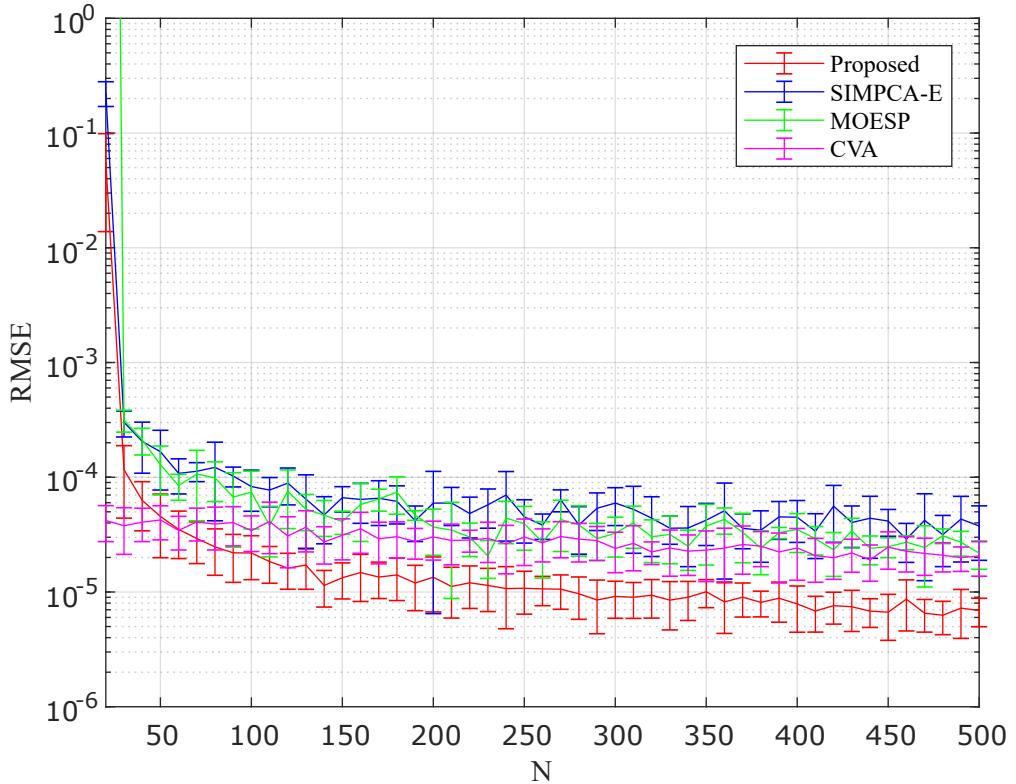


Figure 3.3 : Comparison of the RMSE for the Scenario 1

3.6.2 Scenario 2

In this case, the standard deviation of measurement noise is $\sigma_y = 0.4$ with the same setup as mentioned in previous case. The estimated poles are depicted in Fig. 3.4. Here, the estimated poles are also close to the actual poles. The estimated poles are within the bounded region of $0.668 + j0.6715$ and $0.6715 + j0.6685$ for the pole located at $0.67 + j0.67$. Similarly, the poles are also within the region of $0.668 - j0.668$ and $0.6715 - j0.6715$ for the original pole at $0.67 - j0.67$.

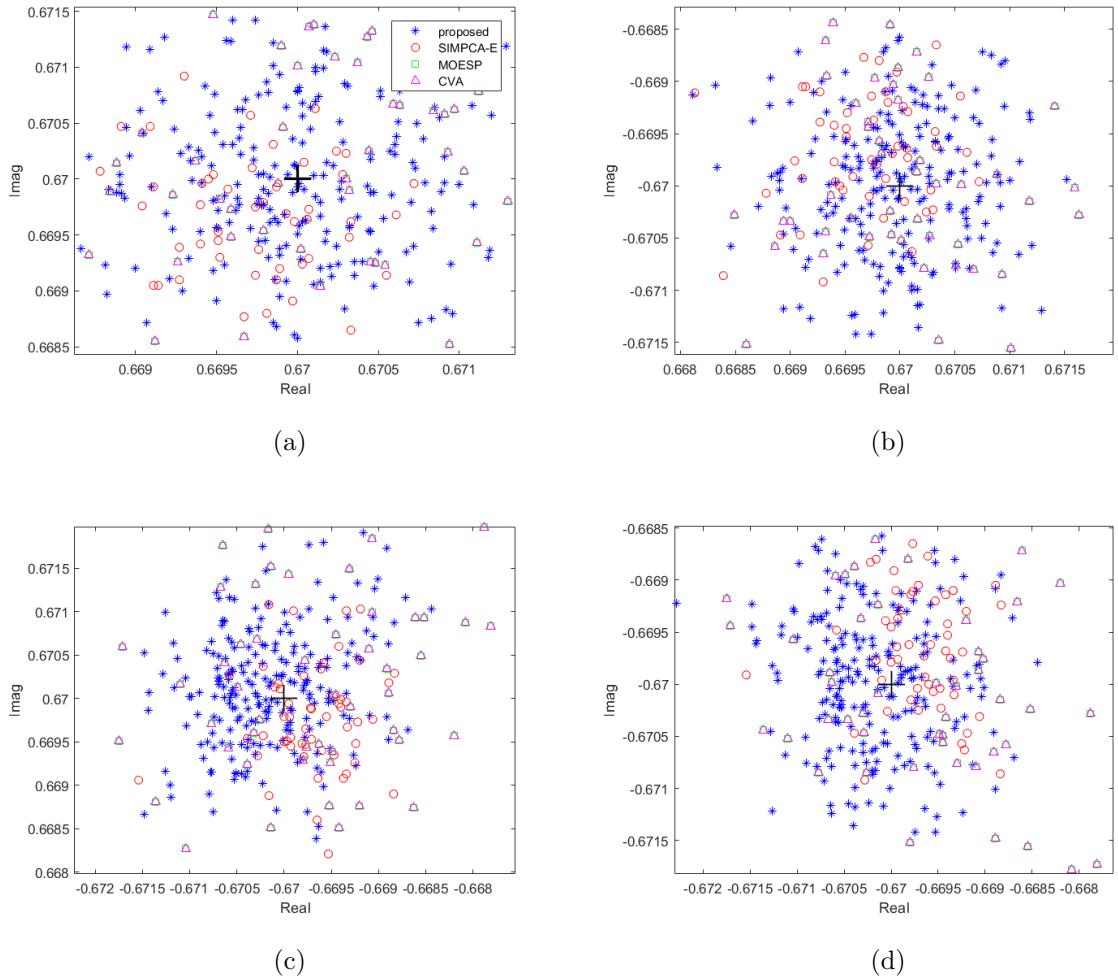
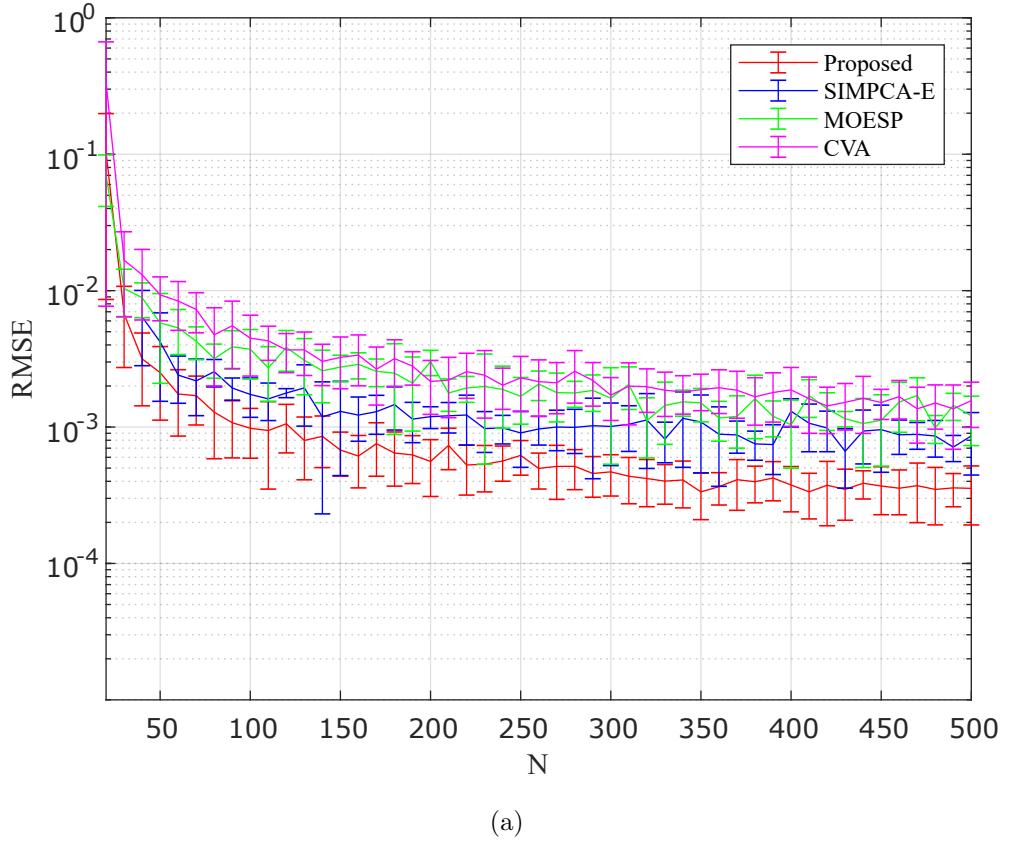


Figure 3.4 : Estimation of the poles for scenario 2.

The RMSE is presented in Fig. 3.5 for $20 \leq N \leq 500$. The result obtained is consistent with that of the previous scenario. Nevertheless, due to a higher noise level, the RMSE values for all the algorithms are higher. For instance, RMSE values are less than 1×10^{-2} compared to 1×10^{-4} for the previous case. Here, the algorithm 3 outperforms MOESP, CVA, and SIMPCA-E. For instance, for $N \geq 300$, RMSEs for the proposed is lower than 1×10^{-3} , compared to 2×10^{-3} for MOESP and CVA.

Figure 3.5 : RMSE with respect to N for scenario 2.

3.6.3 Scenario 3

In this scenario the following signals are applied to the inputs:

$$\begin{aligned} u_1[k] &= \frac{1}{A_1} \sum_{i=1}^{10} \sin(0.3898\pi ik) \\ u_2[k] &= \frac{1}{A_2} \sum_{i=1}^{10} \sin(0.4536\pi ik), \end{aligned} \quad (3.32)$$

where $A_{i(i=1,2)}$ are chosen such that the variance of the input signal is 1. In this case, the standard deviation of the measurement noise is also taken as 0.4. Figure 3.6 represents the estimated poles for $600 \leq N \leq 700$ and $h = 10$. From the figure it can be observed that the performance of the proposed and SIMPCA-E are better as compared to the MOESP and CVA. For instance, in case of the poles $0.67 - j0.67$ and $-0.67 + j0.67$, the estimation by the proposed and the SIMPCA-E appears to

be more accurate than MOESP and CVA.

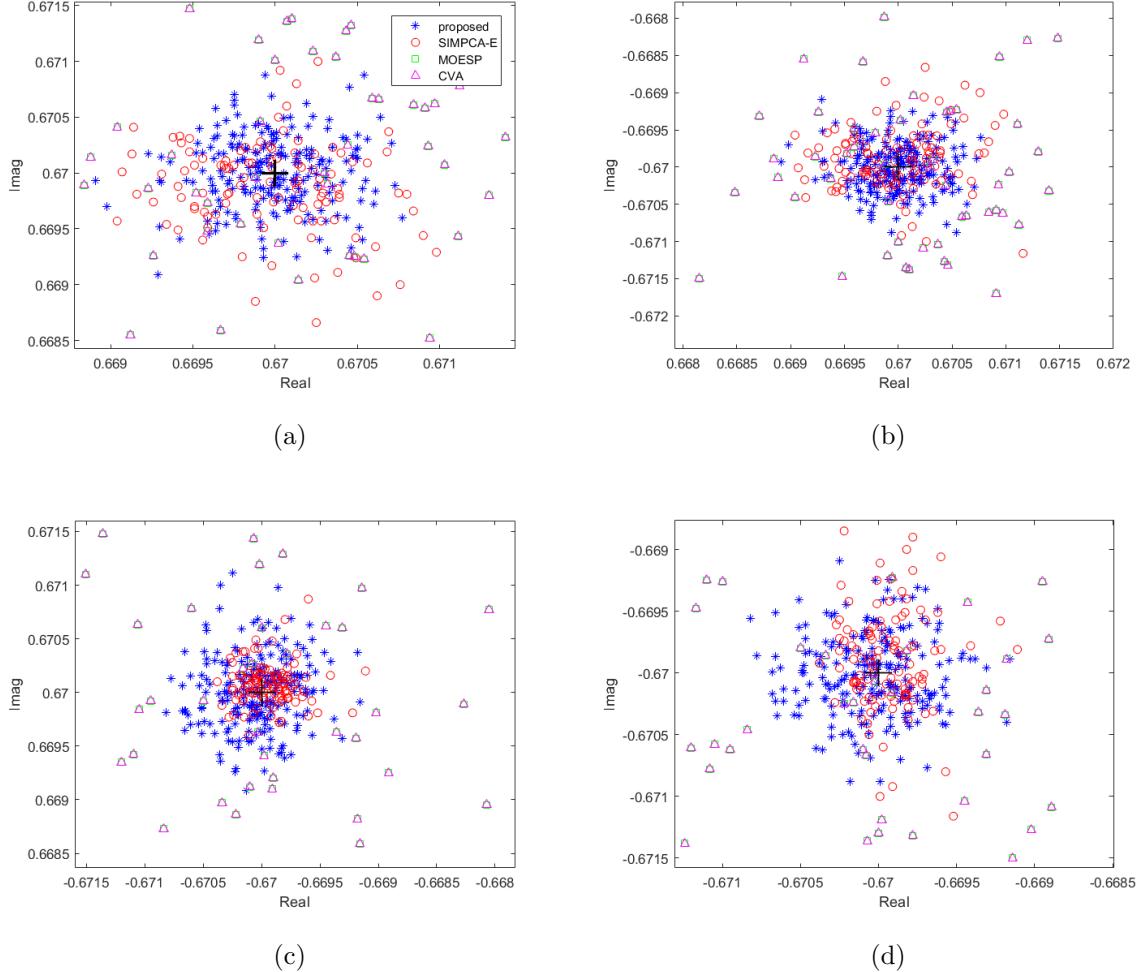


Figure 3.6 : Poles estimation of the system for scenario 3.

Figure 3.6 presents RMSE for all identification algorithms. The results are consistent with previous ones, which clearly indicate advantage of Pseudocode 3 performance for higher N . For instance, for $N \geq 300$, average RMSE here is around 3×10^{-4} , compared to greater than 4×10^{-4} for MOESP, CVA, and SIMPCA-E.

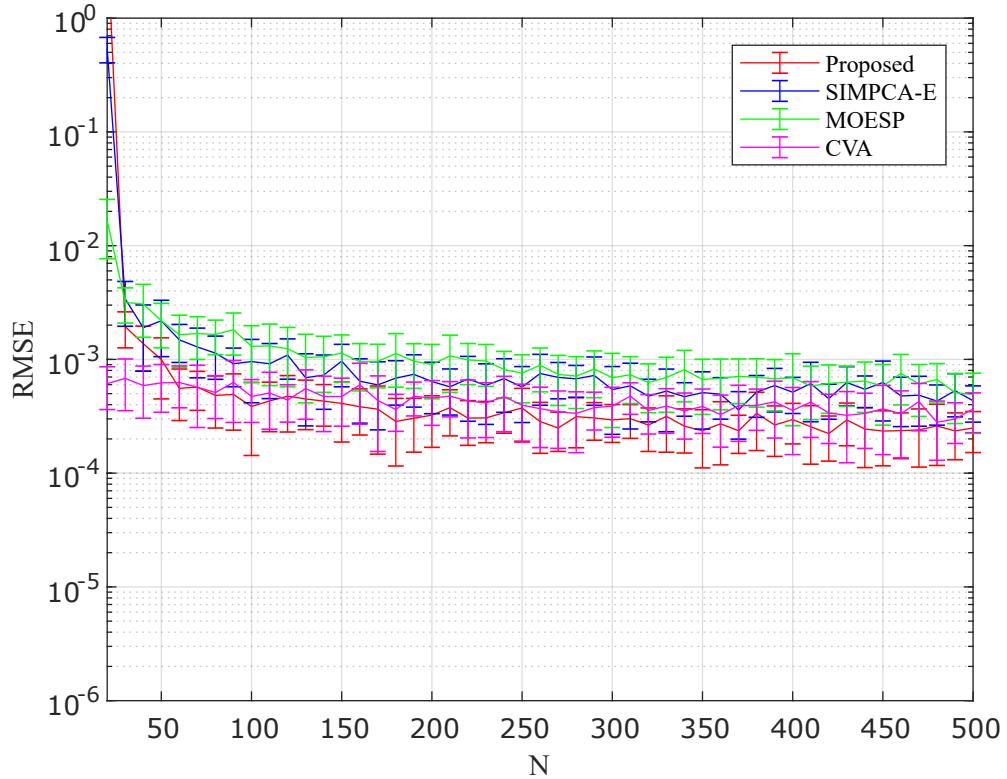


Figure 3.7 : RMSE for the scenario 3.

The comparison in terms of the singular values for scenario 2 and 3 is shown in Fig. 3.8. The results are consistent in this case, although due to the noise, the difference in singular values between 4 and 5 is smaller in comparison to scenario 1. It is noted in scenario 3, SIMPCA-E shows a small change which adversely affects in determining the order of the system.

In summary, these results confirm the effectiveness of the proposed method within the framework of subspace identification methods.

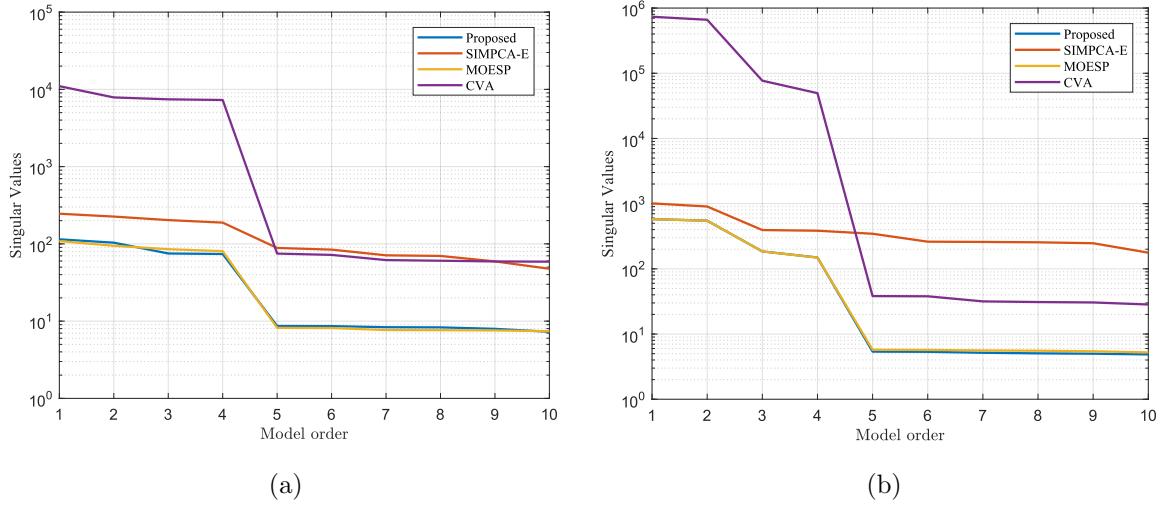


Figure 3.8 : Singular values for a) scenario 1, b) scenario 2.

3.7 Summary

An effective method for system identification using the Least Square Support Vector Regression (LS-SVR) has been developed in this chapter. The proposed method utilizes subspace based identification algorithms using the block Hankel input-output matrices from input and output data sequences. Here, the orthogonal projection of the output block matrix onto the input block matrix is proposed to calculate using LS-SVR. Since the block Hankel matrices involve multiple variables, the projection is formulated as a multi-output LS-SVR primal problem. After computing the projection matrices, the observability and Toeplitz matrices are obtained which leads to the calculation of system matrices A , B , C , and D matrices. For the verification of the proposed method, the algorithm is tested in a simulation environment. In addition, the proposed method is compared with MOESP, CVA and SIMPCA-E ones, which indicate that effectiveness of the estimated models by using the proposed method for identification in terms of RMSE performance index for all the scenarios.

Part II

Fast Terminal Sliding Mode Control

Chapter 4

Literature Survey on Fast Terminal Sliding Mode

4.1 Introduction

Advancement in every field such as robotics, manufacturing, aerospace, etc, has been possible due to the continuous development of automation technology. One of the fundamental components in automating a system is its control system. For instance, in a tower crane as shown in Fig. 4.1, its controller plays a big role in the picking and placing operation. Similarly, controllers are also used in the formation of unmanned aerial vehicles (UAV) for the purpose of structure inspection (Fig. 4.2).



Figure 4.1 : Tower crane working under harsh weather condition for the construction of buildings.

Performance of controllers is affected by various factors such as parametric variation, external disturbances, etc. For instance, in crane automation, winds affect the operation of cranes due to the induced oscillation of payloads. In these circumstances, controllers should be robust enough to provide immunity against such

disturbances.



Figure 4.2 : Bridge inspection using multiple UAVs in a formation (Source: Hoang et al. (2018)).

Among many control methodologies that deal with uncertainties, sliding mode control (SMC) is known for robustness. SMC is a type of variable structure control methods (Young et al., 1999). In this methodology, first, a hyperplane is designed for a system, which is also known as sliding surfaces. Then, a discontinuous control is applied to drive the system towards the plane. When the system is in the sliding surface it is known as sliding mode. SMC ideally provides immunity against the system uncertainties. However, there are some shortcomings in this approach. Particularly, it suffers from high-frequency noise in the input due to discontinuous control. This phenomenon is also known as chattering. However, numerous studies have been conducted to overcome the chattering issue.

In SMC, sliding surfaces affect the dynamics of a system. Conventionally, linear sliding surfaces are designed (Ngo and Hong, 2012), but such surfaces have asymptotic nature. As a result, a plant may take infinite time to reach its equilibrium. This may affect the performance of the system in high precision applications, e.g surgery using robotic manipulators. One of the solutions to address the issue is by

designing a nonlinear sliding surface which guarantees the stability of the system in finite-time. Such an approach is known as Terminal Sliding Mode (TSM) (Yu and Zhihong, 1998).

The finite-time reaching property of TSM can have an advantage in high precision applications. As a result, many researchers are interested in utilizing this method in various areas. For instance, utilization of this method in tracking the trajectory of robotic manipulators has been presented in (Zhihong et al., 1994; Jin et al., 2009). Similarly, usage of TSM in controlling the trajectory of spacecraft and their formation have been presented in (Zou et al., 2011; Hui and Li, 2009). Furthermore, researchers have also shown its application in the motion control of permanent magnet synchronous motors in (Feng et al., 2009; Li et al., 2013).

Sliding functions of TSM consist of a nonlinear term, known as the terminal attractor, which makes a system stable in a finite-time. However, when the system is far from equilibrium, it has a slower response. To address the issue a linear term is added into the sliding function of TSM, which improves the transient response of the system. This approach combines the sliding functions of TSM and SMC, and is known as the fast terminal sliding mode (FTSM) (Yu and Zhihong, 2002). FTSM improves the transient response of the system and also guarantee the finite time stability of the system.

4.2 Sliding Mode Control (SMC)

Consider a second-order nonlinear system described by the following state-space equation:

$$\ddot{x} = f_1(x, \dot{x}) + f_2(x)u + d(t), \quad (4.1)$$

where $x \in \mathbb{R}$ is the state of the system, $u \in \mathbb{R}$ is the input, and $d(t) \in \mathbb{R}$ is the disturbance to the system which is assumed to be bounded, i.e. $|d(t)| < \mu$. In SMC a sliding surface is designed at first followed by the control input synthesis (Shtessel

et al., 2014). Conventionally, the sliding manifolds are linear functions of the state variable, i.e.

$$s = \dot{x} + \alpha x, \quad (4.2)$$

where $\alpha > 0$. The synthesized control input for the system (4.1) consists of two components, namely the equivalent control (u_{eq}) and the switching control (u_{sw}), i.e.

$$u = u_{\text{eq}} + u_{\text{sw}}. \quad (4.3)$$

The equivalent control is obtained by substituting (4.1) into $\dot{s} = 0$, which leads to

$$u_{\text{eq}} = -f_2^{-1} \{ f_1(x, \dot{x}) + \alpha \dot{x} \}. \quad (4.4)$$

The objective of u_{eq} is to move the system to the sliding manifold, i.e. $s = 0$ (Young et al., 1999). This transient phase of the system is known as the reaching phase. Similarly, the dynamics of the system after reaching the sliding surface is governed by the equation $s = 0$, and this portion of the dynamics is known as sliding mode. In order to maintain the system in the sliding mode and overcome the disturbances, u_{sw} is applied to the system, which is a discontinuous function of s . The switching input is given by

$$u_{\text{sw}} = f_2^{-1} D_\mu \text{sign}(s), \quad (4.5)$$

where $D_\mu > \mu$. This switching control input, however, results into high frequency noise in the input. This phenomenon is known as chattering. There are many approaches to reduce the chattering. One such method is to replace the `sign()` by the following function (Yao and Tomizuka, 1996)

$$g(s) = \frac{s}{|s| + \epsilon_s}, \quad (4.6)$$

where ϵ_s is a small positive number.

When the system is in the sliding mode, its dynamics can be represented as

$$\dot{x} = -\alpha x. \quad (4.7)$$

Since $\alpha > 0$, the system is stable at the equilibrium point $x = 0$. However, it takes infinite time to reach the equilibrium. In other words,

$$\lim_{t \rightarrow \infty} x(t) = 0.$$

The main reason for such behavior is the linear sliding function (4.2), which results into linear system dynamics into the sliding function.

4.3 Terminal Sliding Mode (TSM)

As mentioned earlier, the limitations of the conventional SMC is that the state of the system reaches equilibrium asymptotically. In other words, it takes infinite time for x to reach equilibrium after the sliding mode has been achieved, i.e. $s = 0$. To work around the problem, a nonlinear function, also known as terminal attractor (Yu and Zhihong, 1996), is introduced into the sliding manifold as

$$s = \dot{x} + \beta x^{q/p}, \quad (4.8)$$

where $\beta > 0$, $q > 0$, and $p > 0$, in which q and p are odd integers such that $q < p$ (Park and Lee, 1996). One can obtain the analytical solution of $x(t)$ by solving the $\dot{x} = -\beta x^{\frac{q}{p}}$, which is given by

$$x(t) = \begin{cases} \text{sgn}(x_0) \left(x_0^{(p-q)/p} - \beta(1 - q/p) \right)^{\frac{p}{p-q}} & t \leq t_{TSM} \\ 0 & t > t_{TSM}, \end{cases} \quad (4.9)$$

where $x_0 = x(0)$. From the solution, it is evident that $x(t)$ reaches the equilibrium in finite time, i.e. t_{TSM} which is also known as the reaching time. The explicit form for the reaching time is given by

$$t_{TSM} = \frac{1}{\beta(1 - \frac{q}{p})} x_0^{(1-q/p)} \quad (4.10)$$

which can be obtained by solving for t from the differential equation $\dot{x} = -\beta x^{\frac{q}{p}}$. In other words,

$$\frac{dx}{dt} = -\beta x^{\frac{q}{p}}$$

or,

$$dt = -\frac{dx}{\beta x^{\frac{q}{p}}}$$

or,

$$t = - \int_{x_0}^{x_1} \frac{dx}{\beta x^{\frac{q}{p}}} - \frac{1}{\beta \left(1 - \frac{q}{p}\right)} x^{\left(1 - \frac{q}{p}\right)} \Big|_{x_0}^{x_1} \quad (4.11)$$

which leads to (4.10) since $x_1 = 0$. As reported by (Zak, 1988), the limiting slope of \dot{x} w.r.t x is infinite at equilibrium, i.e

$$\frac{d\dot{x}}{dx} \rightarrow -\infty \text{ as } x \rightarrow 0,$$

thus, the system's state reaches equilibrium in finite time. This concept is illustrated in the Fig. 4.3

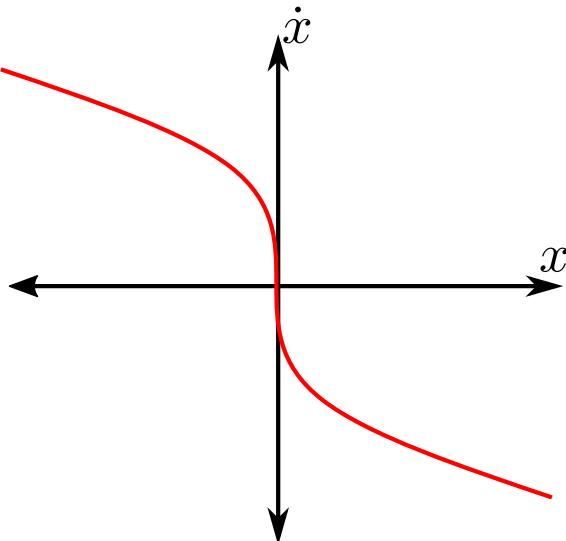


Figure 4.3 : Phase portrait of a system states in TSM manifold.

4.4 Fast Terminal Sliding Mode (FTSM)

In FTSM (Yu and Zhihong, 2002), the linear attractor term αx is added into the sliding function of TSM. The sliding manifold of FTSM is given by

$$s = \dot{x} + \beta x^{q/p} + \alpha x, \quad (4.12)$$

where $\alpha > 0$, $\beta > 0$, $q > 0$, and $p > 0$, in which q and p are odd integers such that $q < p$.

As in the case of TSM, one can explicitly represent the reaching time for FTSM as:

$$t_{FTSM} = \frac{p}{\alpha(p-q)} \ln \left(1 + \frac{\alpha x_0^{(p-q)/p}}{\beta} \right), \quad (4.13)$$

where $x_0 = x(0)$. Moreover, the FTSM dynamics can be obtained analytically in the sliding mode by solving the differential equation $\dot{x} + \beta x^{q/p} + \alpha x = 0$. An explicit form of the system state $x(t)$ is given by:

$$x(t) = \begin{cases} \operatorname{sgn}(x_0) h(t) & t \leq t_{FTSM} \\ 0 & t > t_{FTSM}, \end{cases}$$

where,

$$h(t) = \left(\frac{(\alpha x_0^{(p-q)/p} + \beta) e^{-\frac{\alpha(p-q)}{p} t} - \beta}{\alpha} \right)^{\frac{p}{p-q}}.$$

4.5 Comparison of TSM and FTSM

The intuitive interpretation of the sliding dynamics $s = 0$ of FTSM is that when the state x is far away from the equilibrium, the dynamics approximately becomes $\dot{x} = -\alpha x$, and then is reduced to $\dot{x} = -\beta x^{q/p}$ while close to $x = 0$ (Yu and Zhihong, 2002). As a result, the reaching time for the FTSM is always shorter than TSM. This can be confirmed by the following proposition.

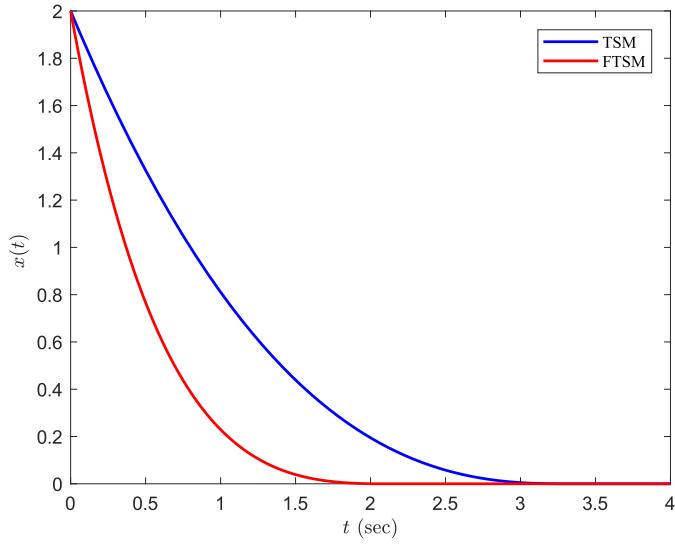


Figure 4.4 : Comparison of FTSM and TSM dynamics at the sliding manifold.

Proposition 1: For the sliding functions of FTSM and TSM, represented by equations (4.8) and (4.12), respectively, if the sliding mode is achieved, i.e. $s = 0$, then the time for FTSM to reach the equilibrium is less than that for TSM.

Proof: From the fact that $e^\xi > 1 + \xi$, $\forall \xi > 0$, we have

$$\exp\left(\frac{\alpha}{\beta}x_0^{(1-q/p)}\right) > 1 + \frac{\alpha}{\beta}x_0^{(1-q/p)}.$$

Thus,

$$\frac{\alpha}{\beta}x_0^{(1-q/p)} > \ln\left(1 + \frac{\alpha}{\beta}x_0^{(1-q/p)}\right).$$

By multiplying both sides with $\frac{1}{\alpha(1-\frac{q}{p})}$, we obtain:

$$\frac{1}{\beta(1-\frac{q}{p})}x_0^{(1-q/p)} > \frac{1}{\alpha(1-\frac{q}{p})}\ln\left(1 + \frac{\alpha}{\beta}x_0^{(1-q/p)}\right),$$

or

$$T_{TSM} > T_{FTSM}.$$

This completes the proof.

The comparison of TSM and FTSM in the sliding mode is presented in Fig. 4.4 for $\alpha = \beta = 1$, $q = 5$, and $p = 3$. The figure clearly indicates that the convergence of FTSM is faster than TSM.

4.6 Summary

In summary, a brief introduction on SMC, TSM and FTSM, which are known for robustness, has been presented in this chapter. Similarly, the advantage of terminal sliding surfaces in reaching the equilibrium by the system has also been demonstrated. Furthermore, the comparison of TSM and FTSM in the sliding manifold has indicated that FTSM has faster convergence rate compared to TSM.

Chapter 5

Fast Terminal Sliding Mode and Applications

5.1 Introduction

Underactuated systems such as cranes have fewer inputs than their degree of freedom, and they are widely used in construction, manufacturing, and so on. However, because of their underactuation and other uncertainties mentioned previously, there are numerous challenges in designing controllers for such systems. In a gantry crane, for instance, the tight coupling between its swing angle dynamics and the cart dynamics complicate the controller design process and its performance. Therefore, control of such system has been an increasingly important area in academics (Thibeault and Smith, 2001; Davila et al., 2005; Lee et al., 2009; Sun et al., 2015; Huang et al., 2017). For instance, in (Olfati-Saber, 2001) nonlinear control methods based on feedback linearization are presented. Similarly, dynamics of wheeled inverted pendulum, a type of underactuated system, is analysed, and control method based on partial feedback linearization is presented in (Pathak et al., 2005).

A great deal of previous research into underactuated system has also focused on implementing robust controller based on SMC. For example, in (Xu and Özgüner, 2008), a SMC-based controller is presented for a class of underactuated systems that are in cascaded form. Furthermore, SMC controller for underactuated wheeled inverted pendulum is presented in (Huang et al., 2010).

As presented in Chapter 4, FTSM is theoretically promising because it has faster response and it can provide stability in finite-time. However, it has found little application in underactuated systems. So far, most of the studies have reported its

application on fully actuated systems such as robotic manipulators (Yu et al., 2005), actuated exoskeleton (Madani et al., 2016), and power flow control (Shotorbani et al., 2014). Therefore, in this Chapter, FTSM based robust controllers are studied on the underactuated systems, along with the issues in design and implementation. Finally, a controller based on FTSM is also formulated for such systems.

FTSM can also be studied on systems requiring a faster response. One such system is a mirror-based pointing sensor, which consists of a mirror and other sensors such as cameras, laser sensors, and so on (Sueishi et al., 2015). In such sensors, the mirror undergoes rotation to provide the sensing capability. As a result, responses of such systems are faster compared to other sensors such as pan-tilt systems, and there is a huge demand for such sensors in the robotics community for various applications. For example, such devices are integrated with laser sensors and cameras for robot navigation (Wood and Bishop, 2012).

5.2 Underactuated Mechanical Systems and Modelling

Underactuated systems are the special types of system that have fewer number of the inputs than the degree of freedom (DOF) of the system. Mathematically, the underactuated system can be described by using Lagrangian formulation. Following equation represent system with two degree of freedom (2-DOF).

$$\begin{bmatrix} m_{11}(\mathbf{q}) & m_{12}(\mathbf{q}) \\ m_{21}(\mathbf{q}) & m_{22}(\mathbf{q}) \end{bmatrix} \begin{bmatrix} \ddot{q}_a \\ \ddot{q}_u \end{bmatrix} + \begin{bmatrix} C_1(\mathbf{q}, \dot{\mathbf{q}}) \\ C_2(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix} + \begin{bmatrix} g_1(\mathbf{q}) \\ g_2(\mathbf{q}) \end{bmatrix} = \begin{bmatrix} \tau \\ 0 \end{bmatrix} + F_s(\dot{\mathbf{q}}), \quad (5.1)$$

where $\mathbf{q} = [q_a \ q_u] \in \mathbb{R}^2$, $q_a \in \mathbb{R}$ and $q_u \in \mathbb{R}$ are, respectively, the actuated and underactuated generalized coordinates of the system,

$$\begin{bmatrix} m_{11}(\mathbf{q}) & m_{12}(\mathbf{q}) \\ m_{21}(\mathbf{q}) & m_{22}(\mathbf{q}) \end{bmatrix} = M(\mathbf{q}) \in \mathbb{R}^{2 \times 2}$$

is the inertial matrix such that $M(\mathbf{q}) = M(\mathbf{q})^\top$ and $M(\mathbf{q}) \geq 0$, $C_1(\mathbf{q}, \dot{\mathbf{q}})$ and $C_2(\mathbf{q}, \dot{\mathbf{q}})$ are the Coriolis terms, $g_1(\mathbf{q})$ and $g_2(\mathbf{q})$ are the gravitational terms, $\tau \in \mathbb{R}$ is the

generalized input force or torque, and $\mathbf{F}_s(\dot{\mathbf{q}})$ is due to friction, which can be modeled by the following nonlinear dynamics

$$\begin{aligned}\dot{\mathbf{z}} &= \dot{\mathbf{q}} - \Upsilon_0 \frac{\|\dot{\mathbf{q}}\|}{\varphi(\dot{\mathbf{q}})} \mathbf{z}, \\ \mathbf{F}_s(\dot{\mathbf{q}}) &= \Upsilon_0 \mathbf{z} + \Upsilon_1 \dot{\mathbf{z}} + \Psi(\dot{\mathbf{q}}),\end{aligned}\tag{5.2}$$

where $\mathbf{z} \in \mathbb{R}^2$ is the internal friction state, $\varphi(\dot{\mathbf{q}}) \in \mathbb{R}$ is a velocity dependent function, $\Psi(\dot{\mathbf{q}}) \in \mathbb{R}^2$ is a memoryless velocity-dependent function, $\Upsilon_0 \in \mathbb{R}^{2 \times 2}$ is a matrix of stiffness, and $\Upsilon_1 \in \mathbb{R}^{2 \times 2}$ is a memoryless velocity-dependent matrix. Similarly, $\varphi(\dot{\mathbf{q}})$ can be represented by the following equation:

$$\varphi(\dot{\mathbf{q}}) = F_c + (F_{\hat{s}} - F_c) \exp\left(\frac{\|\dot{\mathbf{q}}\|}{\dot{\mathbf{q}}_s}\right)^2,\tag{5.3}$$

where F_c is Coulomb friction, $F_{\hat{s}}$ is the stiction friction, and $\dot{\mathbf{q}}_s$ is the Stribeck velocity.

Now, (5.1) can also be formulated as

$$\begin{aligned}\ddot{q}_a &= f_1(\mathbf{q}, \dot{\mathbf{q}}) + b_1(\mathbf{q})u + \sigma_1(\mathbf{q}) \\ \ddot{q}_u &= f_2(\mathbf{q}, \dot{\mathbf{q}}) + b_2(\mathbf{q})u + \sigma_2(\mathbf{q}),\end{aligned}\tag{5.4}$$

where

$$f_1(\mathbf{q}, \dot{\mathbf{q}}) = -\{m'_{11}(\mathbf{q})(C_1(\mathbf{q}, \dot{\mathbf{q}}) + g_1(\mathbf{q})) + m'_{12}(\mathbf{q})(C_2(\mathbf{q}, \dot{\mathbf{q}}) + g_2(\mathbf{q}))\},$$

$$\begin{aligned}f_2(\mathbf{q}, \dot{\mathbf{q}}) &= -\{m'_{21}(\mathbf{q})(C_1(\mathbf{q}, \dot{\mathbf{q}}) + g_1(\mathbf{q})) + m'_{22}(\mathbf{q})(C_2(\mathbf{q}, \dot{\mathbf{q}}) + g_2(\mathbf{q}))\}, \\ &\quad \begin{bmatrix} m'_{11} & m'_{12} \\ m'_{21} & m'_{22} \end{bmatrix} = M^{-1}(\mathbf{q}),\end{aligned}$$

$b_1(\mathbf{q}) = m'_{11}$, $b_2(\mathbf{q}) = m'_{21}$, and $u = \tau$ is the control input. In order to simplify control system design process, the terms $\sigma_1(\mathbf{q})$ and $\sigma_2(\mathbf{q})$ in (5.4) are assumed to be uncertain, which represent the nonlinear friction model, i.e. (5.2), and the unmodelled dynamics of the system. Furthermore, those functions are assumed to be bounded, i.e. $|\sigma_1(\mathbf{q})| \leq \mu_1$, and $|\sigma_2(\mathbf{q})| \leq \mu_2$. The control system design process will be discussed in detail in the following section.

5.3 Hierarchical Sliding Mode Framework

Hierarchical sliding mode (HSM) (Wang et al., 2004) is a framework for a class of underactuated nonlinear systems (5.4). In this framework, hierarchy of sliding surfaces are defined. Specifically, sliding manifolds are divided into two layers. The bottom layer sliding manifolds consist of functions for each variable, i.e.

$$\begin{aligned} s_1 &= \dot{q}_a + \alpha_1 q_a \\ s_2 &= \dot{q}_u + \alpha_2 q_u, \end{aligned} \tag{5.5}$$

where $\alpha_1 > 0$ and $\alpha_2 > 0$. On the other hand, those surfaces are linearly combined in the higher layer, i.e.

$$S = k_1 s_1 + k_2 s_2 \tag{5.6}$$

Furthermore, HSM framework defines the following control input:

$$u = -\frac{1}{k_1 b_1(\mathbf{q}) + k_2 b_2(\mathbf{q})} \{u_1 + u_2 + u_3\}, \tag{5.7}$$

where

$$\begin{aligned} u_1 &= k_1 (f_1(\mathbf{q}, \dot{\mathbf{q}}) + \alpha_1 q_a) \\ u_2 &= k_2 (f_2(\mathbf{q}, \dot{\mathbf{q}}) + \alpha_2 q_u) \\ u_3 &= \eta \text{sgn}(S) + KS, \end{aligned}$$

and where the control gains K and η are chosen such that $K > 0$ and $\eta > D_m = \sup_{t \geq 0} |k_1 \sigma_1(\mathbf{q}) + k_2 \sigma_2(\mathbf{q})|$. The given control law can be formulated in the following theorem.

5.3.1 Stability Analysis

In order to guarantee the stability of the higher-level sliding surface S , one can consider the following lyapunov function:

$$V = \frac{1}{2} S^2, \tag{5.8}$$

Taking time derivative one can get

$$\dot{V} = S \dot{S}. \quad (5.9)$$

Let us consider the time derivative of the sliding function (5.13):

$$\begin{aligned} \dot{S} = & k_1 (\ddot{e}_1 + \alpha_1 \dot{e}_1) \\ & + k_2 (\ddot{e}_2 + \alpha_2 \dot{e}_2) \end{aligned}$$

or,

$$\begin{aligned} \dot{S} = & k_1 (f_1(\mathbf{q}, \dot{\mathbf{q}}) + \alpha_1 \dot{e}_1 - \ddot{q}_{ad} + \sigma_1(\mathbf{q})) \\ & + k_2 (f_2(\mathbf{q}, \dot{\mathbf{q}}) + \alpha_2 \dot{e}_2 - \ddot{q}_{ud} + \sigma_2(\mathbf{q})) \\ & + (k_1 b_1(\mathbf{q}) + k_2 b_2(\mathbf{q})) u. \end{aligned} \quad (5.10)$$

Now by substituting the control law (5.7) into the expression of \dot{S} and eventually into equation (5.9), and after some arrangements we find that

$$\begin{aligned} \dot{V} &= S \dot{S} = S \{k_1 \sigma_1(\mathbf{q}) + k_2 \sigma_2(\mathbf{q}) - \eta \text{sgn}(S) - K S\} \\ &= -K S^2 - \eta |S| + (k_1 \sigma_1(\mathbf{q}) + k_2 \sigma_2(\mathbf{q})) S. \end{aligned} \quad (5.11)$$

Therefore, we have

$$\dot{V} = S \dot{S} \leq -K S^2 - \eta |S| + |k_1 \sigma_1(\mathbf{q}) + k_2 \sigma_2(\mathbf{q})| |S|.$$

Now, let

$$D_m = \sup_{t \geq 0} |k_1 \sigma_1(\mathbf{q}) + k_2 \sigma_2(\mathbf{q})|.$$

Then, if η is chosen such that $\eta > D_m$ one has

$$\dot{V} = -K S^2 - (\eta - D_m) |S| < 0.$$

Thus, by the Lyapunov stability theory, the sliding function S is stable and the sliding mode, i.e. $S = 0$, can be achieved in the second layer of the underactuated system (5.4) in finite time.

Eventhough $S = 0$ is stable by Lyapunov stability theory, it does not guarantee the stability of the lower-level sliding surfaces s_1 and s_2 . Nevertheless, (Wang et al., 2004) in their research article have demonstrated the asymptotic stability of the sliding surfaces, which can be formulated into the following theorem

Theorem 2: (Wang et al., 2004) Consider a class of underactuated system with a stable equilibrium point, design the sliding surfaces as (5.5), (5.6), and use the control law (5.7). If $s_1 \in L_1$ and $s_2 \in L_2$ (i.e. s_1 and s_2 are bounded) and $\eta > D_m$; then the first-level sliding surfaces s_1 and s_2 are also asymptotically stable.

5.4 HSM and FTSM based control system design for underactuated systems

This section presents a methodology to synthesize FTSM based control system for underactuated systems. To do that, error variables for both actuated and unactuated variables are defined, respectively, as $e_1 = q_a - q_{ad}$ and $e_2 = q_u - q_{ud}$, where q_{ad} and q_{ud} are the desired reference for actuated and unactuated variables. As in HSM, 1st layer sliding surfaces are defined as

$$\begin{aligned} s_1 &= \dot{e}_1 + \beta_1 e_1^{q_1/p_1} + \alpha_1 e_1 \\ s_2 &= \dot{e}_2 + \beta_2 e_2^{q_2/p_2} + \alpha_2 e_2, \end{aligned} \quad (5.12)$$

where the control parameters p_1 , q_1 , p_2 , and q_2 are odd integers such that $\frac{q_1}{p_1} < 1$, $\frac{q_2}{p_2} < 1$, $\beta_1 > 0$, $\beta_2 > 0$, $\alpha_1 > 0$ and $\alpha_2 > 0$. Here sliding functions s_1 and s_2 consists of FTSM manifold. Similarly, the second layer sliding function is defined as

$$S = k_1 s_1 + k_2 s_2 \quad (5.13)$$

where, $k_1 > 0$ and $k_2 > 0$.

The control law for the class of 2-DOF underactuated systems (5.4) of this chapter is then proposed as:

$$u = -\frac{1}{k_1 b_1(\mathbf{q}) + k_2 b_2(\mathbf{q})} \{u_1 + u_2 + u_3\}, \quad (5.14)$$

where

$$\begin{aligned} u_1 &= k_1 \left(f_1(\mathbf{q}, \dot{\mathbf{q}}) + \beta_1 \frac{q_1}{p_1} e_1^{q_1/p_1-1} \dot{e}_1 + \alpha_1 e_1 - \ddot{q}_{ad} \right) \\ u_2 &= k_2 \left(f_2(\mathbf{q}, \dot{\mathbf{q}}) + \beta_2 \frac{q_2}{p_2} e_2^{q_2/p_2-1} \dot{e}_2 + \alpha_2 e_2 - \ddot{q}_{ud} \right) \\ u_3 &= \eta \text{sgn}(S) + KS, \end{aligned}$$

and where the control gains K and η are chosen such that $K > 0$ and $\eta > D_m = \sup_{t \geq 0} |k_1 \sigma_1(\mathbf{q}) + k_2 \sigma_2(\mathbf{q})|$. In order to avoid singularity, the following condition must be valid

$$k_1 b_1(\mathbf{q}) + k_2 b_2(\mathbf{q}) \neq 0 \quad \forall \mathbf{q} \in \mathbb{R}^2.$$

Further analysis on this singularity condition for cranes will be discussed in the Chapter 6.

5.4.1 Stability analysis

For the stability of the proposed control input let us define a Lyapunov candidate function for the sliding surface S as

$$V = \frac{1}{2} S^2, \quad (5.15)$$

Taking time derivative one can get

$$\dot{V} = S \dot{S} \quad (5.16)$$

Let us consider the time derivative of the sliding function (5.13):

$$\begin{aligned} \dot{S} &= k_1 \left(\ddot{e}_1 + \beta_1 \frac{q_1}{p_1} e_1^{q_1/p_1-1} \dot{e}_1 + \alpha_1 \dot{e}_1 \right) \\ &\quad + k_2 \left(\ddot{e}_2 + \beta_2 \frac{q_2}{p_2} e_2^{q_2/p_2-1} \dot{e}_2 + \alpha_2 \dot{e}_2 \right) \end{aligned}$$

or,

$$\begin{aligned} \dot{S} &= k_1 \left(f_1(\mathbf{q}, \dot{\mathbf{q}}) + \beta_1 \frac{q_1}{p_1} e_1^{q_1/p_1-1} \dot{e}_1 + \alpha_1 \dot{e}_1 - \ddot{q}_{ad} + \sigma_1(\mathbf{q}) \right) \\ &\quad + k_2 \left(f_2(\mathbf{q}, \dot{\mathbf{q}}) + \beta_2 \frac{q_2}{p_2} e_2^{q_2/p_2-1} \dot{e}_2 + \alpha_2 \dot{e}_2 - \ddot{q}_{ud} + \sigma_2(\mathbf{q}) \right) \\ &\quad + (k_1 b_1(\mathbf{q}) + k_2 b_2(\mathbf{q})) u. \end{aligned} \quad (5.17)$$

Now by substituting the control law (5.14) into the expression of \dot{S} and eventually into equation (5.16), and after some arrangements we find that

$$\begin{aligned}\dot{V} &= S\dot{S} = S \{k_1\sigma_1(\mathbf{q}) + k_2\sigma_2(\mathbf{q}) - \eta \operatorname{sgn}(S) - KS\} \\ &= -KS^2 - \eta |S| + (k_1\sigma_1(\mathbf{q}) + k_2\sigma_2(\mathbf{q}))S.\end{aligned}\quad (5.18)$$

Therefore, we have

$$\dot{V} = S\dot{S} \leq -KS^2 - \eta |S| + |k_1\sigma_1(\mathbf{q}) + k_2\sigma_2(\mathbf{q})| |S|. \quad (5.19)$$

Now, let

$$D_m = \sup_{t \geq 0} |k_1\sigma_1(\mathbf{q}) + k_2\sigma_2(\mathbf{q})|.$$

Then, if η is chosen such that $\eta > D_m$ one has

$$\dot{V} < -KS^2 - (\eta - D_m) |S| < 0. \quad (5.20)$$

Thus, by the Lyapunov stability theory, the sliding function S is stable and the sliding mode, i.e. $S = 0$, can be achieved in the second layer of the underactuated system (5.4) in finite time. By integrating both sides of (5.18), we can obtain

$$\begin{aligned}V &= \frac{1}{2}S^2 \\ &= V(0) + \int_0^t [-KS^2 - \eta |S| + (k_1\sigma_1(\mathbf{q}) + k_2\sigma_2(\mathbf{q}))S] d\tau \\ &\leq V(0) - \int_0^\infty (KS^2 + (\eta - D_m) |S|) d\tau < V(0) < \infty.\end{aligned}$$

or,

$$\lim_{t \rightarrow \infty} \int_{t=0}^\infty (KS^2 + (\eta - D_m) |S|) d\tau < \infty. \quad (5.21)$$

Now, from the Barbarat lemma

$$\lim_{t \rightarrow \infty} (KS^2 + (\eta - D_m) |S|) = 0. \quad (5.22)$$

In other words,

$$\lim_{t \rightarrow \infty} S = 0. \quad (5.23)$$

Hence, the higher layer sliding surface S is asymptotically stable and bounded for the control input (5.14).

5.4.2 Stability for crane dynamics

Let us consider a crane and its dynamics. An underactuated overhead crane is shown in Fig. 7.3(a). It consists of a cart of mass m_c , a hoisting mechanism through a rope of length l , and a payload of mass m_p . The position of the cart (x) and the swing angle (θ) of the payload represent the generalized coordinates q_a and q_u , respectively. Here, it is assumed that the payload consists of a point mass and the stiffness of the hoisting rope is negligible. As shown in Fig. 7.3(a), the cart is driven by a DC motor via a rack and pinion mechanism. It should be noted that all nonlinearities introduced due to the drive system here is treated as uncertainties as described in Equation (5.4). The dynamic equations for the crane system of the form (5.4) are obtained as:

$$\begin{aligned} f_1(\mathbf{q}, \dot{\mathbf{q}}) &= \frac{-m_p l^2 \left(\frac{k_T k_b}{R_a r^2} \right) \dot{x} + m_p^2 l^2 \sin \theta (l \dot{\theta} + g \cos \theta)}{(m_p + m_c)m_p l^2 - m_p^2 l^2 \cos^2 \theta}, \\ f_2(\mathbf{q}, \dot{\mathbf{q}}) &= \frac{m_p \left(\frac{k_T k_b}{R_a r} \right) l \dot{\theta} \cos \theta}{(m_p + m_c)m_p l^2 - m_p^2 l^2 \cos^2 \theta} \\ &\quad - \frac{m_p l \sin \theta \left\{ (m_p + m_c)g + m_p l \cos \theta \dot{\theta}^2 \right\}}{(m_p + m_c)m_p l^2 - m_p^2 l^2 \cos^2 \theta}, \\ b_1(\mathbf{q}) &= \frac{m_p l^2 \frac{k_T}{R_a r}}{(m_p + m_c)m_p l^2 - m_p^2 l^2 \cos^2 \theta}, \\ b_2(\mathbf{q}) &= \frac{-m_p l \frac{k_T}{R_a r} \cos \theta}{(m_p + m_c)m_p l^2 - m_p^2 l^2 \cos^2 \theta}. \end{aligned} \quad (5.24)$$

Remark 2: For generic underactuated systems, as noted by Park (Park and Lee, 1996) and Ma (Ma, 2007), although S is asymptotically stable under the HSM framework, it may not guarantee the simultaneous stability of s_1 and s_2 . Nevertheless, for

the crane system (7.1), suppose that FTSM-based functions s_1 and s_2 are bounded for the crane dynamics, one of the sliding surfaces is stable under the HSM based control input, say $s_1 \rightarrow 0$. Then, from the definition of the higher-order sliding function, $S = k_1 s_1 + k_2 s_2$; if s_2 is approaching a nonzero value then S will not approach 0, which contradicts the fact that $S \rightarrow 0$ under the control input (5.14). Therefore, for the given crane dynamics with the proposed control law, both s_1 and s_2 are asymptotically stable.

5.5 Mirror-based Pointing Technology and modelling

5.5.1 Pointing System Construction and System Variables

A mirror-based pointing sensor, as shown in Fig. 5.1, has a light-weight mirror which is directly placed above the sensor. The sensor can be of any type such as thermal camera, vision cameras, etc. The advantage of this type of sensor is that the mirror, which has low momentum, undergoes rotational motion to provide sensing capability. Therefore, the dynamic response of the sensor is tremendously improved, which can be of great advantage in high-speed tracking applications, such as the one mentioned previously. In this chapter, the pointing sensor developed and commercialised by Ocular Robotics Pty. Ltd. is considered, which is also known as RobotEye.

This type of sensor can be represented by two variables namely azimuth and the elevation angles, as shown in Fig. 5.1 (b). The azimuth angle represents the rotation of the sensor head about the vertical axis as depicted in the figure. Similarly, the elevation angle represents the angle made by the viewing direction with respect to the horizontal plane.

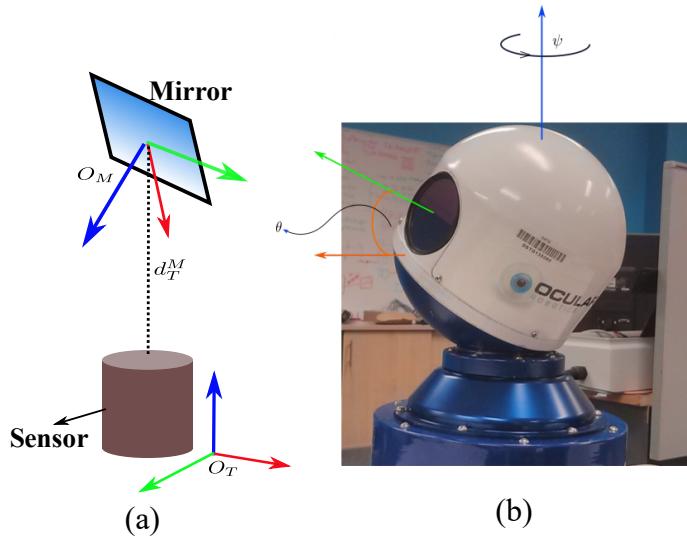


Figure 5.1 : Mirror-based pointing sensor from Ocular Robotics Pty. Ltd.

5.5.2 System Modelling

The motion dynamics of the system can be represented by the following MIMO transfer function, i.e.

$$\begin{bmatrix} \theta(s) \\ \psi(s) \end{bmatrix} = \begin{bmatrix} H_{11}(s) & H_{12}(s) \\ H_{21}(s) & H_{22}(s) \end{bmatrix} \begin{bmatrix} \theta_u(s) \\ \psi_u(s) \end{bmatrix}, \quad (5.25)$$

where s represents the laplace variable, $\theta_u(s)$ and $\psi_u(s)$ represent the inputs to the system. Similarly, $\theta(s)$ and $\psi(s)$ are the outputs of the system. Furthermore,

$$\begin{aligned} H_{11}(s) &= \left. \frac{\theta(s)}{\theta_u(s)} \right|_{\psi_u=0}, & H_{21}(s) &= \left. \frac{\psi(s)}{\theta_u(s)} \right|_{\psi_u=0}, \\ H_{12}(s) &= \left. \frac{\theta(s)}{\psi_u(s)} \right|_{\theta_u=0}, & H_{22}(s) &= \left. \frac{\psi(s)}{\psi_u(s)} \right|_{\theta_u=0}. \end{aligned}$$

Here, it should be noted that system (5.25) represents the coupling of the azimuth and elevation dynamics via H_{12} and H_{21} transfer functions. Therefore, it can be referred as coupled model, which can be represented in the following state-space form:

$$\begin{aligned} \dot{\mathbf{x}} &= A\mathbf{x} + B\mathbf{u} \\ \mathbf{y} &= C\mathbf{x}, \end{aligned} \quad (5.26)$$

where $\mathbf{x} \in \mathbb{R}^4$, $\mathbf{y} \in \mathbb{R}^2$, $\mathbf{u} \in \mathbb{R}^2$. Similarly, $A \in \mathbb{R}^{4 \times 4}$ can be referred as transition matrix, $B \in \mathbb{R}^{4 \times 2}$ as the input matrix and $C \in \mathbb{R}^{2 \times 4}$ is output matrix.

Decoupling of the Elevation and Azimuth space

Now, from (5.25), one can represent elevation angle as:

$$\theta(s) = H_{11}(s)\theta_u(s) + H_{12}(s)\psi_u(s),$$

or,

$$\theta(s) = H_{11}(s)\theta_u(s) + D_\psi(s), \quad (5.27)$$

where $D_\psi(s) = H_{12}(s)\psi_u(s)$ represents the coupling due to ψ . Here, in order to decouple the elevation dynamics from the azimuth, one can consider $D_\psi(s)$ as the disturbance. Similarly, azimuth angle ($\psi(s)$) can also be represented as

$$\psi(s) = H_{22}(s)\psi_u(s) + D_\theta(s), \quad (5.28)$$

where $D_\theta(s)$ represents the disturbance arising from the coupling. Furthermore, to simplify the control system design process, one can consider $D_\psi(s)$ and $D_\theta(s)$ to be bounded, i.e. $|D_\psi(s)| < \mu_\psi$ and $|D_\theta(s)| < \mu_\theta$.

The transfer functions $H_{11}(s)$ and $H_{22}(s)$ can be represented by a second-order transfer function as:

$$H(s) = \frac{b_0}{s^2 + a_1 s + a_2}. \quad (5.29)$$

Now, one can also represent the transfer function in time-domain via the following state-space equation:

$$\begin{aligned} \dot{x}_1 &= x_2(t) \\ \dot{x}_2 &= -a_1 x_1(t) - a_2 x_2(t) + b_0 u(t) + d(t) \end{aligned} \quad (5.30)$$

where $x_1 \in \mathbb{R}$ and $x_2 \in \mathbb{R}$ are the states of the system, and $u \in \mathbb{R}$ is the input to the system. The lumped term $d(t) \in \mathbb{R}$ in the equation represents the disturbance due to the coupling, i.e $d_\theta(t)$ and $d_\psi(t)$, and also due to other factors, such as unmodelled terms in the dynamics. Furthermore, $d(t)$ is assumed to be bounded, i.e. $|d(t)| < \mu$.

5.5.3 Identification of System Model

The parameters b_0 , a_1 , and a_2 of $H_{11}(s)$ and $H_{22}(s)$ should be estimated by applying identification algorithms. For that, input-output datas were collected from the pointing sensor, as mentioned previously. Then, nonlinear least-square methods were applied, which are provided as an application program interface (API) in Matlab (see (Garnier et al., 2003) for the details on the methods). The identified parameters are listed in Table 5.1

Parameters	H_{11}	H_{22}
b_0	3581	3317
a_1	59.6	58.6
a_2	3568	3310

Table 5.1 : Identified parameters of the coupled-model of the RobotEye.

Similarly, for the identification of the system's parameter of (5.26), the angles and inputs were measured through the application programming interface (API) provided by the robot. Then, a standard identification algorithm such as N4SID was applied in the framework provided by Matlab. The identified parameters are listed below:

$$A = \begin{bmatrix} -0.4141 & 255.6 & -848.2 & 54.97 \\ -75.28 & -40.89 & -495.1 & -127 \\ 757.5 & 43.73 & -546.4 & -80.56 \\ 75.16 & 48.69 & -210.6 & -13.13 \end{bmatrix},$$

$$B = \begin{bmatrix} -96.69 & -82.96 \\ -42.41 & -45.76 \\ -108.7 & -92.47 \\ -28.4 & -24.58 \end{bmatrix},$$

and

$$C = \begin{bmatrix} -9.633 & -9.392 & -8.936 & 72.69 \\ 19.64 & -11.09 & 7.066 & -98.29 \end{bmatrix}.$$

Verification of the identified models are presented in the Fig. 5.2. The figure shows the responses of identified H_{11} and H_{22} along with the state-space model (5.26). From the figure, it is clear that the outputs of the models are close to the actual response. The accuracy of the model response is around 93 and 94 %, respectively, for H_{11} and H_{22} . Similarly, the accuracy is around 94% for the coupled model. The accuracies are measured in terms of normalized root mean square error (NRMSE). The performance index is defined as

$$\text{NRMSE} = \left(1 - \frac{\sqrt{\frac{\sum_{i=1}^{N_s} (y_i - \hat{y}_i)^2}{N_s}}}{\sqrt{\sum_{i=1}^{N_s} (y_i - \bar{y})^2}} \right), \quad (5.31)$$

where y_i ($i=1 \dots N_s$) are the actual outputs, \hat{y}_i ($i=1 \dots N_s$) are the predicted outputs of the estimated models, \bar{y} is the mean of the actual outputs, i.e. y_i ($i=1 \dots N_s$), and N_s is the number of samples.

5.6 FTSM Control for the Optical Pointing Sensor

5.6.1 FTSM based control formulation

In order to track a reference signal $r(t)$ for system (5.30), one can design the following FTSM based sliding functions:

$$\begin{aligned} \sigma_1 &= x_1 - r(t) \\ \sigma_2 &= \dot{\sigma}_1 + \alpha\sigma_1 + \beta\sigma_1^{q_1/p_1}, \end{aligned} \quad (5.32)$$

where $q_1 > 0$ and $p_1 > 0$ are odd integers such that $0 < \frac{q_1}{p_1} < 1$. Similarly $\alpha > 0$ and $\beta > 0$. Now, the control law for the system (5.30) using FTSM manifold is given by:

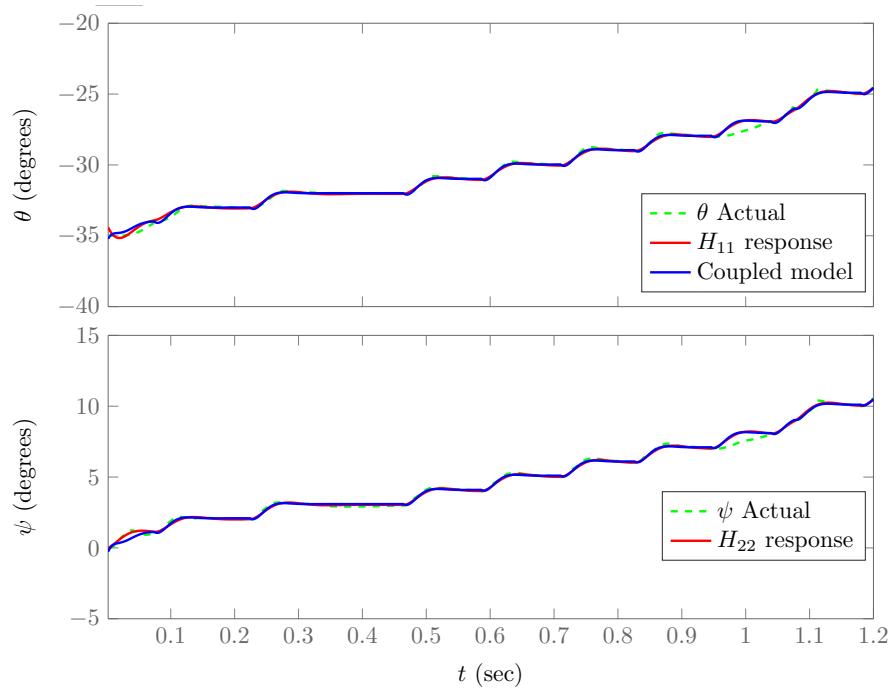


Figure 5.2 : Responses of the identified models, i.e. H_{11} and H_{22} .

$$\begin{aligned} u &= -\frac{u_0}{b_0} \\ u_0 &= -a_1 x_1 - a_2 x_2 - \ddot{r} + \alpha \dot{\sigma}_1 + \beta \frac{q_1}{p_1} \sigma^{\frac{q_1-p_1}{p_1}} \dot{\sigma}_1 + \Phi \sigma_2 + K \text{sign}(\sigma_2), \end{aligned} \quad (5.33)$$

where $K > \mu$ and $\Phi > 0$.

5.6.2 Discrete-time FTSM control input synthesis

There are many research papers on continuous-time FTSM, but very few studies deal with the synthesis and analysis of discrete-time FTSM. In this regard, Shihua Li et al., in (Li et al., 2014), provided a comprehensive study on the control system design methodology based on Terminal Sliding Mode (TSM), followed by its analysis in steady-state condition by Behera et al. in (Behera and Bandyopadhyay, 2015). Here also their methodology to design discrete-time FTSM controller is followed by applying Euler's discretization to (5.30), as in (Galias and Yu, 2007), which leads

to:

$$\begin{aligned} x_1[n+1] &= x_1[n] + Tx_2[n] \\ x_2[n+1] &= a_1Tx_1[n] + (1 + a_2T)x_2[n] \\ &\quad + b_0Tu[n] + Td[n], \end{aligned} \tag{5.34}$$

where T is the sampling period. Furthermore, the discretization of the FTSM sliding functions (5.32) results in:

$$\begin{aligned} \sigma_1[n] &= x_1[n] - r[n] \\ \sigma_2[n] &= \Delta\sigma_1[n] + \alpha\sigma_1[n] + \beta\sigma_1^{q_1/p_1}[n], \end{aligned} \tag{5.35}$$

where $\alpha > 0$, $\beta > 0$, and q_1 and p_1 are odd integers such that $q_1 < p_1$. Here, Δ is discrete-time approximation of differential operator (Li et al., 2014), which is also known as forward differential operator:

$$\Delta(x[n]) = \frac{x[n+1] - x[n]}{T}. \tag{5.36}$$

By applying differential operator, σ_1 can also be represented as:

$$\Delta\sigma_1[n] = x_2[n] - \Delta(r[n]). \tag{5.37}$$

Similarly, $\Delta(\sigma_1^{q_1/p_1})$ is given by:

$$\Delta(\sigma_1^{q_1/p_1}) = \frac{q_1}{p_1}\sigma_1^{q_1/p_1-1}\Delta(\sigma_1). \tag{5.38}$$

Now, following the control system design methodology by Shihua et al. (Li et al., 2014), one obtains the following control law:

$$\begin{aligned} u &= -\frac{u_0}{b_0} \\ u_0 &= -a_1x_1[n] - a_2x_2[n] - \Delta^2(r[n]) \\ &\quad + (\alpha + \beta\frac{q_1}{p_1}\sigma_1^{q_1/p_1-1})\Delta(\sigma_1^{q_1/p_1}) + K\text{sign}(\sigma_2), \end{aligned} \tag{5.39}$$

where $K > 0$ and $\frac{1}{T} > \Phi > 0$. Here, it should be noted that the discrete-time sliding function σ_2 dynamics does not go to 0 as in continuous-time, but remains within a bounded region, which is also known as quasi sliding mode band (Gao et al., 1995). The bounded region depends on K (Behera and Bandyopadhyay, 2015).

Furthermore, K affects the transient behaviour of the sliding function σ_2 . To understand the phenomena, consider the dynamics of σ_2 resulting from the application of (5.39), as in (Li et al., 2014), i.e.

$$\sigma_2[n+1] - \sigma_2[n] = -KT \operatorname{sign}(\sigma_2). \quad (5.40)$$

Now, consider σ_2 outside the boundary region or when the system is in transient, i.e. $|\sigma_2| > 2KT$. In this region, let us consider two cases, i.e. $\sigma_2 > 0$ and $\sigma_2 < 0$. Expression of σ_2 for both cases are given by

$$\begin{cases} \sigma_2[n+1] = \sigma_2[n] - KT, & \sigma_2 > 0 \\ \sigma_2[n+1] = \sigma_2[n] + KT, & \sigma_2 < 0, \end{cases} \quad (5.41)$$

and the plot of σ_2 in these regions is shown in Fig. 5.3. It is obvious from the equations that K affects the slope of σ_2 trajectory. As a result, higher values of K result in faster response of the system. However, it also means the region of oscillation in the steady state condition is also higher. Therefore, there exists a tradeoff while selecting values for K , and hence it limits the transient response of σ_2 .

In order to improve the transient behaviour of the sliding function σ_2 , following control input is proposed:

$$\begin{aligned}
u &= -\frac{u_0}{b_0} \\
u_0 &= -a_1 x_1[n] - a_2 x_2[n] - \Delta^2(r[n]) \\
&\quad + (\alpha + \beta \frac{q_1}{p_1} \sigma_1^{q_1/p_1-1}) \Delta(\sigma_1^{q_1/p_1}) \\
&\quad + \Phi \sigma_2 + K \text{sign}(\sigma_2).
\end{aligned} \tag{5.42}$$

Here $\Phi \sigma_2$ is added in comparison to (5.39), which lead to the following sliding dynamics:

$$\sigma_2[n+1] = (1 - \Phi T) \sigma_2[n] - K T \text{sign}(\sigma_2[n]). \tag{5.43}$$

The resulting transient dynamics of σ_2 is depicted in Fig. 5.3 by the pink color curve. The result clearly demonstrates the improvement in the transient behaviour of σ_2 .

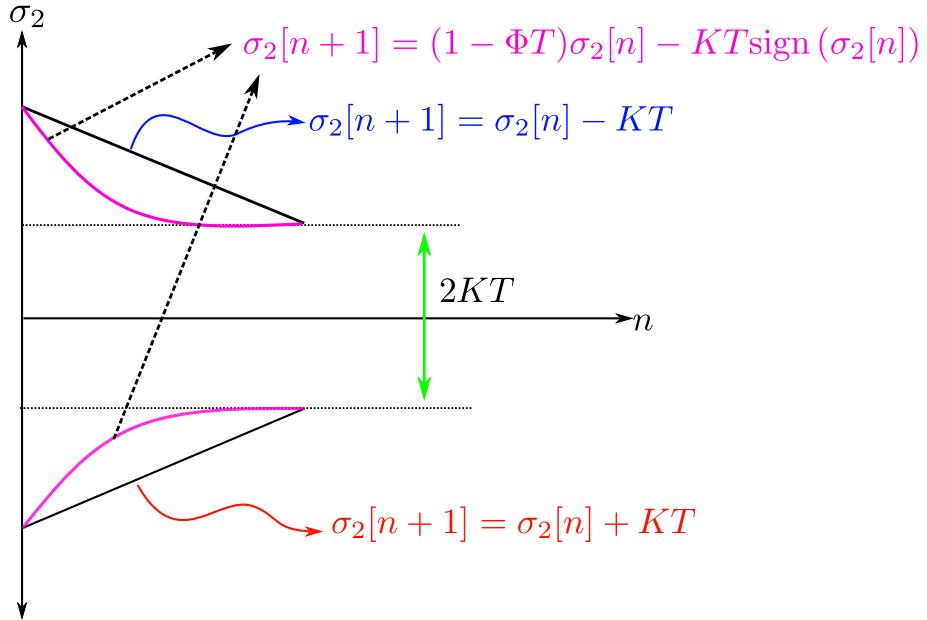


Figure 5.3 : Transient analysis of discrete-time sliding function $\sigma_2[n]$.

Reaching condition for sliding surfaces

Owing to the sampling process, the reaching condition for discrete-time system cannot be directly mapped from its continuous-time counterpart by using the concept of equivalent control (Shtessel et al., 2014). Many studies have been conducted in the past to establish the reaching condition. One such condition was proposed by Sarpturk et al. (Sarpturk et al., 1987), which is given by

$$|\sigma[k+1]| < |\sigma[k]|. \quad (5.44)$$

The condition can also be represented as

$$\begin{aligned} [\sigma[k+1] - \sigma[k]] \operatorname{sign}(\sigma[k]) &< 0 \\ [\sigma[k+1] + \sigma[k]] \operatorname{sign}(\sigma[k]) &> 0, \end{aligned}$$

which indicates that a system satisfying the condition (5.44) converges towards the sliding surface, and then it remains within a bounded region after that. The bounded region is also known as quasi sliding mode band (Zheng et al., 2006).

5.6.3 Stability Analysis

Equation (5.35) can be represented as:

$$\sigma_2[n] = x_2[n] - \Delta(r[n]) + \alpha\sigma_1[n] + \beta\sigma_1^{q_1/p_1}[n]. \quad (5.45)$$

Similarly, one can get

$$\begin{aligned} \sigma_2[n+1] &= x_2[n+1] - \Delta(r[n+1]) + \alpha\sigma_1[n+1] \\ &\quad + \beta\sigma_1^{q_1/p_1}[n+1]. \end{aligned} \quad (5.46)$$

Now, by applying the forward differential operator to $\sigma_2[n]$, and then to (5.38) and (5.37), one obtains

$$\begin{aligned} \frac{\sigma_2[n+1] - \sigma_2[n]}{T} &= x_2[n+1] - x_2[n] - \Delta^2(r[n]) \\ &\quad + \left(\alpha + \beta \frac{q_1}{p_1} \sigma_1^{q_1/p_1-1} \right) \Delta(\sigma_1). \end{aligned} \quad (5.47)$$

By applying (5.34) and the control law (5.42) to (5.47), one obtains

$$\sigma_2[n+1] - \sigma_2[n] = -\Phi T \sigma_2 - K T \text{sign}(\sigma_2),$$

or,

$$\sigma_2[n+1] = (1 - \Phi T) \sigma_2[n] - K T \text{sign}(\sigma_2[n]).$$

Now, by taking the absolute values on both sides of the equation leads to

$$|\sigma_2[n+1]| = |(1 - \Phi T) \sigma_2[n] - K T \text{sign}(\sigma_2[n])|,$$

or,

$$|\sigma_2[n+1]| = |(1 - \Phi T) \sigma_2[n]| - |K T \text{sign}(\sigma_2[n])|.$$

Since $0 < (1 - \Phi T) < 1$, we have $|(1 - \Phi T) \sigma_2[n]| - |K T \text{sign}(\sigma_2[n])| < |\sigma_2[n]|$. As a result,

$$|\sigma_2[n+1]| < |\sigma_2[n]|.$$

Hence, the system satisfies the Sarptuk condition (5.44). Therefore, the system converges towards sliding surfaces and remains bounded. Similarly, once $\sigma_2[n]$ is bounded, by using (5.35) it can be concluded that $\sigma_1[n]$ or the error is also bounded (Li et al., 2014).

5.7 Summary

In this chapter, design and implementation of a FTSM controller for a class of 2-DOF underactuated systems has been presented. The proposed method is based on a hierarchy of the first layer of FTSM sliding functions for each system variable and the second layer with a linear combination of the FTSM sliding functions. The advantage of the proposed control technique is faster convergence to equilibrium with respect to the terminal sliding mode (TSM) and conventional sliding mode control (SMC). Furthermore, an effective discrete-time FTSM controller is also presented for a mirror-based pointing sensor. The controller has been applied on the decoupled

state space models of the system. Stability of the discrete-time system has been verified in terms of a reaching condition. In addition, this chapter contributes to the improvement in the sliding dynamics of the discrete-time FTSM controller.

Part III

Applications

Chapter 6

Application of Mirror-based Technology in Tracking Thermal Targets

6.1 Introduction

Mechanical stress on a structure is related to the change in the temperature of the structure by the following expression which was formulated by William Thompson in 17th century (Rajic and Street, 2014):

$$\Delta T = -\frac{\alpha_T}{\rho C_p} T \Delta \sigma, \quad (6.1)$$

where ΔT is the change in the temperature in the structure, $\Delta \sigma$ is the stress experienced by the structure, α_T is the coefficient of thermal expansion, ρ is the density, and C_p is the specific heat at constant pressure. This, along with the advancement in the thermal imaging sensors, led to development of a non-destructive approach to analyse stress in mechanical structures. This method is also known as thermoelastic stress analysis (TSA) (Fig. 6.1).

TSA is effectively used in the full-field stress measurement of mechanical structures for their condition assessment. For instance, a system for thermographic analysis of the cracks in steel bridges has been reported in (Sakagami et al., 2010) for their structural integrity evaluation. Similarly, TSA is widely used in the aerospace industry to monitor the structural health of aircraft. In (Rajic et al., 2018) and (Wong et al., 2014), stress measurement of F/A-18 Hornet and P-3C Orion aircraft has been reported (see Fig. 6.2 for detail).

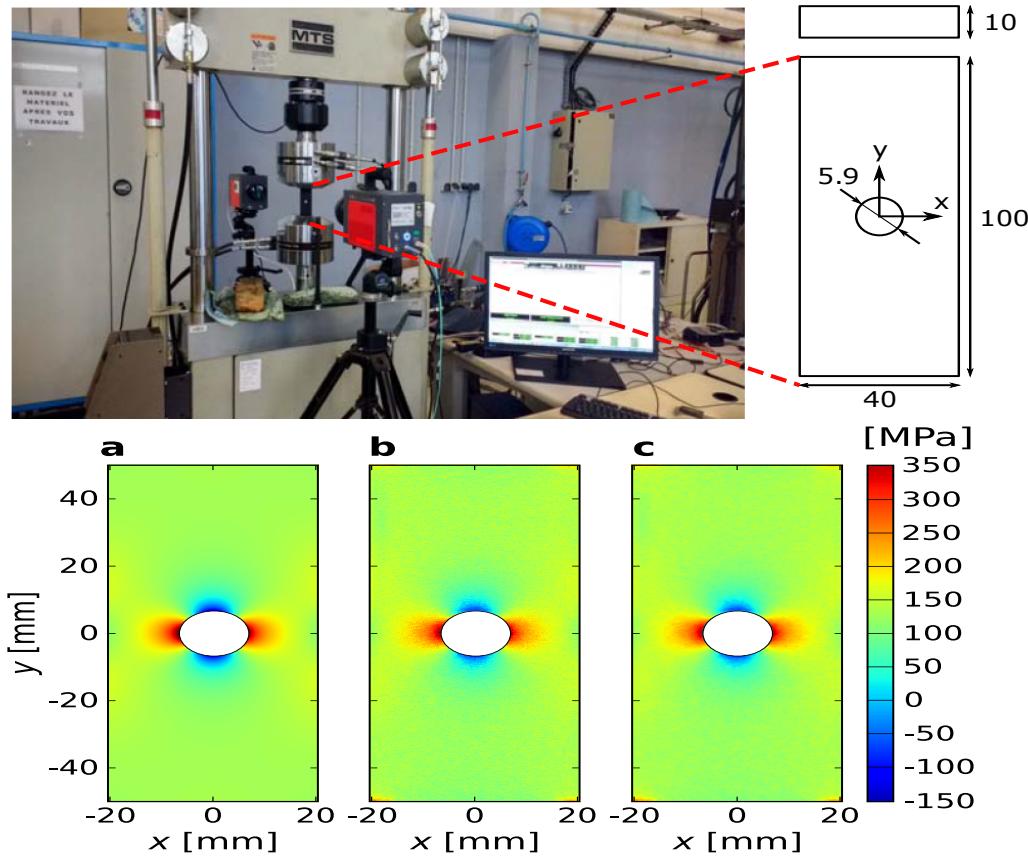


Figure 6.1 : Thermoelastic stress analysis (TSA) on a test coupon using a thermal camera (Source: Saux and Doudard (2017)).

For the TSA, structure-under-test is placed on the field-of-view of a thermal camera, a constant cyclic loading is applied to the structure. However, its motion affects the stress analysis due to the blur images from the camera. Many techniques can be applied to compensate the motion of the structure, such as by a robotic arm, but such methods have been reported ineffective in (Wong et al., 2014). To address the issue, researchers from DSTG proposed of using a mirror placed directly above the sensor to compensate for the motion of the target. This chapter presents the structure of the motion compensating system. In addition, this chapter also presents the application of FTSM controllers (Chapter 5) and LS-SVR based identification methods (Chapter 3) in the system.

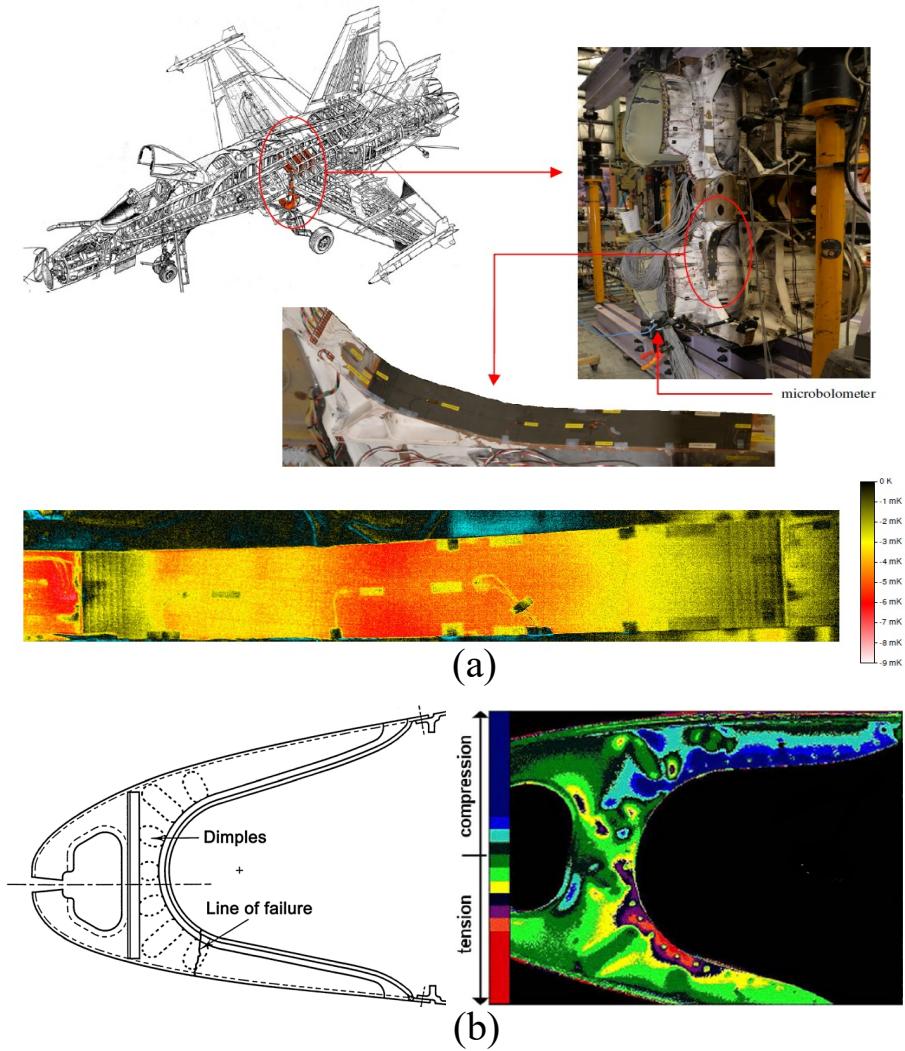


Figure 6.2 : Thermoelastic stress analysis (TSA) of aircraft components: a) TSA of center-barrel structure of a F/A-18 Hornet aircraft at Defence Science Technology Group (DSTG) (Source: Rajic et al. (2018)), and b) Stress analyses of a P-3C Orion wing leading edge rib (Source: Wong et al. (2014)).

6.2 System overview

The proposed system is presented in Fig. 6.3, which consists of a pointing system, vision camera, thermal camera, and a marker detection system. The thermal camera in this system is embedded inside the optical pointing device, so that it can

capture the view where the robot is pointing at. As a result, the responsiveness of the pointer is important for the performance of the tracking system. There are various options for the pointer, e.g. pan and tilt, robotic manipulator, and so on. Nevertheless, RobotEye technology from Ocular Robotics Pty. Ltd. is considered in this project because it has faster response in comparison to the existing state-of-the-art technology. Details on its construction are provided in Chapter 5, and its comparison with the existing such devices are provided in the studies by Wood and Bishop (2012).

In the proposed tracking system, the object-under-test is accompanied by a fiducial marker, which is placed in the field-of-view of the vision camera. Summaries of the specification of the camera along with the pointing system and thermal camera are presented in Table 6.1. Similarly, details on the marker construction and its detection algorithm are presented in Section 6.3.



Vision Camera



RobotEye



Structure under test



Thermal Camera

Figure 6.3 : Components of the prototype thermal target tracking system.

	Parameter	Values
Vision camera	Focal Length	16mm
	Resolution	658×492
	Frame rate	100fps
Thermal camera	Focal Length	19mm
	Resolution	640×512
	Frame rate	30fps
RobotEye	Model no	RELW50
	Aperture diameter	50mm

Table 6.1 : Specification of the vision camera, thermal camera, and the mirror-based pointing device for the tracking system.

6.2.1 System configuration

The important components in this system are the vision and the optical pointing system. The coordinate systems assigned to the sensor and the camera are shown in Fig. 6.4. In the figure, the reference frames for the vision and the sensor are represented by O_v and O_R , respectively. Similarly, R_V^R is the rotational matrix that represents the orientation of the robot with respect to the vision camera. Consider an observation of the target position with respect to the vision camera is represented as p_V^t . Then, the observation can be represented with respect to the robot reference frame as

$$p_R^t = R_V^R p_V^t + p_R^V \quad (6.2)$$

where p_R^t is the position of target and p_R^V is the location of vision camera with respect to the pointing device. The gazing direction for the optical pointing sensor is represented by azimuth ψ and elevation θ . Details on the configuration of the variables is presented in Section 5.5. Furthermore the angles are represented in

the spherical coordinate system. Therefore, for the representation of the thermal target as the gazing direction from the robot, conversion from Cartesian to spherical coordinates should be applied as

$$\begin{aligned}\psi &= \tan^{-1} \frac{y}{x} \\ \theta &= \tan^{-1} \left(\frac{\sqrt{y^2 + x^2}}{z} \right)\end{aligned}\quad (6.3)$$

where, $p_R^t = [x \ y \ z]$.

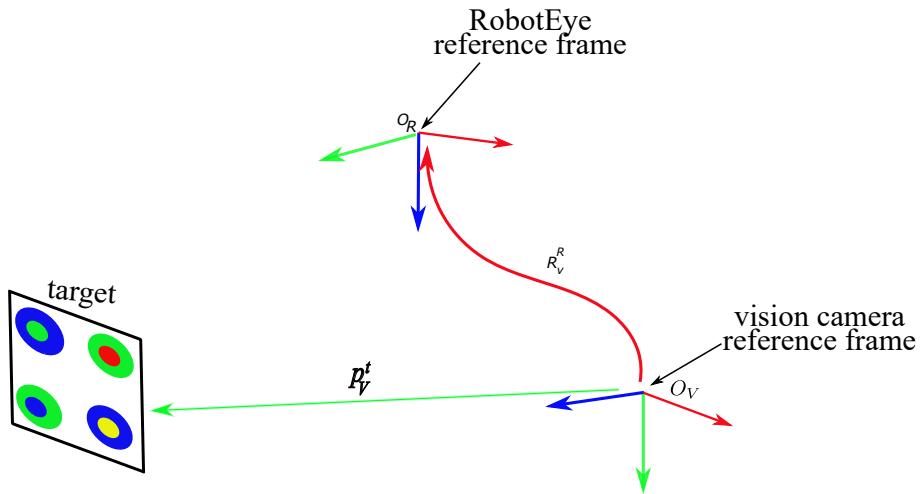


Figure 6.4 : Arrangement of the vision camera and pointing sensor for tracking target.

6.2.2 System architecture

The architecture for the proposed tracking system is depicted in Fig. 6.5. In the system, the position and orientation of the object-under-test is estimated from the frames captured by vision camera which utilizes the circular marker detection algorithm, details of which will be presented in the Section 6.3. After that, the position of the thermal target with respect to the pointing sensor is calculated. The cartesian coordinates of the target is converted to spherical using the expression (6.3), which

requires the extrinsic parameters such as R_V^T and p_R^V . After that, a prediction algorithm based on Kalman filter is utilised to predict its future trajectory. The filter requires the model of the target in state-space form (2.1). In order to estimate the model, the proposed LS-SVR based identification algorithm, as described in Chapter 3, is applied. Details on the experimental setup for the identification, along with the comparison with standard methods such as MOESP are presented in Section 6.4. Finally, future trajectories of the structure-under-test is fed to a lag-compensating controller which generates commands for the optical pointing device.

The pointing device is an important component in this system, therefore its control system plays a vital role in tracking. There are various challenges for the controller. Nevertheless, one of the challenges comes from the random delays in the system because all the components in the system are connected to a local area network (LAN), which can easily introduce lags. As a result, the controller should be able to cope such delays in the system.

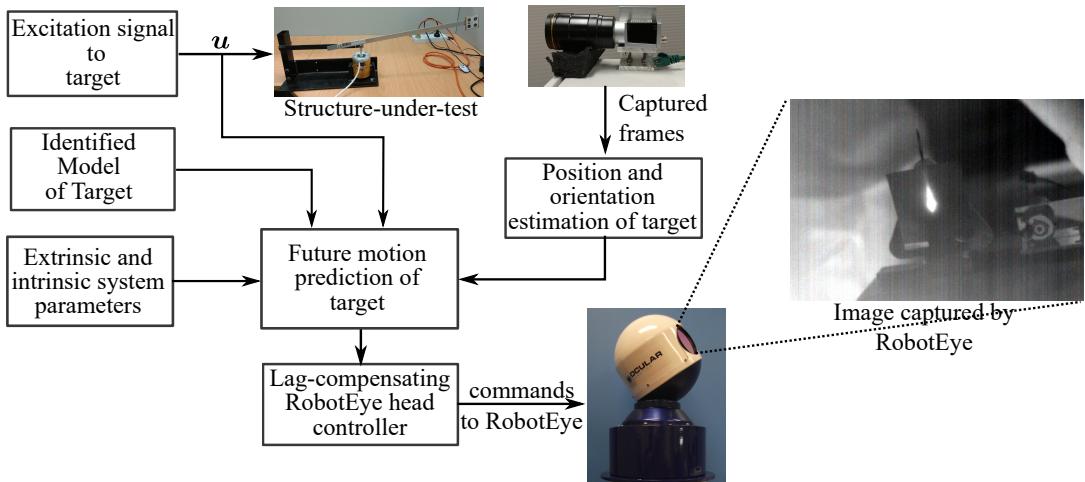


Figure 6.5 : System architecture for thermal target tracking system

6.3 Marker Detection System

Marker detection system basically consists of a fiducial and detection system for estimating the position of the target. The fiducial considered in this application consists of concentric circles, details on its design procedure and the detection methodology are discussed in the following subsection. Furthermore, timing requirement of the processes within the system is very strict in this application. In other words, as shown in Table 6.1, the frame rate of the vision camera is 100 fps. Therefore, all the processes related to the processing of images taken from the camera has to be completed within 10 ms. As a result, the fiducial detection system is developed in such a way that it satisfies the timing requirement of the system.

6.3.1 Fiducial design

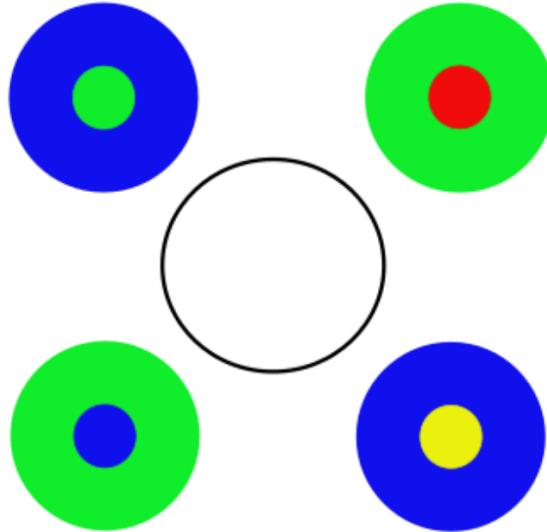


Figure 6.6 : The proposed concentric circle based fiducial for the marker detection system.

The proposed marker, as shown in Fig. 6.6, consists of concentric circular patches. The outer circles are made up of two distinct colours, blue and green

in this case. The rationale for its choice is that if the inner circles can be searched for by using only the pixels consisting of the colour of outer circles, then the incorrect detection of circles outside the pattern region of interest is significantly reduced.

The flow chart of detection algorithm for the proposed marker is presented in the Fig. 6.7

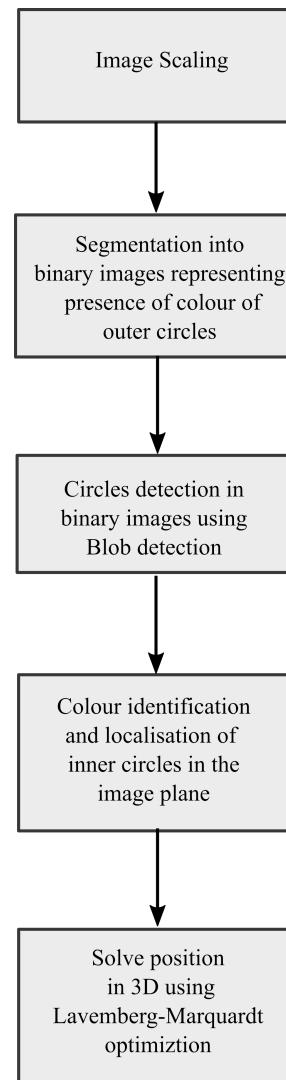


Figure 6.7 : Flow chart for marker detection algorithm.

6.3.2 Marker detection and pose estimation of target

The algorithm can be divided into following components:

1. Image segmentation
2. Inner circle detection
3. Circle localisation and pose estimation

Image Segmentation

In this step, the images captured by the camera is first scaled down by a factor of 4:1. The scaling effect reduces the computational latency of the detection algorithm. After scaling, two binary images are produced to represent the presence or absence of colour components of the outer circles. In order to obtain the binary images, we apply a threshold on every pixel of the scaled image. The threshold depends on the conventional Hue-Saturation-Value (HSV) model of the colour of the outer circles, where H corresponds a pure colour with $H=0$ referred to Red, S describes the whiteness with $S=0$ for White, and V is for darkness with $V=0$ for Black. Figure 6.8 represents the segmentation step.

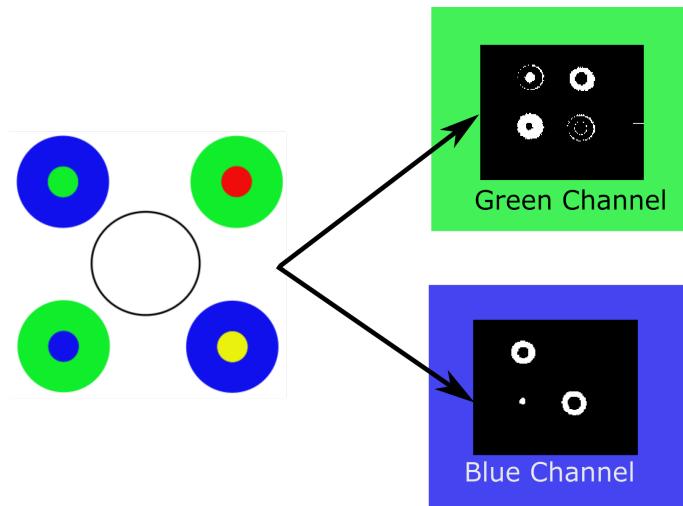


Figure 6.8 : Image segmentation of the captured images into green and blue channel using HSV.

Image Circle Detection

The inner circles are detected on the segmented image by using a blob detection algorithm. Based on the general concept that blobs of an image are meaningful regions to interpret its main features. Here, the blob detection algorithm (BDA) is based on the contour detection for extracting distinct regions from the background. This algorithm first detects a contour in the provided image, which is not necessarily a circle but can be elliptical patches. Then, the BDA connects the overlapped contours. This method is invariant to the circle deformation due to rotation of the marker. Therefore it is suitable for inner circle identification of the marker. The identification of the inner coloured circles using the blob detection algorithm is presented in Fig. 6.9.

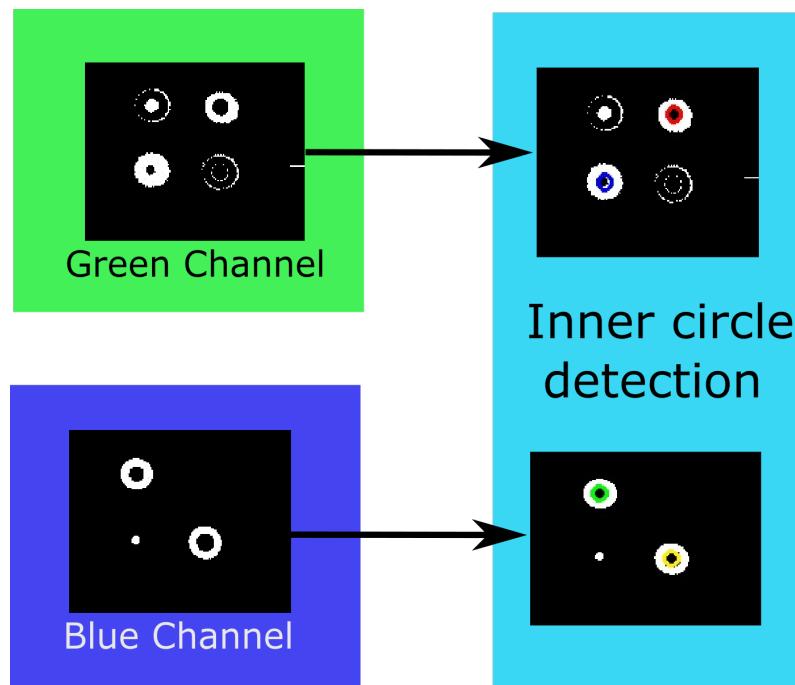


Figure 6.9 : Inner circle detection using the blob detection algorithm.

Circle localisation and pose estimation

After detecting circles in the binary images, the inner circles are identified by checking the H-S-V value of the center of the circles in the original image. The inner circles are identified in the image plane if the H-S-V values lies within the pre-defined colour threshold for the inner circles. In other words, suppose x_i is the centre of a detected circle in the binary images and $h() : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ is the function mapping a pixel to the H-S-V value, then the detected centre is identified as the k^{th} inner circle if the condition $\lambda_{k(min)} \leq h(x_i) \leq \lambda_{k(max)}$ ($k = 1, 2, 3, 4$) is satisfied, where $\lambda_{k(min)}$ and $\lambda_{k(max)}$ are the minimum and maximum colour threshold for the k^{th} inner circle, respectively. After the localisation of the inner circles in the image plane, the orientation and position of the marker are estimated by solving the Levenberg-Marquardt optimisation for the non-linear least square minimisation problem:

$$\mathbf{P} = \arg \min \sum_{j=1}^N d(\mathbf{x}_j, \mathbf{P}\mathbf{X}_j) \quad (6.4)$$

where \mathbf{x}_j and \mathbf{X}_j are, respectively, the image point in the image plane and the scene point in the world coordinate system, and $d(\mathbf{x}, \mathbf{y})$ is an Euclidian distance function between vector \mathbf{x} and \mathbf{y} , and \mathbf{P} is the projection matrix. The projection matrix \mathbf{P} is given by

$$\mathbf{P} = \mathbf{K} [\mathbf{R} \mathbf{t}] = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} [\mathbf{R} \mathbf{t}] \quad (6.5)$$

where, f_x and f_y are respectively the focal length in x and y direction, c_x and c_y are the principal axes. Similarly, \mathbf{R} is the rotational matrix and \mathbf{t} is the translation vector from camera to the pattern. The camera matrix \mathbf{K} is obtained from the camera calibration and should be considered in the optimisation. The library of OpenCV provides the interfaces for the optimisation. The solver solves for the \mathbf{R} and \mathbf{t} after the recursive optimization. Figure 6.10 summarises the marker detection process.

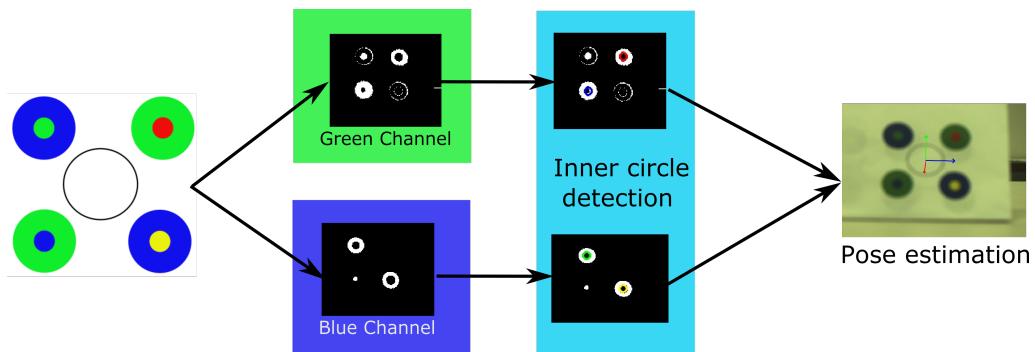


Figure 6.10 : Orientation and position estimation of the fiducial marker in 3D space.

6.3.3 Experiment with other marker detection systems

In this subsection, the benchmarking of the proposed marker detection system with standard marker detection systems, such as the Aruco and AprilTags markers is presented.

Aruco Marker system

Aruco marker system uses the binary-coded markers, as shown in the Fig. 6.11 (a). To correct the binary codes in the detected image, hamming codes can be used as reported in (Garrido-Jurado et al., 2014). The detection algorithm for a marker system consist of three stages, namely (i) image segmentation (ii) contour extraction and (iii) marker identification. The Aruco marker is one of the standard marker systems for camera calibration. Its interfaces are included in the OpenCV 3.1.0 library.

AprilTags marker system

AprilTags is also one of the widely used marker systems. A marker of Apriltags consists of binary codes, as depicted in Fig. 6.11(b). In the detection algorithms of the marker system: first, the line segments are detected, followed by quad detection, and finally, the binary codes are extracted and corrected (Olson, 2011).

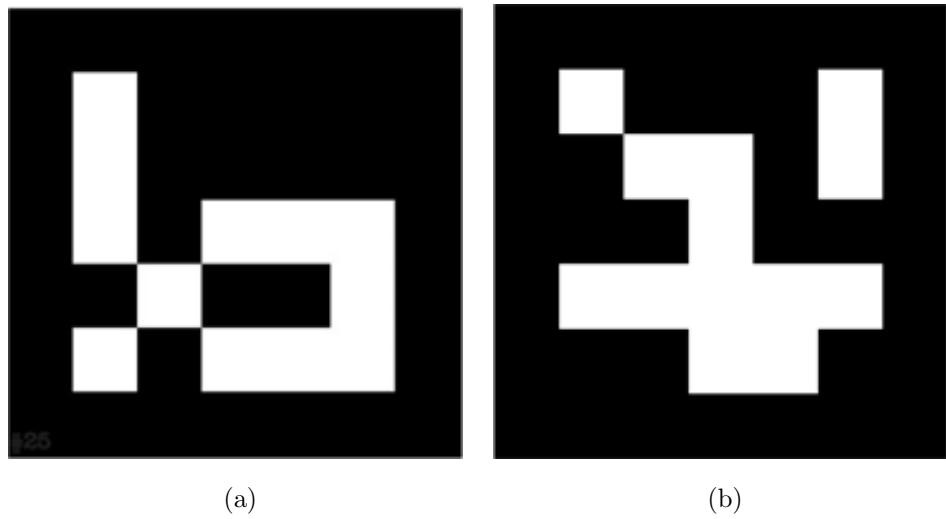


Figure 6.11 : Marker detection systems: a) Aruco marker b) AprilTag marker system.

Comparison results

The setup for the experiments is shown in Fig. 6.12. The sizes of markers used in the experiment are listed in Table 6.2 (a). In the experiments, a high frame rate camera was used to take images of the markers. The camera operates at 100 frames per seconds. Furthermore, in the experiment the depth of field for the camera was set to 50 cm with the lower limit of 50 cm and the upper limit of 100 cm. The specifications of the camera are listed in Table 6.2. Similarly, the algorithms for markers detection were tested in a PC with Intel ® Core™ i7 CPU operating at the clock frequency of 3.4 GHz, and 16 GB RAM.

Marker	Size
Aruco	5.5×5.5 cm
AprilTag	5×5 cm
Proposed	5×5 cm

Table 6.2 : Sizes of the markers during experiment.

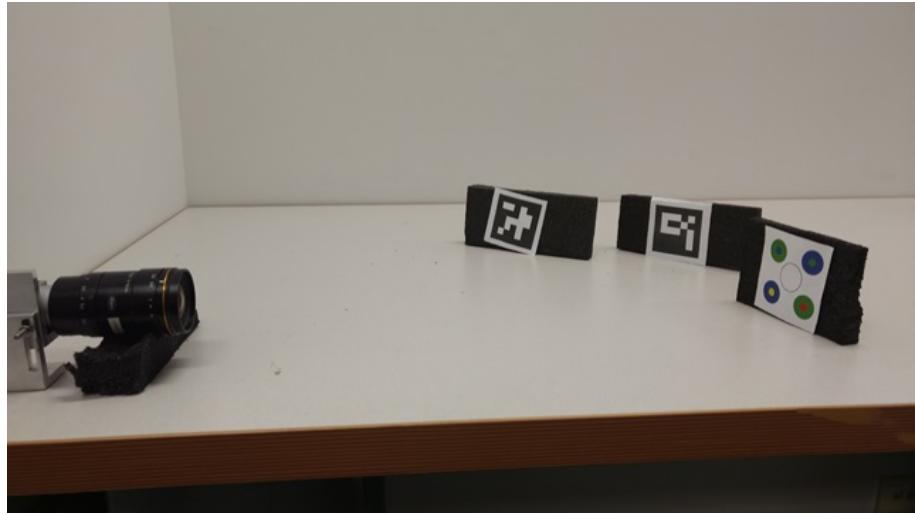


Figure 6.12 : Experimental setup for benchmarking the proposed marker detection system.

The experiments were performed by varying the distance of the markers from the camera. The markers were placed at the distance of 40 cm, 50 cm, 90 cm, and 120 cm, respectively. The results of the experiments are presented in Fig. 6.13 and Fig. 6.14, respectively. Fig. 6.13 shows the detection ratio of the fiducial systems. The detection ratio for the individual marker systems is calculated by taking the ratio of the number of detected frames over total frames during an experiment. From the figure it is clear that the detection ratio for our proposed marker system and the Aruco marker system is better than AprilTags, when the markers are placed from 40 cm to 90 cm from the camera. However, the performance of the detection algorithms decreases when the markers are placed beyond the far limit of the depth of field of camera, i.e. 100 cm. Even in this case, the detection ratio of the proposed system is still better than Aruco markers.

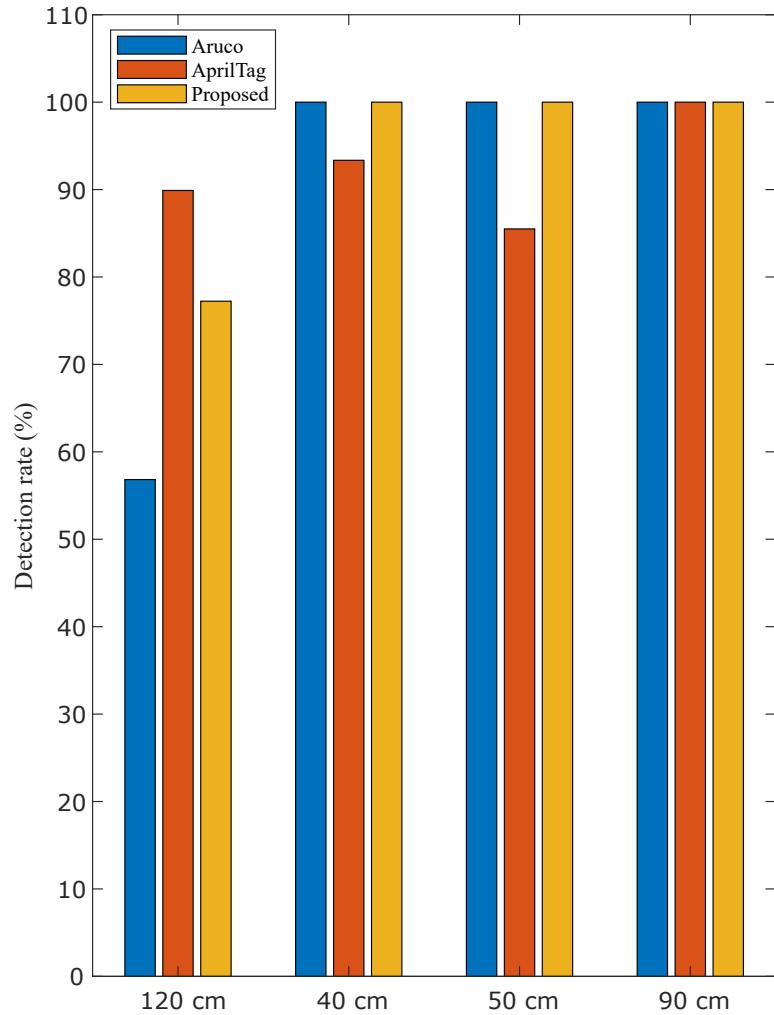


Figure 6.13 : Performance evaluations of the marker systems in terms of detection rate.

Figure 6.14 shows the average computation time for all the fiducial tracking systems. The computational time is evaluated in milliseconds while the vertical bars in the graph shows the standard deviation in the marker detection time. It is evident from the figure that the proposed marker detection system performs much better compared to the other standard markers. For instance, the detection time for the proposed algorithm is 5 ms, compared to 35 ms and 28 ms for the Apriltags and Aruco marker systems, respectively. In addition, the standard deviation for

the computation time is also small compared to other standard methods. Although the computation time increases slightly when the marker is beyond the upper limit of the depth of field of the camera, i.e. 100 cm from the camera, the low latency in terms of computation time is still upheld. Notably, the computation time for the Aruco marker is less than that of the AprilTags, but the standard deviation is much greater. From this experiment, it can be concluded that our proposed marker detection system can detect and localise marker in less than 10 ms to meet the requirement of high-speed real-time vision-based systems.

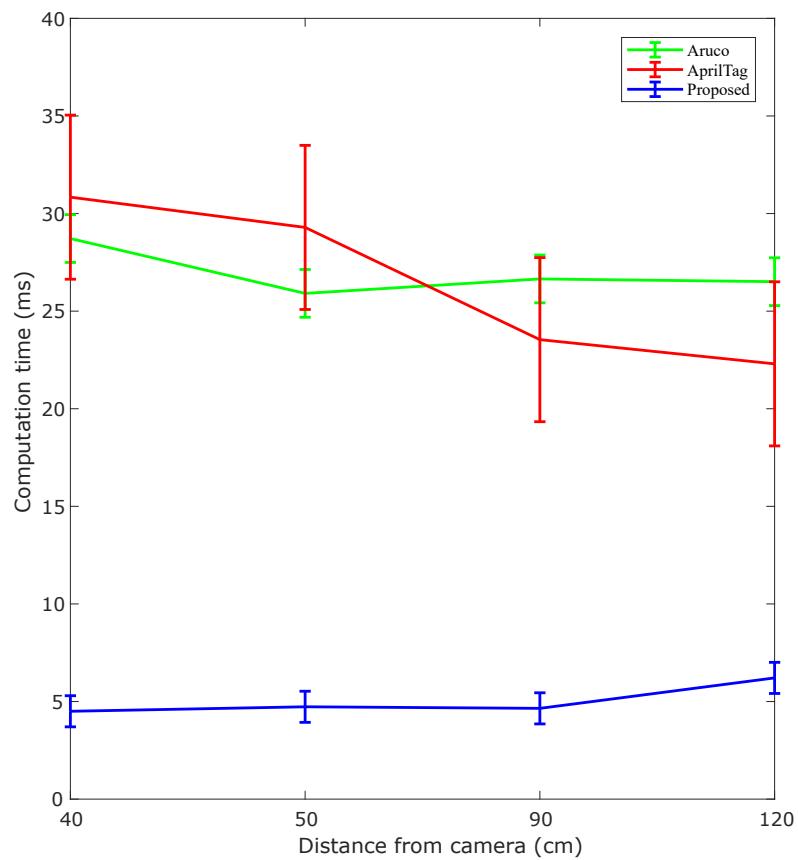


Figure 6.14 : Computation time for Aruco, AprilTag, and the proposed marker detection system.

6.4 Target Motion Modeling and Identification

The structure considered for the verification of the prototype tracking system is shown in Fig. 6.15. The structure consists of a cantilever and a linear actuator. One end of the cantilever is fixed whereas the other end consists of a linear motor.

As mentioned earlier in the system architecture (Section 6.2.2), the system requires the motion model of the structure-under-test to predict its future trajectory using Kalman filter. Therefore, accurate identification of the model is essential for the performance of the system, for which LS-SVR based identification method is applied in this project. The proposed identification method has been discussed detailedly in Chapter 3. However, in this section, implementation of the identification method for the test-specimen is presented, along with the comparison with other methods such as MOESP, CVA, and SMIPCA-E.



Figure 6.15 : Cantilever whose motion-model is under study.

6.4.1 Experimental Setup and data gathering

The experimental setup for gathering input-output data for the structure-under-test is presented in Fig. 6.16, which consists of a marker-based position estimation system, vision camera, National Instrument data acquisition board (DAQ), and an amplifier. The marker is attached to the cantilever via a stick, as shown in Fig.

6.15. Its details are provided in Section 6.3. Similarly, images of the fiducial marker are acquired by the camera (see Fig. 6.16 (b)), and its specification is provided in the table 6.1. In addition, commands are given to the motor through an amplifier, as shown in Fig. 6.16 (a), which is interfaced to a computer through the DAQ (Fig. 6.16 (a)). The measurement of the lever position was performed at the sampling rate of 100 Hz.

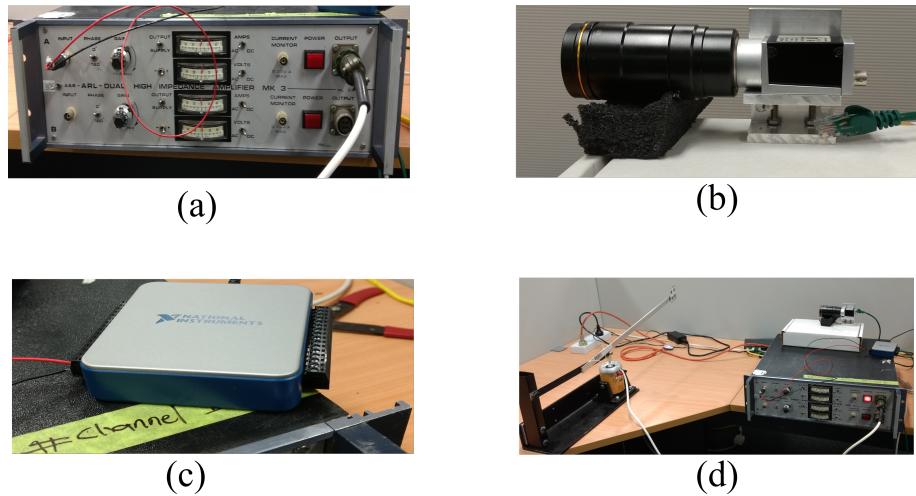


Figure 6.16 : Components in experiment: (a) Amplifier for motor, (b) 100 fps camera (c), National Instrument DAQ, and (d) experimental setup.

For the identification, two different experiments were conducted by applying a random and a sinusoidal signal. Details on the results along with the comparison with MOESP, CVA, and SMIPCA-E are provided in the following subsections.

6.4.2 Identification using random signals

In this case, the cantilever system is excited by a random signal with the standard deviation of 1V. The responses of the system in y and x -axis of the camera reference frame are presented in the figure 6.17. The variation in y axis and x axis signal is around 0.2 cm and 0.05 cm, respectively.

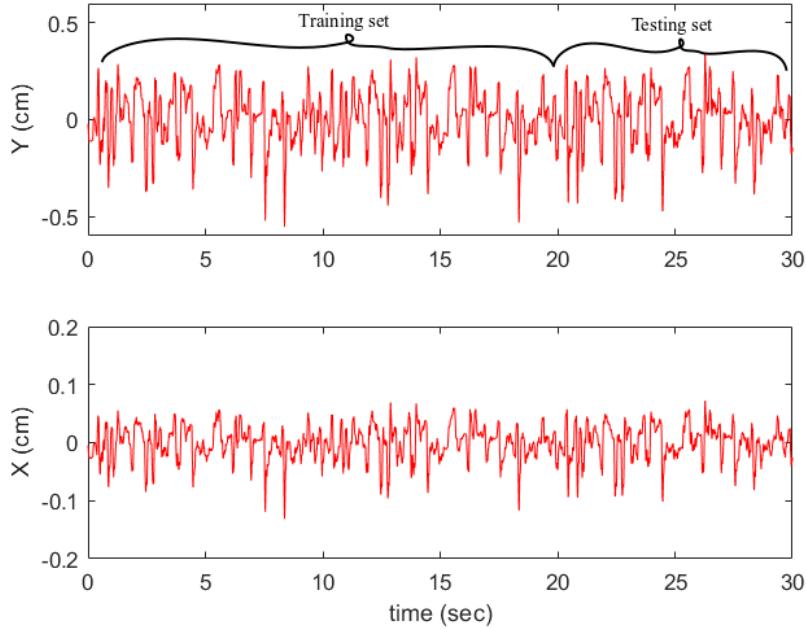


Figure 6.17 : Output of the cantilever system with random input.

The whole dataset is divided into two parts: training set and testing set. The training set is used for the identification and the testing set is used for the verification of the identified model. In this case, data samples of upto 20 sec are used for training while between 20 to 30 sec are used for testing.

In order to compare the methods, the order of the system should be determined first. As the determination of a system order of many practical applications can be quite often proceeded with the Akaike Information Criterion (AIC) (see, e.g., Wang and Qin (2002)), it is also used for this cantilever system. Here the AIC is defined as

$$\text{AIC}(n_x) = N_s \left(n_u (1 + 2\pi) + \ln \left(\left| \frac{1}{N_s} \sum_{k=1}^N \mathbf{e}_{n_y}(k) \mathbf{e}_{n_y}(k)^\top \right| \right) \right) + 2\delta_{n_x} M_{n_x}, \quad (6.6)$$

where N_s is the number of samples,

$$M_{n_x} = 2n_x n_u + \frac{n_u(n_u + 1)}{2} + (n_x + n_u)n_y,$$

$$\delta_{n_x} = \frac{N}{N - \left(\frac{M_{n_x}}{n_u} + \left(\frac{n_u+1}{2} \right) \right)},$$

and $\mathbf{e}_{n_y}(k) = \mathbf{y}(k) - \hat{\mathbf{y}}(k)$. According to the criterion, the order is obtained by finding an integer n_x that results in the minimum value of AIC. In other words,

$$\hat{n}_x = \arg \min \text{AIC}(n_x). \quad (6.7)$$

Figure 6.18 shows the AIC for various n_x , which indicates that the criterion is minimum for $n_x = 5$. Therefore in the identification methods order of the system was considered to be 5.

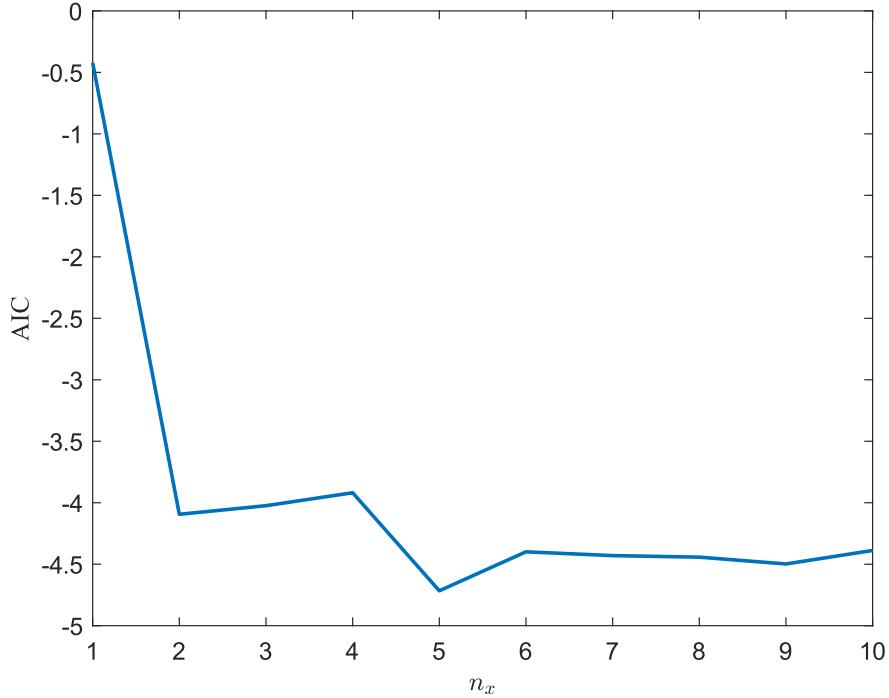


Figure 6.18 : AIC for different values of n_x .

Figure 6.19 shows the responses of the estimated models from all the considered algorithms for $h = 10$ and $N = 800$. The figure clearly demonstrates that the results for the proposed, MOESP, and CVA are close to the actual one, except for the SMIPCA-E.

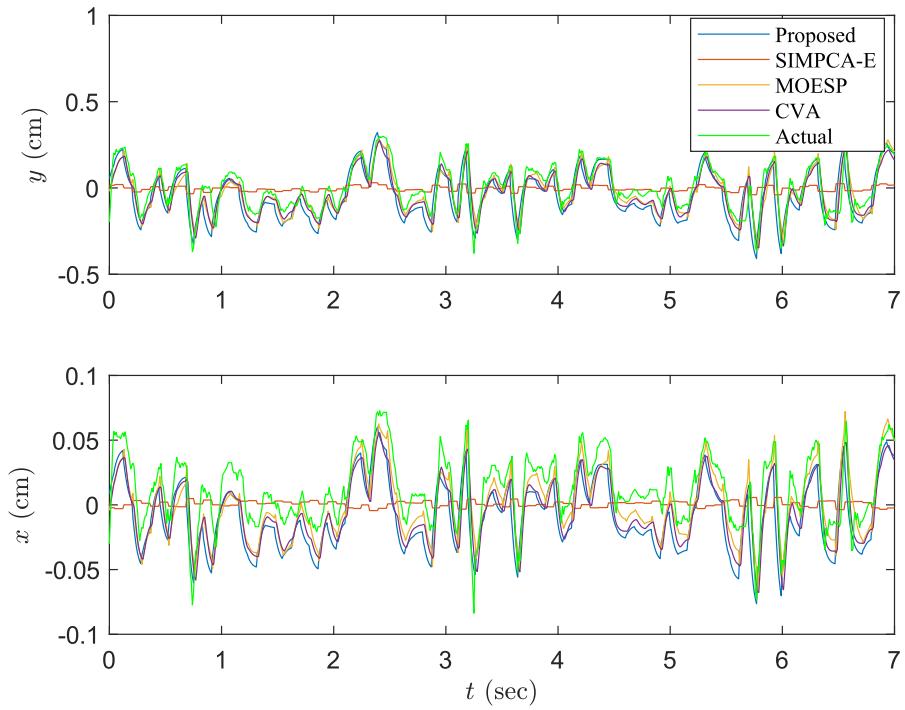


Figure 6.19 : Predicted and actual output for (a) LS-SVR based method, (b) SMIPCA-E, c) MOESP, and d) CVA.

6.4.3 Identification of sinusoidal input

In this scenario, a sinusoidal signal of the frequency 1 Hz and the amplitude of 1 V is applied to the system. The proposed identification method is then applied on the data sequences to estimate the state-space model of the system. Figure 6.20 shows response of the model along with the real response. The figure also represents the responses of the models identified by using MOESP, CVA, and SMIPCA-E methods. This results clearly demonstrate that the responses of estimated models by the proposed, MOESP, and CVA is close to the actual one, which signifies that the models closely represent the actual system. However, the response of the model is clearly identified by SMIPCA-E.

To better quantify the result, one can consider a performance metric. In this project, the performance indices the mean of Relative Square Error (MRSE) and

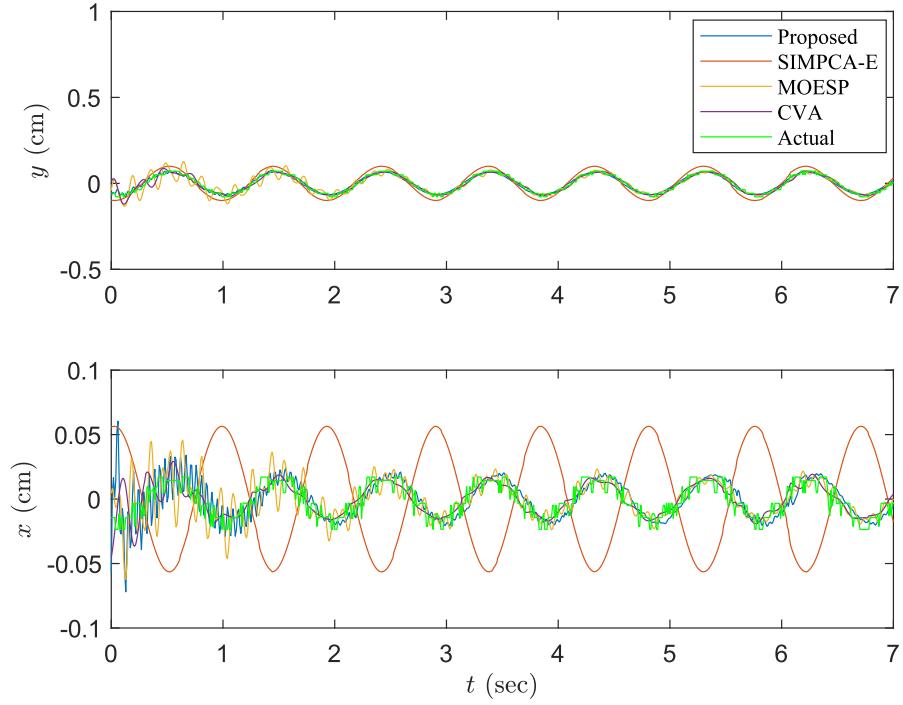


Figure 6.20 : Predicted and actual output for (a) the proposed method, (b) SMIPCA-E , c) MOESP, and d) CVA.

mean variance-accounted-for (MVAF) (Sotomayor et al., 2003) are considered, which are defined as

$$\begin{aligned} \text{MRSE} &= \frac{1}{n_y} \sum_{j=1}^{n_y} \sqrt{\frac{\sum_{i=0}^l e_j^2[i]}{\sum_{i=0}^l y_j^2[i]}} \\ \text{MVAF} &= \frac{1}{n_y} \sum_{j=1}^{n_y} \left(1 - \frac{\text{variance}(\mathbf{e}_j)}{\text{variance}(\mathbf{y}_j)} \right) \end{aligned} \quad (6.8)$$

where l is the number of samples, $e_j[i] = y_j[i] - \hat{y}_j[i]$ is the difference of the response of identified model from the actual output. Similarly, $\mathbf{e}_j = [e_j[1], e_j[2], \dots, e_j[l]]$

Table 6.3 shows the performance indices for both experiments, where Case 1 and Case 2 represent the identification scenarios with the random and sinusoidal inputs, respectively. The table clearly highlights the advantages of the proposed identification algorithm over other methods. For instance, the MRSE value for the proposed method for Case 2 is 0.56, compared to 0.65 and 0.63 for MOESP

and CVA, respectively. Similarly, the performance for the proposed LS-SVR based identification method is 0.56 in Case 1, which is the lowest in comparison to other methods. Furthermore, the MVAF values for the proposed algorithm is highest in both cases, which indicates that the deviation of the output of the identified model is much lower than the actual data. Nevertheless, the identification algorithm based on the PCA showed lower performance in the real-time data. The result is consistent with the poor prediction performance, which is depicted in Figure 6.19 and 6.20, respectively.

Algorithms	MRSE		MVAF	
	Case 1	Case 2	Case 1	Case 2
SMIPCA-E	0.98	0.95	0.63	0.65
Proposed	0.56	0.51	0.68	0.71
MOESP	0.61	0.65	0.62	0.61
CVA	0.59	0.63	0.65	0.66

Table 6.3 : MRSE results for random and sinusoidal input cases.

6.5 Controllers and their Comparisons

6.5.1 Fast Terminal Sliding Mode (FTSM)

This subsection presents the performance evaluation of the implementation of FTSM controller on the pointing device. The FTSM controller is already introduced in Chapter 5. The proposed controller is implemented on the system (5.30) whose parameters are given in Table 5.1. The architecture of the controller for the pointing system is presented in Fig. 6.21. The control system consists of controllers for Azimuth and Elevation angles, respectively, which implement the discrete-time FTSM control law (5.39). Similarly, the controller requires the estimation of states

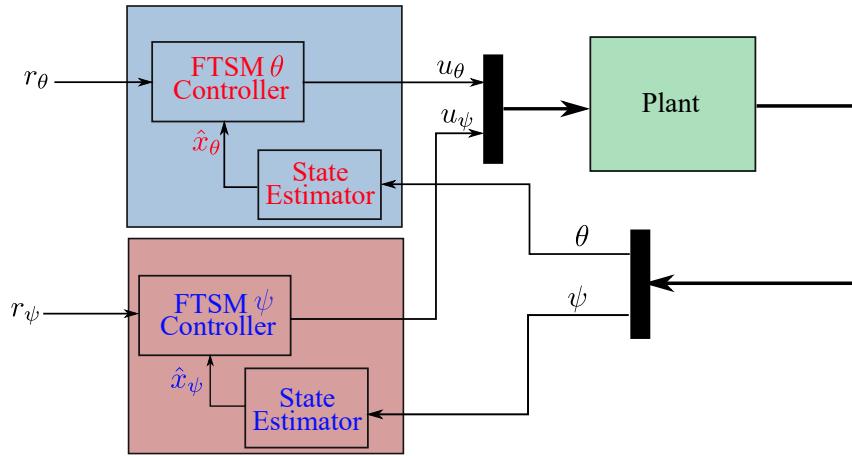


Figure 6.21 : Controller architecture for the pointing system using FTSM

x_1 and x_2 , which is performed by the estimation filters.

Evaluation of the controller is performed in simulations, and the controllers are applied for the tracking of the following signals:

- Step signals
- Sinusoidal signals.

Similarly, gains of the controller were set as: $K = 10$, $\alpha = 1$, $\beta = 2$, $q_1 = 7$, $p_1 = 9$, and $\Phi = 70$. Sampling period for the controller was set to 0.01s. In addition, disturbances of 0.1 degree was also added to the system to judge the robustness.

Tracking response

The step value for the reference signal is 10° . Figure 6.22 shows the tracking response of controller, which clearly indicates that the system is able to track the reference signal. Settling time for the azimuth and elevation is around 0.04 sec, which can be observed in the zoomed section of the step response.

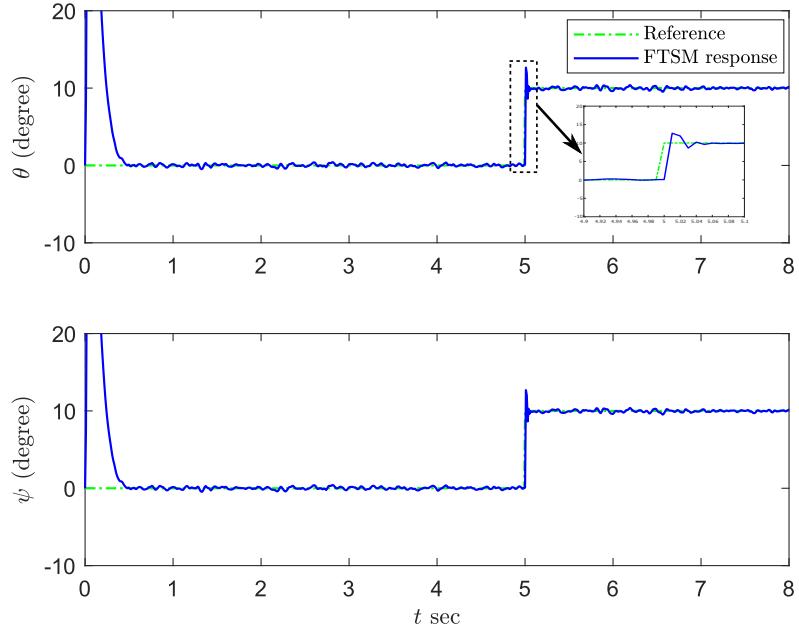


Figure 6.22 : Step reference signal tracking by elevation and azimuth angles.

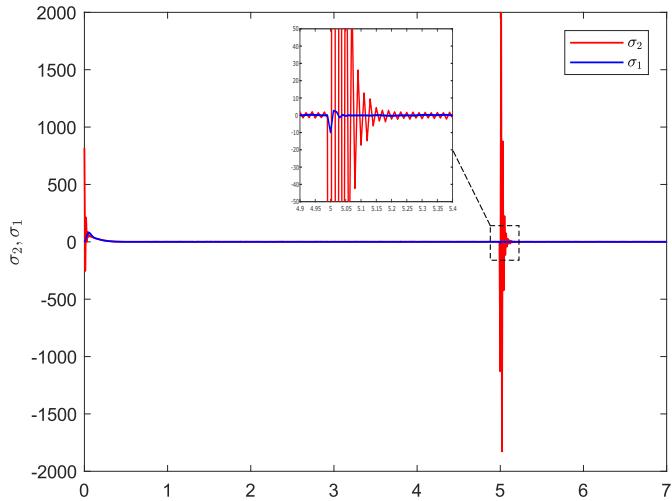


Figure 6.23 : σ_1 and σ_2 for elevation angle.

Figure 6.23 shows the sliding surfaces σ_1 and σ_2 for the elevation angle. From the figure it is clear that σ_2 converges towards sliding mode, that is $\sigma_2 = 0$. The signal, then, remains bounded within the region $-10 < \sigma_2 < 10$. Furthermore, it

can also be observed that σ_1 is also bounded in the steady-state condition, which is much lower than σ_2 .

Tracking response of the system for sinusoidal signal is presented in Fig. 6.24. The amplitude and frequency of the reference is 30° and 1Hz, respectively. The tracking response of the system is close to the step signal which can be verified by observing at the zoomed section of the plot.

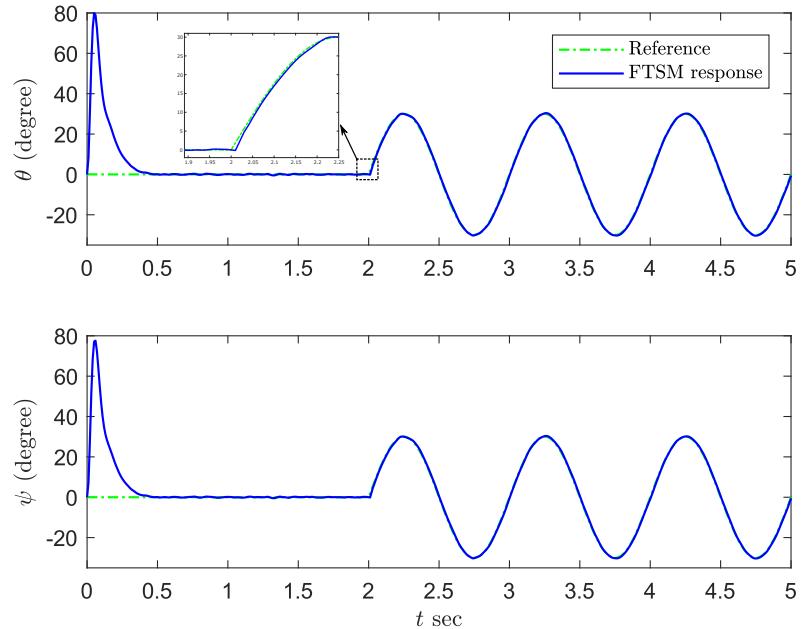


Figure 6.24 : Sinusoidal reference signal tracking by the proposed FTSM controller.

6.5.2 Terminal Sliding Mode (TSM)

For the comparison, the TSM controller proposed by Li et al. (2014) is considered for this system, whose sliding surfaces are defined as:

$$\begin{aligned}\sigma_1[n] &= x_1[n] - r[n] \\ \sigma_2[n] &= \Delta\sigma_1[n] + \beta\sigma_1^{q_1/p_1}[n],\end{aligned}\tag{6.9}$$

which is similar to FTSM, except for the missing linear attractor term, α . Similarly, the control law for the TSM controller is given by:

$$u = -\frac{u_0}{b_0}$$

$$u_0 = -a_1x_1[n] - a_2x_2[n] - \Delta^2(r[n]) + (\beta \frac{q_1}{p_1} \sigma_1^{q_1/p_1-1}) \Delta(\sigma_1^{q_1/p_1}) + K \text{sign}(\sigma_2), \quad (6.10)$$

It should be noted that the TSM control law (6.10) does not include the term Φ in comparison to the proposed FTSM control law (6.10.). As a result it affects the transient response of the system, particularly in the reaching mode of the sliding surface σ_2 . Details on the issue is discussed and analysed in the Section 5.6.2.

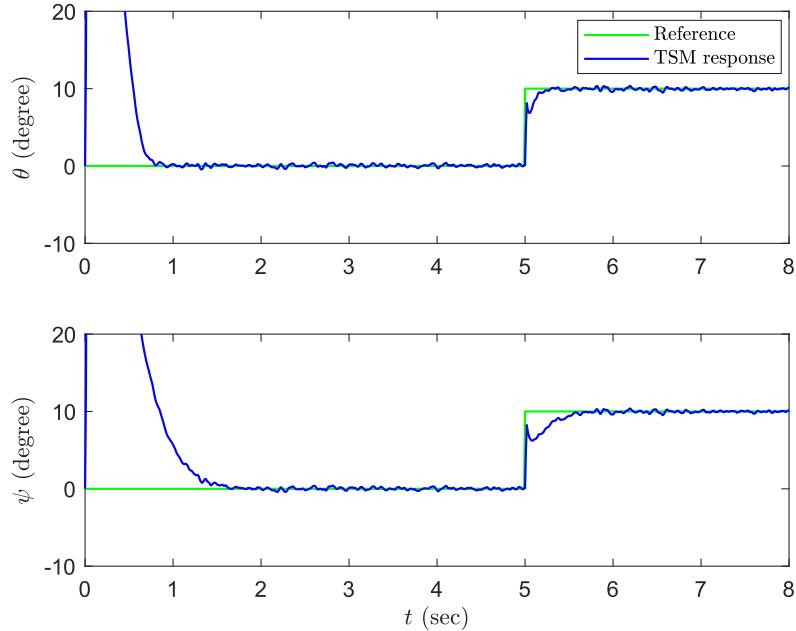


Figure 6.25 : Step signal tracking response for the TSM controller

Sampling period for the TSM controller during the simulation was set to 0.01 s. Similarly, gains were selected as $K = 10$, $\beta = 2$, $q_1 = 7$, and $p_1 = 9$, and $\Phi = 70$. Furthermore, disturbance of 0.1° was added to the system.

Figure 6.25 and 6.26 show the tracking results of the TSM controller for the step and sinusoidal reference signals, respectively. From the tracking response it is clear that the controller is able to track the reference signal. However the controller shows lower performance due to the above mentioned reason. Particularly, for the step reference signals, the elevation and azimuth angles take around 0.5 seconds and 0.8 seconds, respectively, to reach the desired angles, which is slower than the FTSM case.

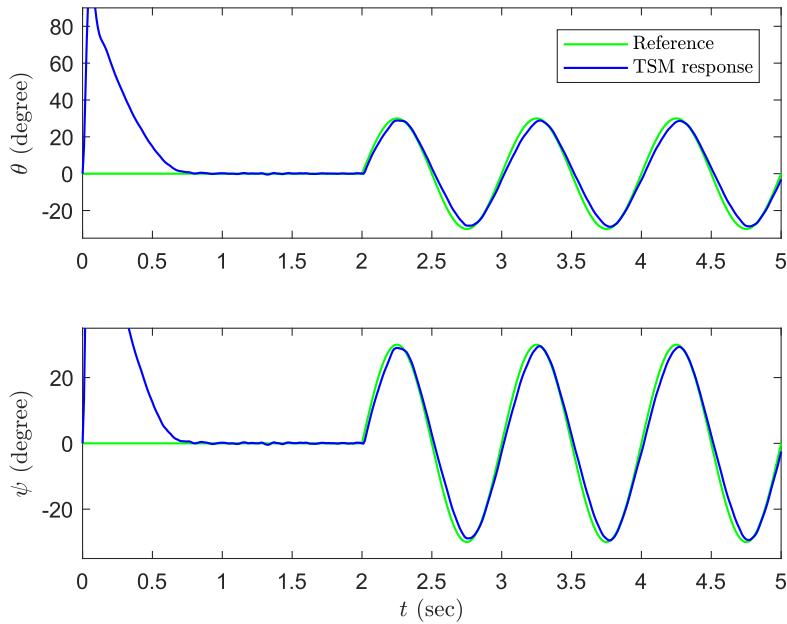


Figure 6.26 : Sinusoidal signal tracking response for the TSM controller.

6.5.3 Model Predictive Control (MPC)

Model Predictive Control (MPC) is a numerical optimization based optimal control method, where future inputs and outputs of a plant is predicted using its model, and then optimized at a regular interval based on a performance index.

MPC is applied to a state-space model of a plant. Therefore, in this application, state-space model that is presented in Section 5.5.2 , i.e. Equation (5.26), is consid-

ered for the pointing device where the input and output are given by $\mathbf{u} = [\theta_{cmd} \psi_{cmd}]$ and $\mathbf{y} = [\theta, \psi]$, respectively. It is noted that θ_{cmd} and ψ_{cmd} represent the command given to the pointing device via the application programming interfaces (API) of the device. In addition, the following output tracking problem is formulated to apply the predictive control.

$$\begin{aligned} V(\mathbf{x}) = & \frac{1}{2} \sum_{i=0}^{N-1} \{ (\mathbf{y}_i - \mathbf{r}_i)^\top Q_y (\mathbf{y}_i - \mathbf{r}_i) + \mathbf{u}_i^\top Q_u \mathbf{u}_i \} \\ & + (\mathbf{y}_N - \mathbf{r}_N)^\top Q_y (\mathbf{y}_N - \mathbf{r}_N) \\ \text{minimize } & u_{min} \leq \mathbf{u} \leq u_{max} \end{aligned} \quad (6.11)$$

subject to : $\mathbf{x}_{i+1} = A\mathbf{x}_i + B\mathbf{u}_i$

$$y_i = C\mathbf{x}_i$$

where, $\mathbf{y}_i = \mathbf{y}[n+i] \in \mathbb{R}^2$ is the i^{th} future output of the system, $\mathbf{u}_i = \mathbf{u}[n+i] \in \mathbb{R}^2$ is the i^{th} future input to the system, \mathbf{r}_i is the i^{th} future reference, N is the prediction horizon, $A \in \mathbb{R}^{4 \times 4}$ is the system matrix, $B \in \mathbb{R}^{4 \times 2}$ is the input matrix, and $C \in \mathbb{R}^{2 \times 4}$ is the output matrix.

As mentioned earlier in the Section 6.2.2, the objective of designing a control system in this application is to compensate lags or delays in the system. In this scenario, MPC can be considered a suitable option because in the cost function (6.11) future output references (\mathbf{r}_i) are also considered, which are neglected in the FTSM and TSM controllers. As a result, MPC controller can have better immunity over lags and delays in the network.

The cost function $V(\mathbf{x})$ in (6.11) can also be represented as

$$\begin{aligned} V(\mathbf{x}) = & (\mathbf{y}_0 - \mathbf{r}_0)^\top Q_n (\mathbf{y}_0 - \mathbf{r}_0) \\ & + (\mathbf{Y} - \mathbf{R})^\top \bar{Q}_y (\mathbf{Y} - \mathbf{R}) + \mathbf{U}^\top \bar{Q}_u \mathbf{U} \end{aligned} \quad (6.12)$$

where,

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_N \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_{N-1} \end{bmatrix},$$

and

$$\mathbf{R} = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \vdots \\ \mathbf{r}_N \end{bmatrix}.$$

Now, by utilising the system dynamics (5.26) and forward substitution one can get

$$\mathbf{Y} = \bar{\mathbf{S}}\mathbf{U} + \bar{\mathbf{T}}\mathbf{x}_0 \quad (6.13)$$

where,

$$\bar{\mathbf{S}} = \begin{bmatrix} CB & 0 & \dots \\ CAB & CB & \dots \\ CA^2B & CAB & \dots \\ CA^3B & CA^2B & CAB \\ \vdots & \vdots & \vdots \\ CA^{N-1}B & CA^{N-2}B & CA^{N-1}B & \dots & CB \end{bmatrix},$$

and

$$\bar{\mathbf{T}} = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^N \end{bmatrix}.$$

Now, by utilizing (6.12) and (6.13) one can formulate the MPC control problem (6.11) into the quadratic programming (QP) optimisation problem as

$$\underset{\mathbf{U}}{\text{minimize}} V(\mathbf{x}) = \frac{1}{2} \mathbf{U}^\top \bar{\mathbf{H}} \mathbf{U} + \bar{\mathbf{F}}^\top \mathbf{U} \quad (6.14)$$

subject to: $\bar{\mathbf{E}}\mathbf{U} \leq \boldsymbol{\delta}$

where,

$$\bar{H} = \bar{S}^\top \bar{Q} \bar{S} + \bar{R},$$

$$\bar{F}^\top = 2\bar{x}_0^\top \bar{T}^\top \bar{Q} \bar{S} - \bar{R}^\top \bar{Q} \bar{S}$$

,

$$\bar{E} = \begin{bmatrix} I \\ -I \\ \bar{S} \\ -\bar{S} \end{bmatrix},$$

and

$$\boldsymbol{\delta} = \begin{bmatrix} \Delta U_{\max} \\ -\Delta U_{\min} \\ \mathbf{Y}_{\max} - \bar{T}\mathbf{x}_0 \\ -\mathbf{Y}_{\min} + \bar{T}\mathbf{x}_0 \end{bmatrix}.$$

Now, this problem can be solved using the available quadratic programming (QP) solver.

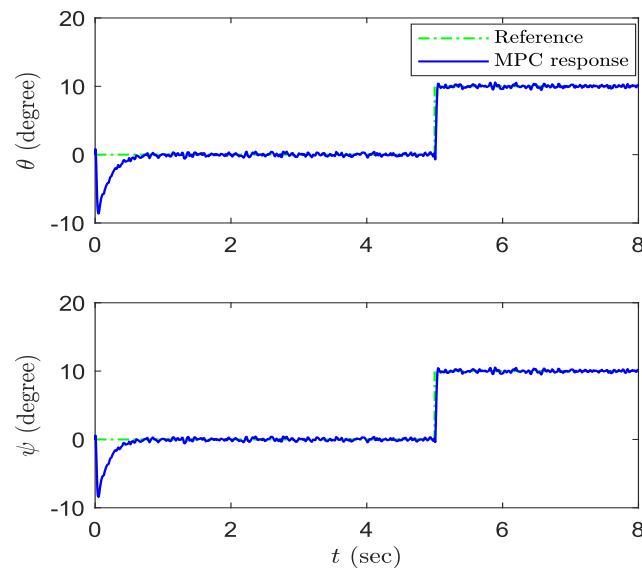


Figure 6.27 : Plot of the response of MPC controller for step reference signals.

Figure 6.27 and 6.28 depict the results of the MPC controller for the step and sinusoidal reference signals in the simulation, where the sampling time for the MPC controller was set to 0.01 sec. The figure indicates the actual signals are close to the reference signals with the settling time of around 0.05 sec for both azimuth and elevation angles. However, some fluctuations can be observed in the steady state condition, particularly for the step reference signal, which is due to the added disturbances to the system during simulation.

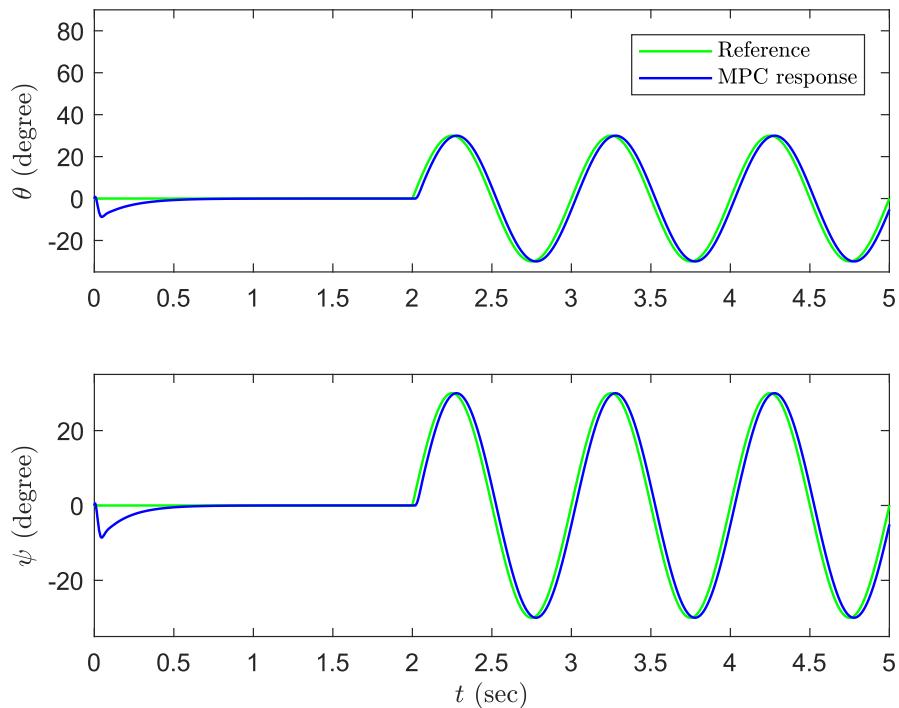


Figure 6.28 : Elevation and azimuth response of MPC control system for sinusoidal reference signals.

6.5.4 Simulation and Comparison of TSM, FTSM, and MPC

Figure 6.29 and 6.30 depict the comparison of the FTSM, TSM and MPC controllers. The tracking responses indicate that the FTSM control law shows better performance over MPC and TSM controllers. For instance, settling time for the FTSM while tracking step signal is 0.04 sec, compared to the 0.05 sec and 0.1 sec,

respectively, for MPC and TSM. Furthermore, response of the FTSM for the sinusoidal signal is close to the reference signal compared to other methods.

Integral of Square Error (ISE)

In order to evaluate the steady state error for all the control methods we consider the Integral of Square Error (ISE) performance index. ISE error is defined as

$$\text{ISE} = \sum_{n=1}^M e^2[n], \quad (6.15)$$

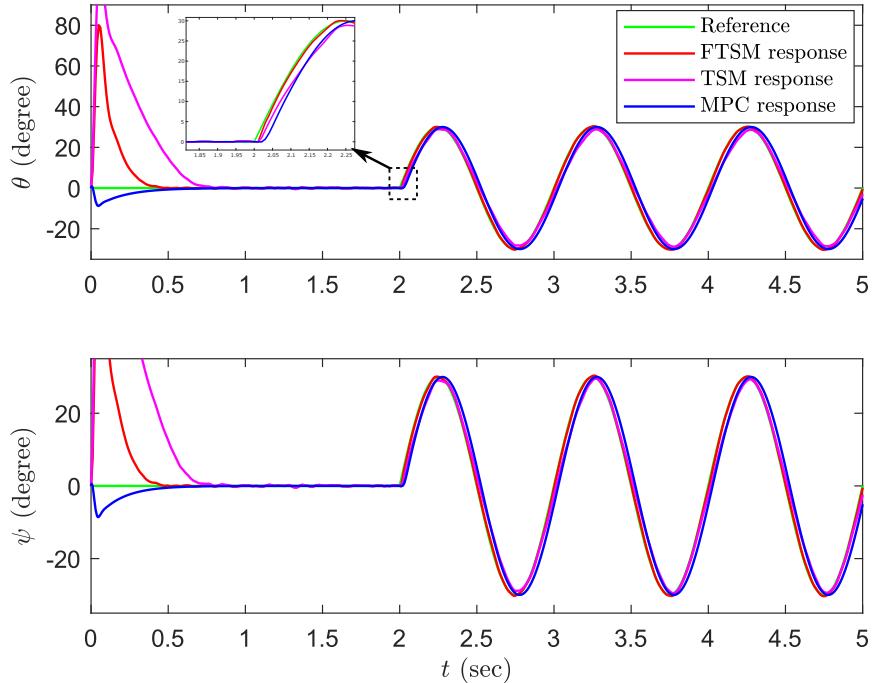


Figure 6.29 : Comparison of the proposed method with TSM and MPC for sinusoidal reference signal.

where $e[n]$ is the difference between actual and reference signal. Plots for the ISE for both elevation and azimuth angles are presented in Fig. 6.31 and 6.32, respectively. The ISE indicates that the FTSM has lowest values for both scenarios, followed by MPC and TSM controller.

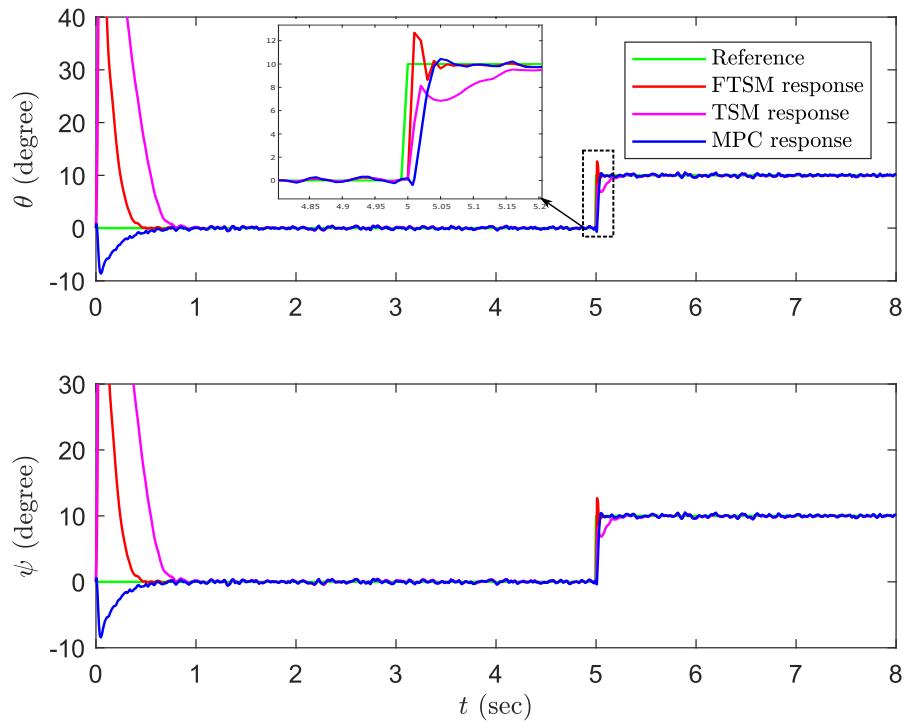


Figure 6.30 : Comparison of the proposed method with TSM and MPC for step reference signal.

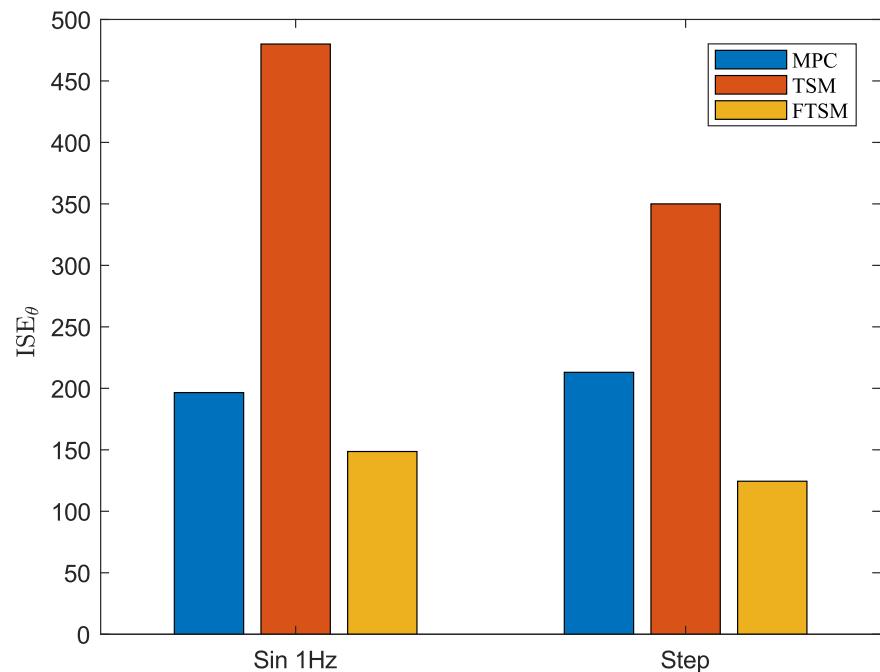


Figure 6.31 : Comparison of ISE for elevation angle.

As mentioned previously in Section 5.6, the proposed FTSM controller is robust, and its error is bounded within a level known as quasi sliding mode band. However, such boundedness of error cannot be guaranteed for MPC controller. As a result, FTSM has lower ISE value than MPC even though there is not much differences in their performance.

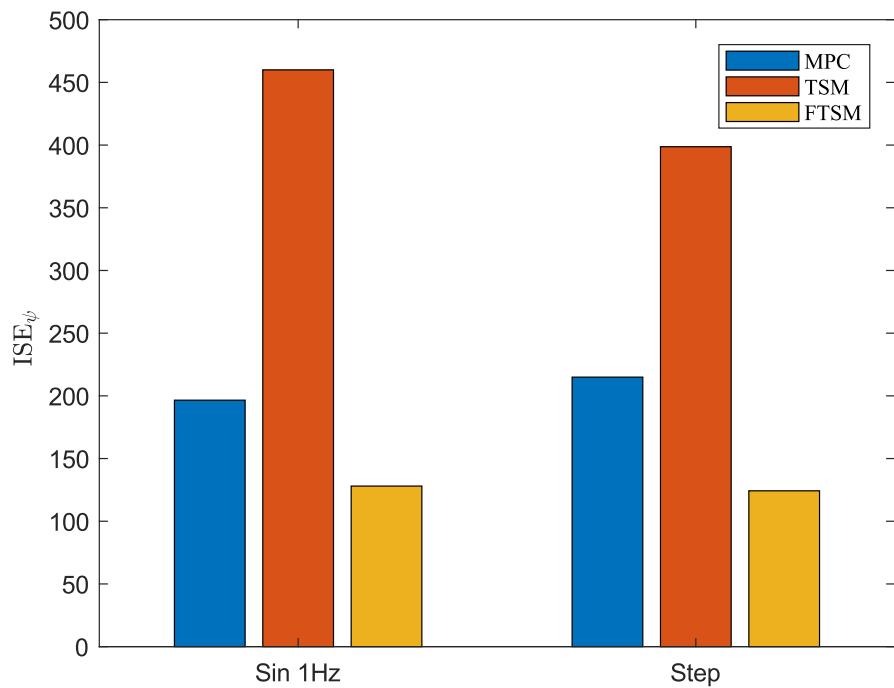


Figure 6.32 : Integral of Square Error for Azimuth angle.

The transient response and ISE performance index indicate that FTSM controller has faster response in comparison to MPC. However, the requirement of the control system in this application is to overcome network delays and lags. To solve the issue, as presented in Section 6.2.2, the proposed solution includes the prediction of the future trajectory of the target, and based on that generation of control signal for the pointing device. In this context MPC is a better option because it is formulated in such a way that it can track the reference signal (Equation 6.12). Therefore MPC is considered for the real-time implementation, and its results will be discussed in

the following subsection.

6.5.5 Experimental results

For the real-time testing, the MPC based predictive control algorithm was applied to a RobotEye with the specifications provided in Table 6.1. The block diagram for the experiment is provided in the Fig. 6.33. In the experiment, sinusoidal signals were provided as the reference signals. The frequencies of the reference signals were 1 Hz and 2 Hz for azimuth and elevation angles, respectively. To send the angular commands and receive the measurements we used the proprietary application programming interface. The tracking errors during the experiment are shown in Figure 6.34. The error is less than 0.2 degrees for elevation and 0.1 degrees for the azimuth angle. Similarly, ISE errors for the azimuth and elevation angles are 3.947 and 33.73 degree², respectively.

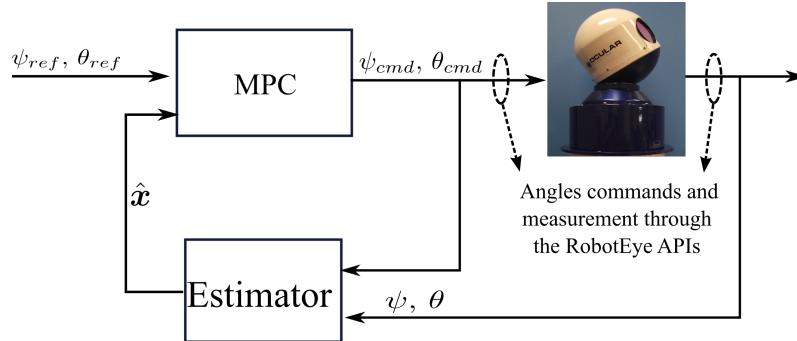


Figure 6.33 : Real time experiment for MPC.

6.6 Performance Evaluation of the Tracking System

In this section, we evaluate performance of the proposed system described above. In addition, this section also presents comparison of the performance with and without the predictive control. The experimental setup for the evaluation of the system is presented in the Figure 6.35 with the specifications of all the components in Table 6.1. The figure also shows the target consisting of the circular patched marker

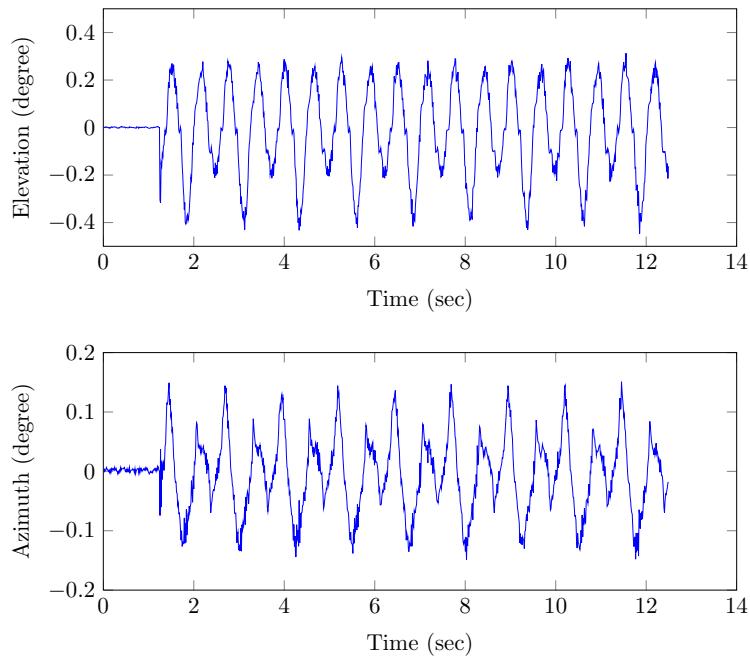


Figure 6.34 : Tracking errors in azimuth and elevation angles in real experiment.

with a heat source at the centre of the marker. The heat source was used to detect the target in the thermal camera. In this real time application, there is no direct way of evaluating the tracking performance of the system. However, as mentioned previously, the thermal camera is embedded inside RobotEye, and it has the view of the direction it is pointing at. Therefore, the thermal targets are detected in the images taken by thermal and vision cameras, and then compared for evaluation.

After the calculation of target points in both vision and thermal camera, one can evaluate the target tracking performance of the eye by comparing standard deviations of the target pixels in x and y-axis. Nevertheless, this approach does not give the clear indication of the performance. For example, if the variance of the target pixels in the vision camera is along y-axis but the variance of the pixels in the thermal camera is along x-axis then by comparing the above mentioned parameters, it does not reflect the performance. In such situation, to better evaluate the tracking performance, the principal component analysis (PCA) of the target points can be

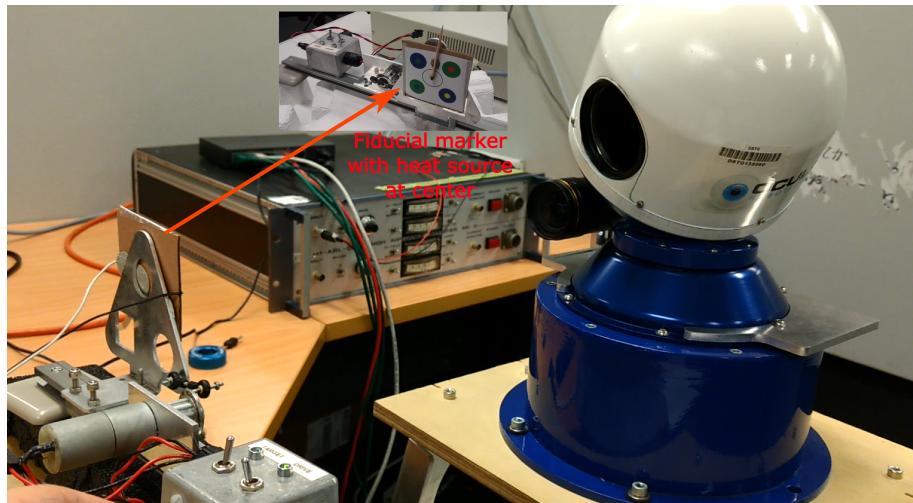


Figure 6.35 : Experimental setup for tracking system.

applied. An example of the PCA of the target points is presented in Fig. 6.36. From the figure, it is clear that the major axis and the minor axis of the PCA represents the direction of maximum and minimum variation of the target points. By using the principal component analysis, the following parameters can be evaluated:

- Maximum eigenvalues of the principal component analysis (PCA) of target pixels
- Minimum eigenvalues of the principal component analysis (PCA) of target pixels

From the calculation of the above mentioned parameters, ratios of the parameters are obtained. Table 2 shows the summaries of the performance evaluations. From the table, it can be observed that there are improvements in all parameters. For instance, in terms of ratio of maximum eigenvalues there is the improvement by 74 percent. Similarly, in the case of the ratio of standard deviation in x-axis the improvement is about 42 percent.

	Without MPC	With MPC	% improvement
Ratio of maximum eigenvalues of PCA (thermal/vision)	0.000105	0.000027	74
Ratio of minimum eigenvalues of PCA (thermal/vision)	1.2213	0.8455	30
Ratio of standard deviation in X-axis (thermal/vision)	0.0101	0.005	42
Ratio of standard deviation (thermal/vision)	1.00659	0.8153	19

Table 6.4 : Summaries of the performance of the thermal tracking system with and without MPC.

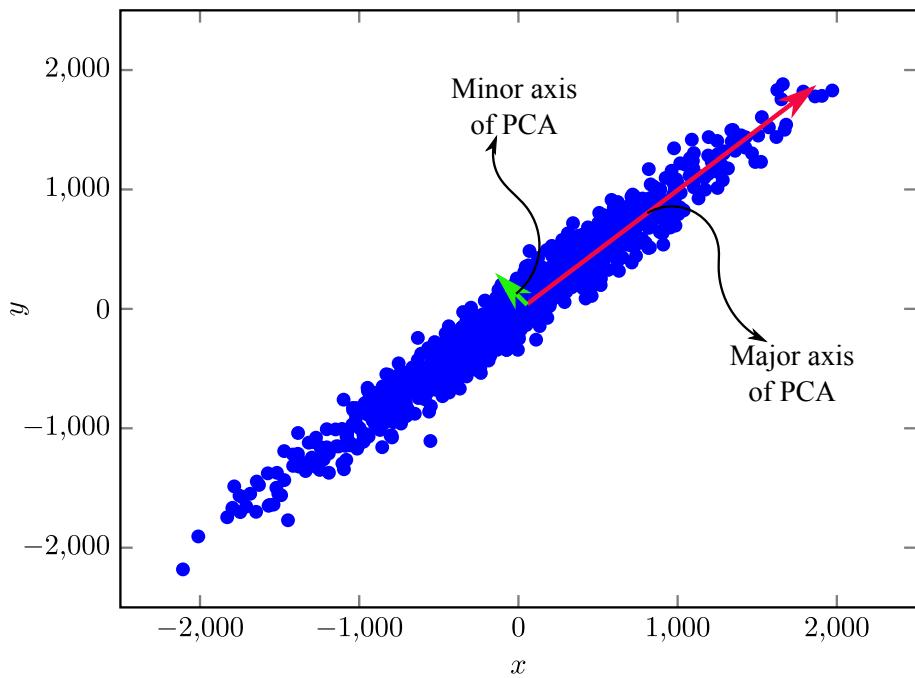


Figure 6.36 : An example of the PCA of target points showing major and minor axis.

6.7 Summary

A thermal target tracking system for thermoelastic stress analysis (TSA) using a mirror-based optical pointing device has been presented. The hardware for the system comprises of an optical pointing device by Ocular Robotics Pty. Ltd., a thermal camera, and a vision camera. In the proposed tracking system estimation of target's orientation and position is carried out using a marker detection system, followed by the prediction of its trajectory using a Kalman filter. The Kalman estimator requires a model of the target in state-space form, which is identified using the proposed LS-SVR based identification algorithm in Chapter 3. The result of the identification algorithm has been presented along with its comparison with other methods such as MOESP, CVA, etc.

The proposed system also consists of a lag-compensating control system for the

optical pointing device, for which three controllers have been studied, namely MPC controller which has been formulated to follow reference signals, a FTSM-based controller for the device which is presented in Chapter 5, and TSM controller. Comparisons of the methods in the simulation have also been presented in terms of the performance index ISE.

Finally, experiments have been conducted for the tracking system to evaluate its performance.

Chapter 7

Application of FTSM on Gantry cranes

7.1 Introduction

Automated cranes with a swing load are quite popular among underactuated electromechanical systems and have been the subject of many studies (Abdel-Rahman and Nayfeh, 2003). In crane automation, underactuation, parametric variations and external disturbances are quite often the reasons for difficulties in controlling the swinging load and trajectory following. For example, in (Kolar et al., 2017), a flatness-based control method is presented for gantry cranes to achieve control goals in minimum transition time. For crane control, a considerable amount of literature has been published to address the problem of suppressing the swinging load vibration. For instance, a controller to minimize the oscillation of the payload due to parametric resonance is presented in (Vázquez et al., 2013).

The problem of uncertainties in crane systems can be solved by employing robust controllers, for which the Sliding Mode Control (SMC) method is very promising. Indeed, many published studies have shown the advantage of such controllers. For example, a sliding mode controller is presented in (Lee, 2004) for the trolley and load-hoisting dynamics where load-swing dynamics was coupled into the sliding surface for the trolley motion. The undesirable chattering in SMC signals can be reduced with a boundary layer or by using a higher-order sliding mode controller (Levant, 2003). To this end, most widely-used for mechanical and mechatronic systems are second-order sliding mode techniques, see e.g. (Liu et al., 2014; Ha et al., 2013). In (Bartolini et al., 2002), second-order sliding mode controllers are designed for container cranes

in two phases, namely traveling and arrival phase. The modeling and trajectory tracking problem for the offshore cranes is addressed in (Raja Ismail et al., 2015) by using robust optimal sliding mode control. Techniques for adjusting control parameters have also been integrated into SMC for performance improvements. For example, in (Moon et al., 2013), a fuzzy disturbance estimator is coupled with the SMC technique to deal with the varying cable length in overhead cranes.

As mentioned previously in Chapter 4 and 5, conventional SMC utilizes linear sliding surfaces which do not directly address the issue of stability in finite-time. Therefore, sliding surfaces based on TSM and FTSM are designed to achieve the requirement. However, most of the literature on FTSM focuses on its implementation on fully actuated systems. Only a handful of studies have focused on its application to underactuated systems, particularly gantry cranes. To address the issue, a control system based on FTSM and HSM was presented in Chapter 5. This chapter presents the implementation of the proposed control system on gantry cranes. As mentioned earlier, the proposed controller has the advantage of robustness and faster convergence to equilibrium, which makes it suitable for cranes. To provide the effectiveness of the controller, this chapter presents the rigorous simulation of the control system, followed by the real-time implementation of a scaled gantry crane model. Furthermore, comparative studies with the TSM and SMC are also provided in this chapter.

7.2 Crane Dynamics

Crane is also an underactuated dynamic system. Diagram of crane is presented in Fig. 7.1, which consists of a cart of mass m_c , a hoisting mechanism through a rope of length l , and a payload of mass m_p . Here, it is assumed that the payload consists of a point mass and the stiffness of the hoisting rope is negligible. As shown in Fig. 7.1, the cart is driven by a DC motor via a rack and pinion mechanism.

In this system, the dynamics of crane can be described in terms of two variables, namely position of the cart (x) and the swing angle of the payload (θ) which can be considered as the actuated (q_a) and unactuated coordinates q_u of the system, respectively, i.e. $\mathbf{q} = [q_a \ q_u] = [x, \theta]^\top$. Similarly, the control input to the system is the voltage V applied to the armature of the DC motor to drive the cart. The dynamics of the crane can be represented in the form (5.4) where the terms f_1 , f_2 , b_1 , and b_2 are defined as:

$$\begin{aligned} f_1(\mathbf{q}, \dot{\mathbf{q}}) &= \frac{-m_p l^2 \left(\frac{k_T k_b}{R_a r^2} \right) \dot{x} + m_p^2 l^2 \sin \theta (l \dot{\theta} + g \cos \theta)}{(m_p + m_c)m_p l^2 - m_p^2 l^2 \cos^2 \theta}, \\ f_2(\mathbf{q}, \dot{\mathbf{q}}) &= \frac{m_p \left(\frac{k_T k_b}{R_a r} \right) l \dot{\theta} \cos \theta}{(m_p + m_c)m_p l^2 - m_p^2 l^2 \cos^2 \theta} \\ &\quad - \frac{m_p l \sin \theta \left\{ (m_p + m_c)g + m_p l \cos \theta \dot{\theta}^2 \right\}}{(m_p + m_c)m_p l^2 - m_p^2 l^2 \cos^2 \theta}, \\ b_1(\mathbf{q}) &= \frac{m_p l^2 \frac{k_T}{R_a r}}{(m_p + m_c)m_p l^2 - m_p^2 l^2 \cos^2 \theta}, \\ b_2(\mathbf{q}) &= \frac{-m_p l \frac{k_T}{R_a r} \cos \theta}{(m_p + m_c)m_p l^2 - m_p^2 l^2 \cos^2 \theta}. \end{aligned} \quad (7.1)$$

where k_T is the motor torque constant, k_b is its back emf constant, R_a is the armature resistance, and r is the radius of the shaft attached to the motor.

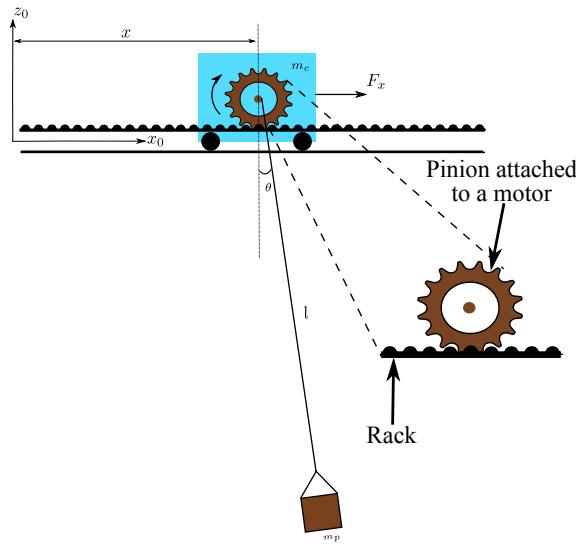


Figure 7.1 : Diagram of a gantry crane.

7.3 Control System

Since crane is an underactuated system one can apply the control law (5.14) that was presented in Chapter 5. The control law can be summarized as

$$u = -\frac{1}{k_1 b_1(\mathbf{q}) + k_2 b_2(\mathbf{q})} \{u_1 + u_2 + u_3\}, \quad (7.2)$$

where

$$\begin{aligned} u_1 &= k_1 \left(f_1(\mathbf{q}, \dot{\mathbf{q}}) + \beta_1 \frac{q_1}{p_1} e_1^{q_1/p_1-1} \dot{e}_1 + \alpha_1 e_1 - \ddot{q}_{ad} \right) \\ u_2 &= k_2 \left(f_2(\mathbf{q}, \dot{\mathbf{q}}) + \beta_2 \frac{q_2}{p_2} e_2^{q_2/p_2-1} \dot{e}_2 + \alpha_2 e_2 - \ddot{q}_{ud} \right) \\ u_3 &= \eta \text{sgn}(S) + KS, \end{aligned}$$

and where the control gains K and η are chosen such that $K > 0$ and $\eta > D_m = \sup_{t \geq 0} |k_1 \sigma_1(\mathbf{q}) + k_2 \sigma_2(\mathbf{q})|$. Here, it should be noted that e_1 is defined as $e_1 = x - x_d$ where x_d is the reference cart position. Furthermore, the objective of the control system is to suppress the swinging load. Therefore, e_2 is defined as $e_2 = \theta$.

It should be noted that in order to avoid the singularity in (5.14),

$$k_1 b_1(\mathbf{q}) + k_2 b_2(\mathbf{q}) \neq 0 \quad \forall \mathbf{q} \in \mathbb{R}^2. \quad (7.3)$$

should be satisfied. By substitution, $b_1(\mathbf{q})$ and $b_2(\mathbf{q})$ from (7.1) one obtains:

$$k_1 l - k_2 \cos(\theta) \neq 0. \quad (7.4)$$

Let us consider the plot of $k_1 l$ and $k_2 \cos(\theta)$ (Fig. 7.2). From the figure it is clear that for all k_1 and k_2 , if $l k_1 > \cos(\theta)$, the singularity is avoided. In other words,

$$\frac{k_1}{k_2} > l \quad (7.5)$$

should be satisfied to avoid the singularities in Equation (7.2).

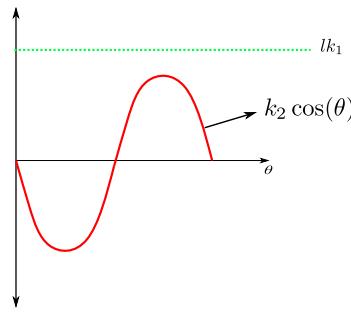


Figure 7.2 : Plot of lk_1 and $k_2 \cos(\theta)$

7.4 Simulation results

The laboratory testbed is shown in Fig. 7.3, whose system parameters, along with the parameters of the controller used in the simulation and experiments are provided in the Table 7.1.

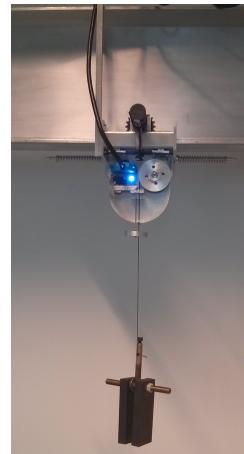


Figure 7.3 : Gantry crane laboratory testbed.

7.4.1 Gain Tuning using Genetic Algorithm

As mentioned earlier the parameters q_1 , p_1 , q_2 , and p_2 should be odd integers such that $q_1 < p_1$ and $q_2 < p_2$. Therefore, the parameters were selected randomly so that it satisfies above mentioned conditions. However, to tune the parameters k_1 , k_2 , β_1 ,

β_2 , α_1 , and α_2 we applied a Genetic Algorithm (GA) in the simulation environment. The GA is a search optimization technique based on natural selection. It starts with the initial population of chromosomes where each individual represents a solution to a problem. Performance of each chromosomes is evaluated using a fitness function. The algorithm has three main processes, namely Selection, Crossover and Mutation, which are evaluated in each generation. Finally, the algorithm terminates if the fitness value of the individuals has reached a user-defined threshold, or the maximum number of generations has reached. Details on the algorithm can be found in Whitley (1994).

Here the tuning process was conducted by using Matlab, and its flowchart is presented in Fig. 7.4. During the process, contents of each chromosomes of GA represents gain parameters of the controller which are randomly initialized. The performance of the chromosomes are evaluated by simulating the system, followed by calculation of the fitness function. We considered the Integral of Time Square Error (ITSE) as the fitness function, which is defined as:

$$\text{ITSE} = 0.5 \int_0^{t_f} t e_x^2 dt + 0.5 \int_0^{t_f} t \theta^2 dt \quad (7.6)$$

where $e_x = x - x_d$ and t_f is the final time. After evaluation of the cost function, the selection, mutation and crossover operations of GA are applied. Finally, this whole process is repeated until a termination condition is reached. The gains after this tuning process are presented in Table. 7.1

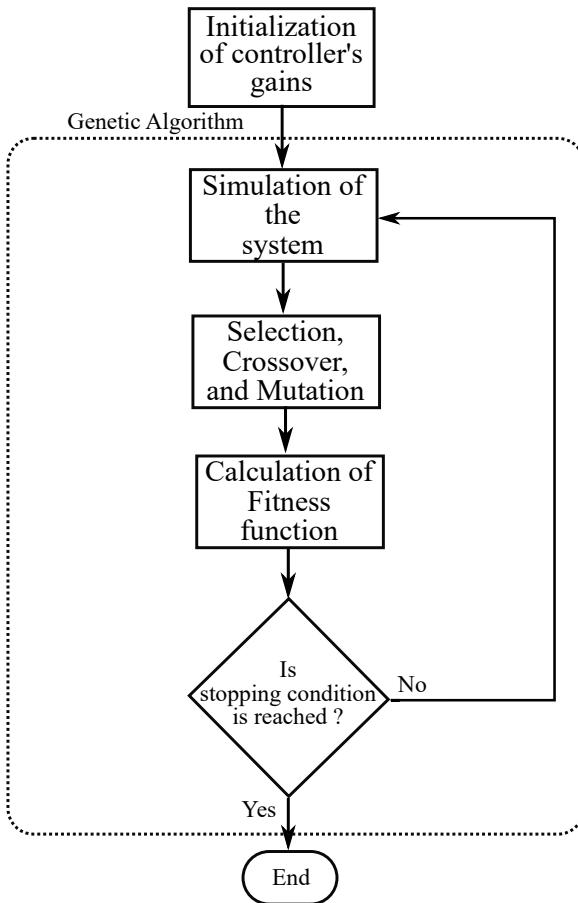


Figure 7.4 : Flowchart for tuning of controller gains using Genetic Algorithm

The control parameter η is related to the convergence rate of the sliding mode, and its value should be greater than upperbounds of disturbances, i.e. μ_1 and μ_2 . Nevertheless, higher values of the η lead to the increment in chattering magnitude, whereas lower values decrease the robustness of the system. Therefore, the value of η is manually adjusted during experiment.

7.4.2 Control performance

Figure 7.5 and 7.6 show simulation results of the position of the cart, the swing angle of the payload, and their velocities, respectively, for a cart position reference of 1.5 m. The cart reaches its desired position in about 3.5 sec with a maximum velocity

System parameter	Value
m_p	2.24 kg
m_c	2.7 kg
l	1 m
g	9.8065 m/s ²
μ_1	0.1 m/s ²
μ_2	0.1 rad/s ²
Controller parameter	$\alpha_1 = 1.75, \alpha_2 = 0.01, \eta = 0.5,$
	$\beta_1 = 0.8, \beta_2 = 0.4,$
	$p_1 = 11, q_1 = 9, p_2 = 9, q_2 = 7$
	$k_1 = 6.8, k_2 = 5.4, K = 1$

Table 7.1 : System and control parameters

of 1 m/s. Furthermore, the swing of the payload diminishes at about the same-time with the peak oscillation of about 0.3 rad (17.17 degrees). The angular velocity of the payload also dies out in approximately 3 seconds. The sliding functions and the control signal are depicted in Fig. 7.7 (a) and (b), respectively. Both first layer and second layer sliding functions, i.e. s_1 , s_2 , and S are asymptotically convergent towards the equilibrium, $s_1 = 0$, $s_2 = 0$, and $S = 0$. Similarly, the control input generated by the proposed technique is rather smooth with a maximum value of 8 V.

In order to test the robustness performance of the system under uncertainties, the system parameters, l and m_p , are changed from their nominal values given in Table 7.1. Figure 7.9(a) shows the responses of the cart position (x) and swing angle (θ) when the length of the rope varies by 20%. From the figure, it can be observed that there is a small variation in the system response. For instance, the cart reaches

its desired position in 3.5 sec, except when the length is 0.8 m. Similarly, swing of the payload dies out in less than 5 sec for all the cases. However, oscillation increases slightly and reaches the maximum value of 0.5 rad (28.64 degrees) when the rope length is reduced to 0.8 m from the nominal value.

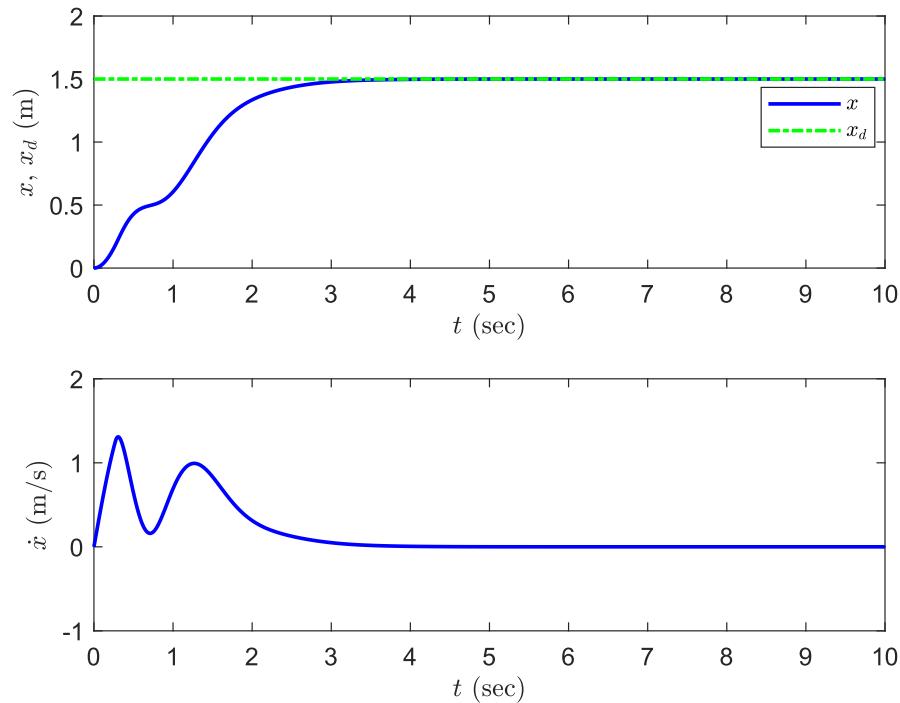


Figure 7.5 : Response of the trolley position.

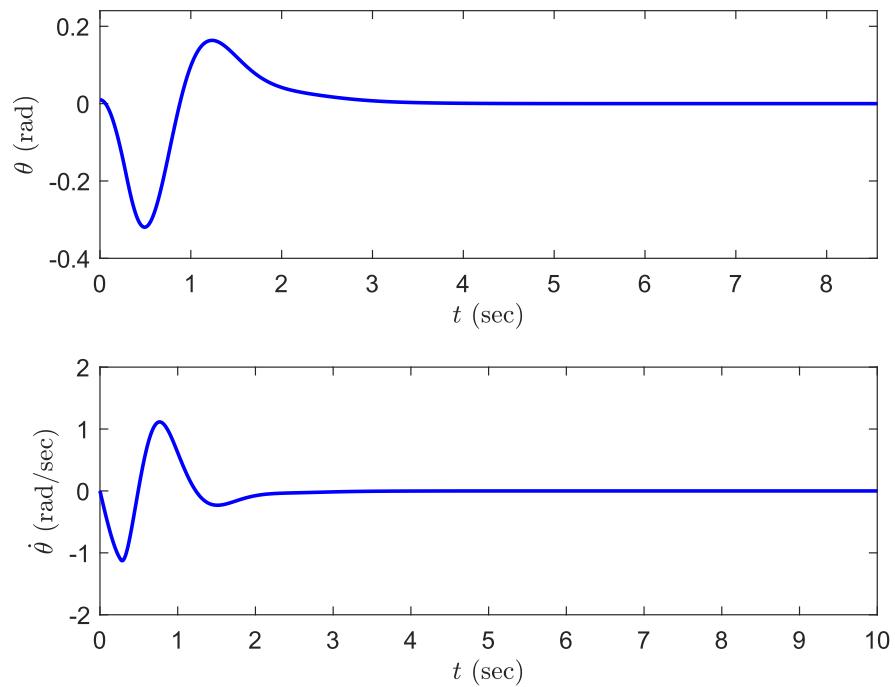
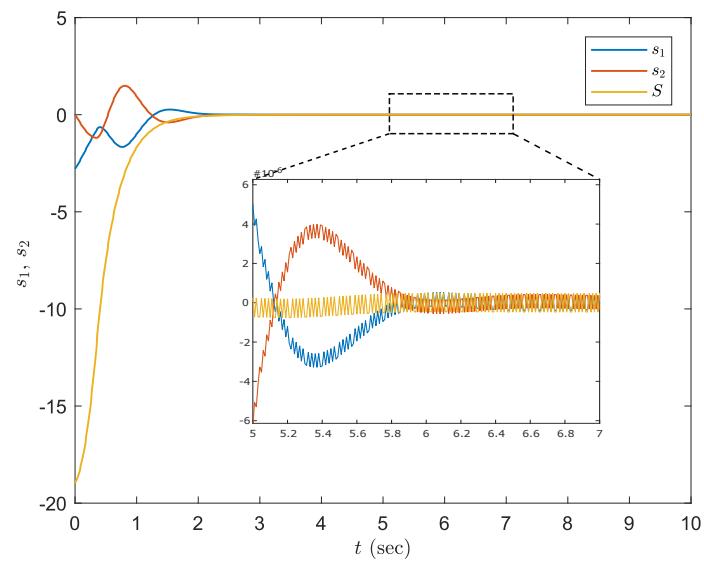


Figure 7.6 : Responses of the payload swing angle.

Figure 7.7 : Responses of sliding functions s_1 , s_2 and S .

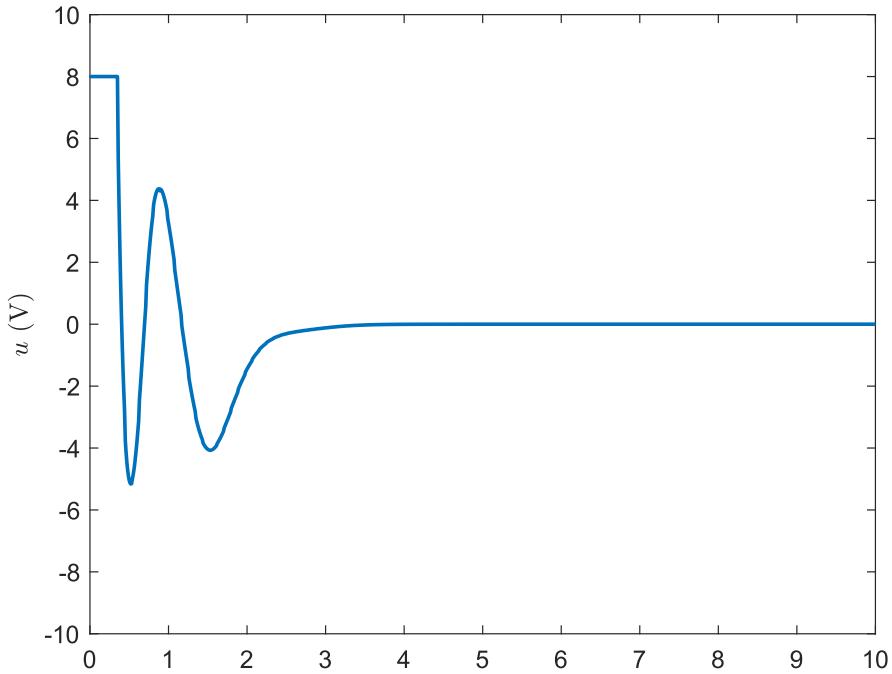


Figure 7.8 : Responses the control input u .

Furthermore, the system shows little variation in performance with respect to the change in the mass of the payload. There are unnoticeable changes in trajectories of cart, and payload shows slight increment in peak oscillation when its mass is reduced to 1.2 kg (Fig. 7.9 (b)). These results clearly indicate that the proposed control law is immune to the parametric uncertainties. The system was also evaluated under intrinsic disturbances by adding the backlash or deadband into the input. The backlash nonlinearity considered during the simulation is given by Selmic and Lewis (2000):

$$D(u) = \begin{cases} \gamma_1 u & u \geq \Delta \\ 0 & |u| < \Delta \\ \gamma_2 u & u \leq -\Delta \end{cases} \quad (7.7)$$

where γ_1 and γ_2 are the slope and Δ is the deadband.

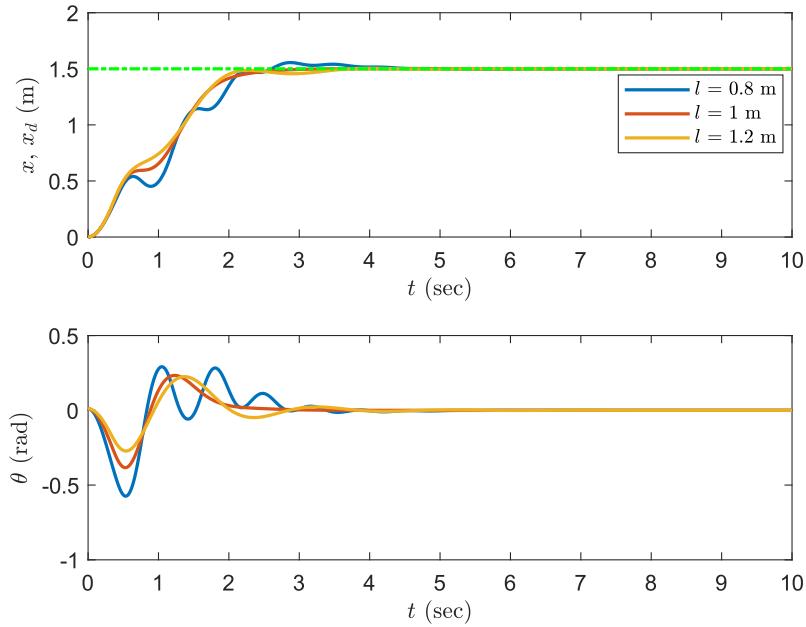


Figure 7.9 : Performance of the controller with respect to the variations in rope length.

Figure 7.11 shows the response of the system for the backlash with deadband (Δ) of 0.5, 1.0, and 1.5 V, respectively, and $\gamma_1 = \gamma_2 = 1$. From the figure it can be observed that the transient response is similar to that of the previous cases, but there are some peak residual oscillations of 0.005 rad (0.25 degree) in the steady state condition, which can be observed in the zoomed section. Furthermore, the amplitude of the oscillation decreases with decreasing backlash voltage. Nevertheless, the residual swing is quite small. Therefore, it can be concluded that the system shows good performance under disturbances.

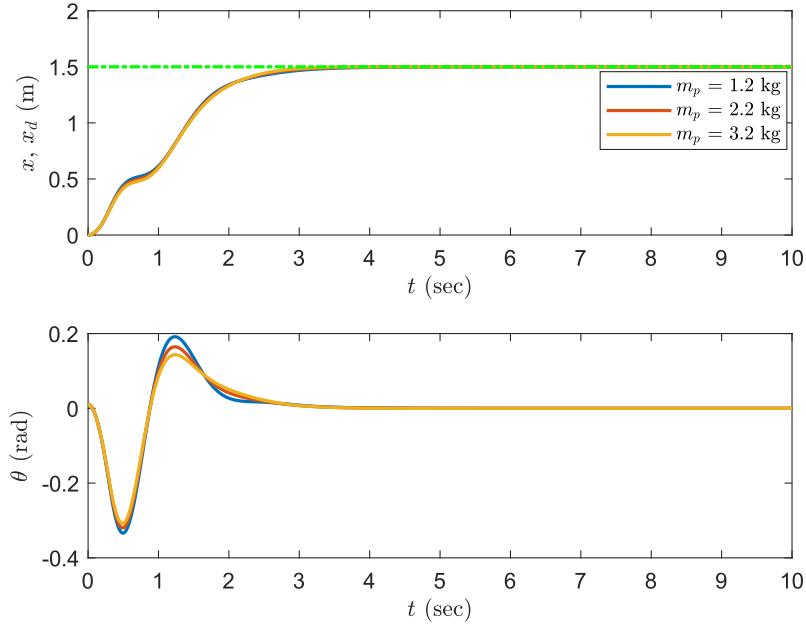


Figure 7.10 : Performance of the proposed control system with respect to varying payload mass.

In Remark 2, it was concluded that s_1 and s_2 are asymptotically stable. However, in practice there always exist system nonlinearities in crane dynamics, such as mechanical hysteresis and backlash, which may affect the system performance. This is clear from the simulation result obtained by adding backlash nonlinearity, i.e. Fig. 7.11.

As we mentioned earlier in Section 2.1, the disturbances such as backlash are modelled as σ_1 and σ_2 in the crane dynamics (5.4), which are assumed them to be bounded i.e. $\sigma_1 < \mu_1$ and $\sigma_2 < \mu_2$. However, in real systems, μ_1 and μ_2 are difficult to quantify because they are affected by various factors, such as unmodelled dynamics, friction, and so on. One of the solutions to the problem is to increase η , but as mentioned earlier in Remark 2, it exacerbates the chattering as well. In this paper this parameter is adjusted manually so that the residual oscillation is minimum. This

behaviour will also be discussed later while presenting the experimental results.

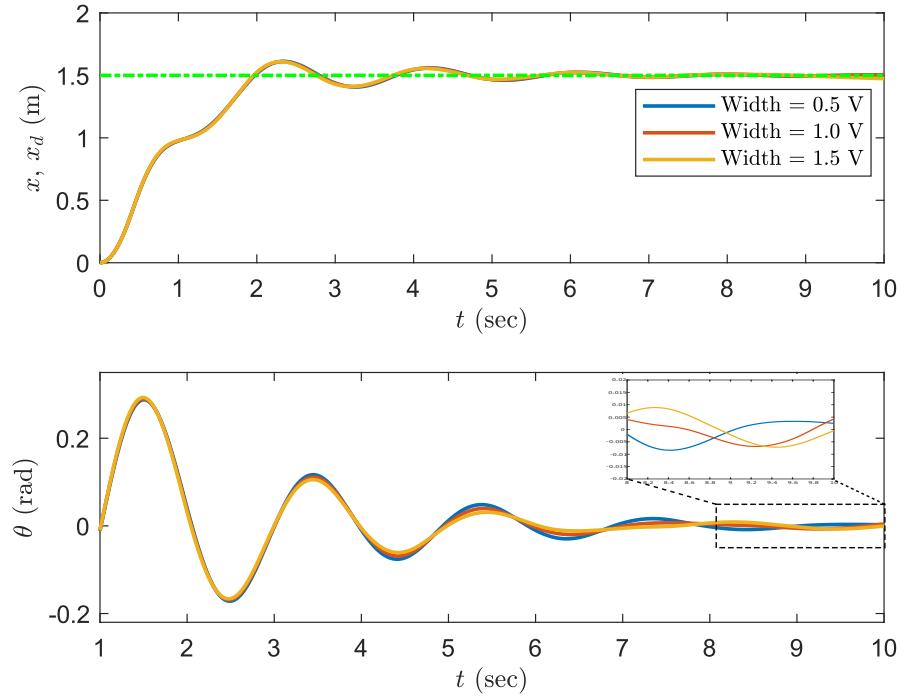


Figure 7.11 : Performance of the controller for different deadband.

7.4.3 Comparison with SMC and TSM

Now to show the advantage of the proposed approach, controllers designed from the Terminal Sliding Mode (TSM) and Sliding Mode Control (SMC) methods are compared with the FTSM controller within the same hierarchical sliding mode framework. The first layer sliding functions for TSM and SMC are defined respectively as in equations (7.8) and (7.9) below,

$$\begin{aligned} s_1 &= \dot{e}_1 + \beta_1 e_1^{\nu_1/\rho_1} \\ s_2 &= \dot{e}_1 + \beta_2 e_2^{\nu_2/\rho_2}, \end{aligned} \tag{7.8}$$

and

$$\begin{aligned} s_1 &= \dot{e}_1 + \alpha_1 e_1 \\ s_2 &= \dot{e}_2 + \alpha_2 e_2. \end{aligned} \tag{7.9}$$

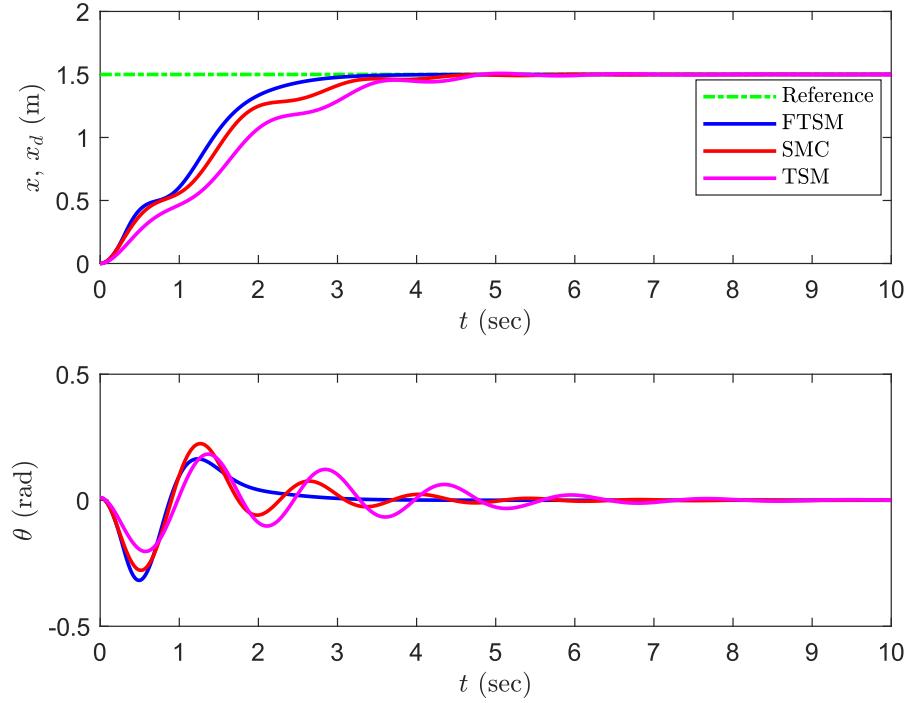


Figure 7.12 : Comparison of FTSM, TSM, and SMC responses for the cart position x , and payload swing angle θ .

The overall control inputs u for both controllers are similar to (7.2) but u_1 , u_2 and u_3 are defined as

$$\begin{aligned} u_1 &= k_1 \left(f_1(\mathbf{q}, \dot{\mathbf{q}}) + \beta_1 e_1^{\nu_1/\rho_1-1} \dot{e}_1 - \ddot{q}_{ad} \right) \\ u_2 &= k_2 \left(f_2(\mathbf{q}, \dot{\mathbf{q}}) + \beta_2 e_2^{\nu_2/\rho_2-1} \dot{e}_2 \right) \\ u_3 &= \eta \text{sgn}(S) + KS, \end{aligned} \quad (7.10)$$

for TSM, while for SMC they are defined as

$$\begin{aligned} u_1 &= k_1 (f_1(\mathbf{q}, \dot{\mathbf{q}}) + \alpha_1 \dot{e}_1 - \ddot{q}_{ad}) \\ u_2 &= k_2 (f_2(\mathbf{q}, \dot{\mathbf{q}}) + \alpha_2 \dot{e}_2) \\ u_3 &= \eta \text{sgn}(S) + KS, \end{aligned} \quad (7.11)$$

where $q_{ad}(t)$ is the desired trajectory of the trolley.

Controller	Parameters
TSM	$\beta_1 = 0.7, \nu_1 = 9, \rho_1 = 11, k_1 = 9.5$
	$\beta_2 = 0.03, \nu_2 = 7, \rho_2 = 9, k_2 = 4.4$
SMC	$\alpha_1 = 1.45, \alpha_2 = 0.05, k_1 = 4.2, k_2 = 2.9$
	$\eta = 0.5, K = 1$

Table 7.2 : Controller parameters for TSM and SMC

The TSM controller parameters $\nu_1, \nu_2, \rho_1, \rho_2$ should be odd integers such that $\nu_1 < \rho_1$ and $\nu_2 < \rho_2$. During the experiment, we used the same parameters that was applied in the proposed controller. However, to tune parameters $k_1, k_2, \alpha_1, \alpha_2, \beta_1$ and β_2 , we applied the Genetic Algorithm as described in Section 4.1. The tuned parameters are presented in Table 7.2. For η , we used the same value as in the proposed controller case.

Results of the comparison for the cart position (x) and the swing angle (θ) are presented in Fig. 7.12. The responses obtained clearly confirm the improved performance of the proposed controller regarding the settling time. For instance, the cart takes about 3.5 seconds to reach the reference position, compared to 4.5 seconds for TSM and about 5 seconds for SMC, respectively. Furthermore, swing angle of the payload reaches the equilibrium, i.e. $\theta = 0$, in about 3 seconds with smooth trajectory, compared to SMC and TSM, which takes around 4 and 6 seconds, respectively. These results indicates the advantage of the proposed method over the other methods.

7.5 Experimental results

7.5.1 Control implementation

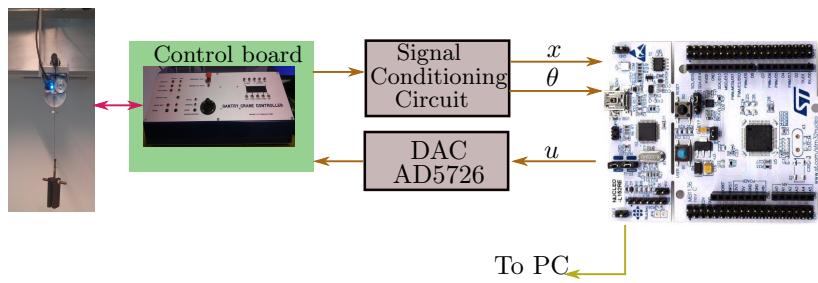


Figure 7.13 : Block diagram for implementation.

The set-up is depicted in Fig. 7.13 which consist of the overhead crane, as shown in Fig. 7.3, a control board, signal conditioning circuits, and a Stm32f411 based microcontroller board. The crane consists of potentiometers for the measurement of the cart's position (x) and the swing angle of the payload (θ), which can be described by the following equations:

$$\begin{aligned} y_\theta &= \varsigma_1 \theta + \varsigma_2 \\ y_x &= \hat{\varsigma}_1 x + \hat{\varsigma}_2, \end{aligned} \tag{7.12}$$

where y_θ and y_x are the measured quantities, ς_1 and $\hat{\varsigma}_1$ are the gains, and ς_2 and $\hat{\varsigma}_2$ are the offset in measurements. The parameters ς_1 , $\hat{\varsigma}_1$, ς_2 and $\hat{\varsigma}_2$ are estimated by applying least-square curve fitting on the measured input and output data. In addition, the sensors are connected to the control board which provides the electrical interface. Similarly, the proposed control algorithm is implemented in the microcontroller which consists of a ARM Cortex M4 processor with 512 KB of flash memory and 2 KB of RAM. It receives the feedback signals, i.e. x and θ from the control board through a signal conditioning circuit, whose purpose is for an interface between the crane and the microprocessor. The control signal generated by the board is fed to crane through an AD5726 DAC IC. To collect crane sensors and control data for post-processing, a PC with the microprocessor board is interfaced via a UART protocol.

The control input (7.2) requires both position and the velocity of the feedback

signals. Nevertheless, the velocities are not available from the system. Therefore, the signals are estimated using an approximate differentiator, which is defined as $\dot{\mathbf{q}} = \frac{1}{T} (\mathbf{q}(t) - \mathbf{q}(t - T))$, where T is the sampling period of the controller.

The Nucleo board runs FreeRTOS operating system (OS) and the proposed controller runs on the board as an OS's stack. The sampling time for the control system is 1 ms. Digital low pass filters are used to reduce the noise in the measurement.

7.5.2 Test results

In the experiments, the system and controller parameters are used as listed in Table 7.1. The experimental response of the cart position x is shown in Fig. 7.14 with the desired position in 5 seconds. Similarly, the velocity of the cart \dot{x} achieves the maximum value 0.75 m/s and reaches the equilibrium eventually.

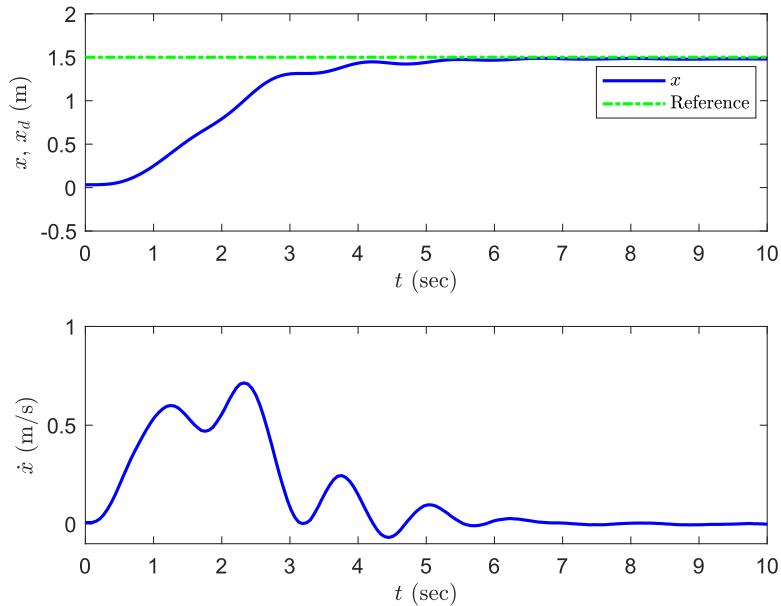


Figure 7.14 : Test results of the proposed FTSM based control scheme for cart position.

Figure 7.15 shows the response of swing angle of payload. From the figure it is

clear that the proposed controller is able to suppress the payload swing (θ) and its angular velocity ($\dot{\theta}$). The maximum angle of the payload motion is about 0.15 rad or 8.52 degrees. Furthermore, the system shows much lower residual oscillation of around 0.25 degrees. As mentioned earlier in Section 4.2, the residual oscillation is due to the disturbances such as backlash. To verify this, a comparison of the simulation results and experimental results is presented in Fig. 7.16 . The figure clearly indicates that the magnitude of residual oscillations, as observed in the zoomed section, are in the same range. Furthermore, the figure also shows that the response of the system is similar to that of simulation. For instance, the settling time of the cart for both cases is around 5 seconds. There is, however, some differences in the transient response of simulation and the real-time experimental results, which is due to unmodelled nonlinearities.

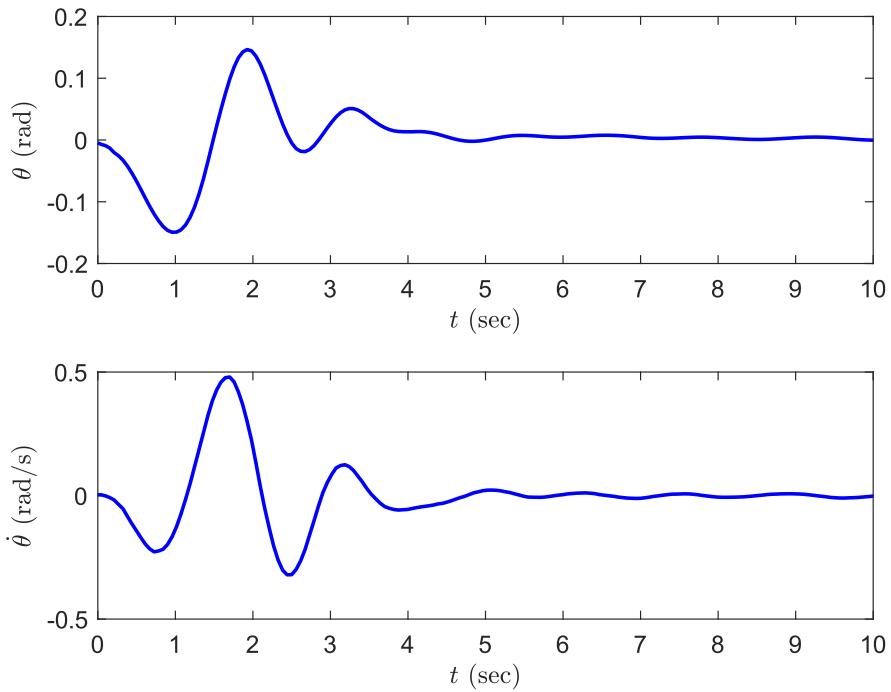


Figure 7.15 : Test results of the proposed method for swing angle.

Overall, the results clearly indicate that the proposed control method can stabilise

the system and outperform the conventional SMC and TSM in terms of its fast response. Feasibility of the proposed control scheme is also verified as shown in Fig. 7.17 for the responses of the control signal and sliding functions.

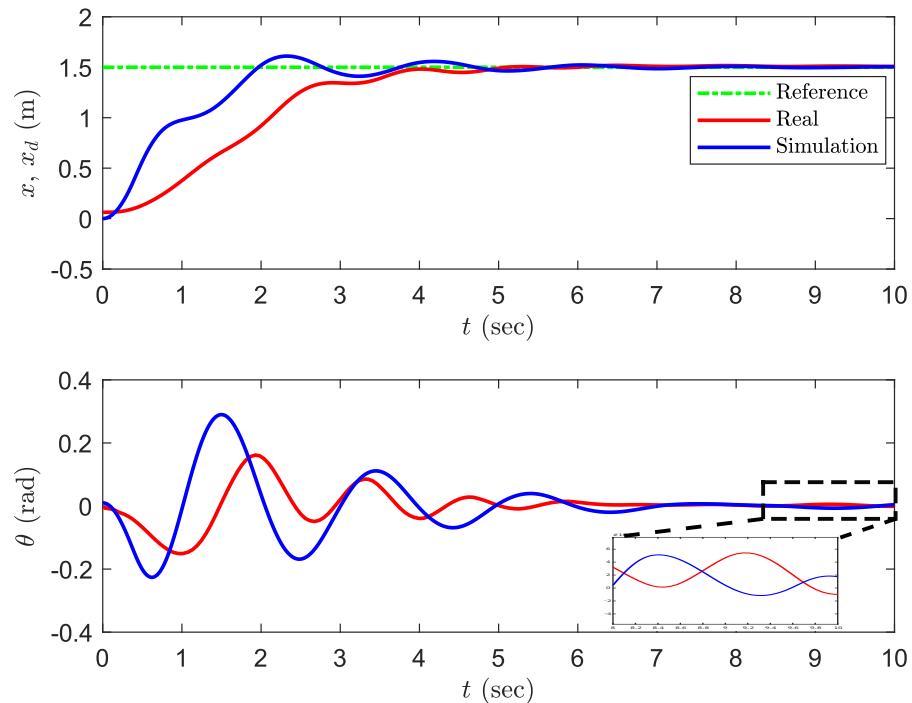


Figure 7.16 : Comparison between simulation and experiment.

It can be noticed that in Fig 7.16 the transient response of control input is not smooth. One of the reasons for such behaviour of the system is due to the unmodeled nonlinearities such as Equation (5.2) which are dependent on velocity.

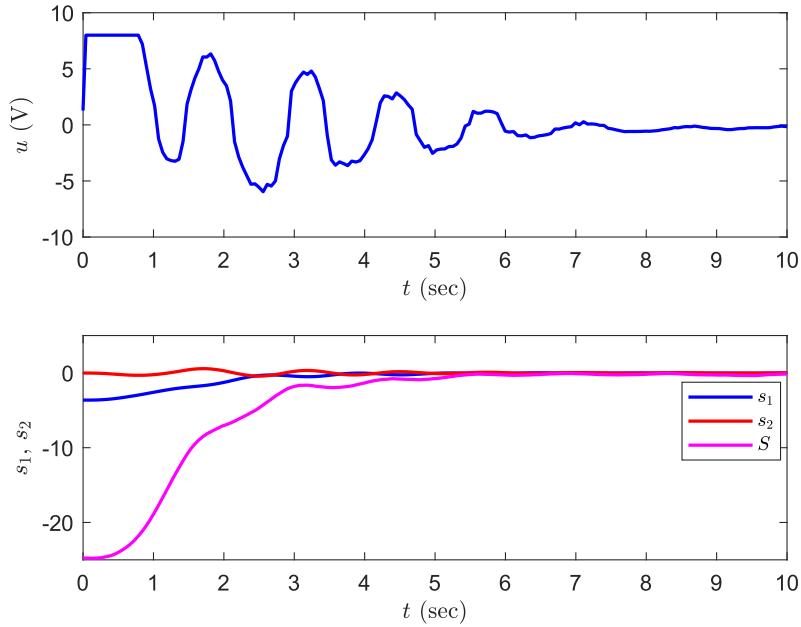


Figure 7.17 : Feasibility and performance of the proposed control algorithm.

To judge robustness of the proposed control method, we created scenarios similar to those of simulation. For which, we tested the crane performance by varying the length of the rope by 20 % from the nominal value, i.e. 1 m. Responses of the cart position and the swing angle are shown in Fig. 7.18(a). The experimental results clearly indicate the insensitivity of the proposed control system with respect to parameter variations. The results observed further coincide with those obtained from simulation.

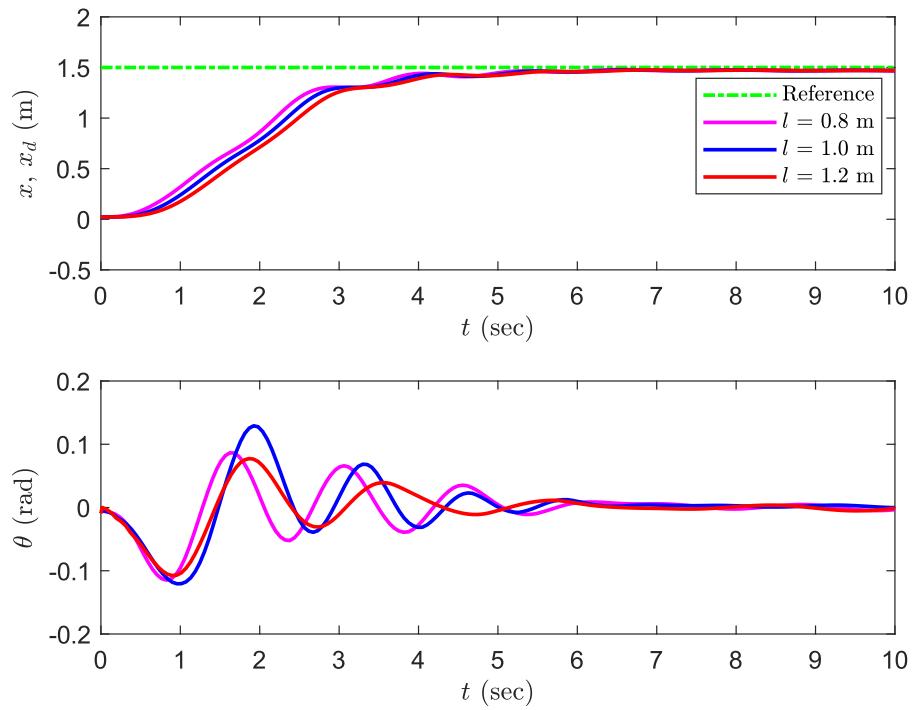


Figure 7.18 : Experimental responses of the cart position and swing angle against parameter variations.

For comparison with TSM and SMC, the control laws as mentioned in Section 4.2 with the same design parameters. The time responses of the crane's cart position and payload angular swing are depicted in Fig. 7.19. Results of the experiment clearly show the advantage of our method regarding the convergence time. For instance, the cart takes about 5.0 sec to reach the goal location in case of the proposed FTSM as compared to 5.5 sec for TSM, and 6.0 sec for SMC. The observations further verify the results obtained from simulation, as described in the previous section.

Parameters	Value
a	0.2857 m/sec^2
t_1	1.5 sec
t_2	3.5 sec
t_F	5 sec
T_s	$1 \times 10^{-3} \text{ sec}$

Table 7.3 : Reference trajectory parameters

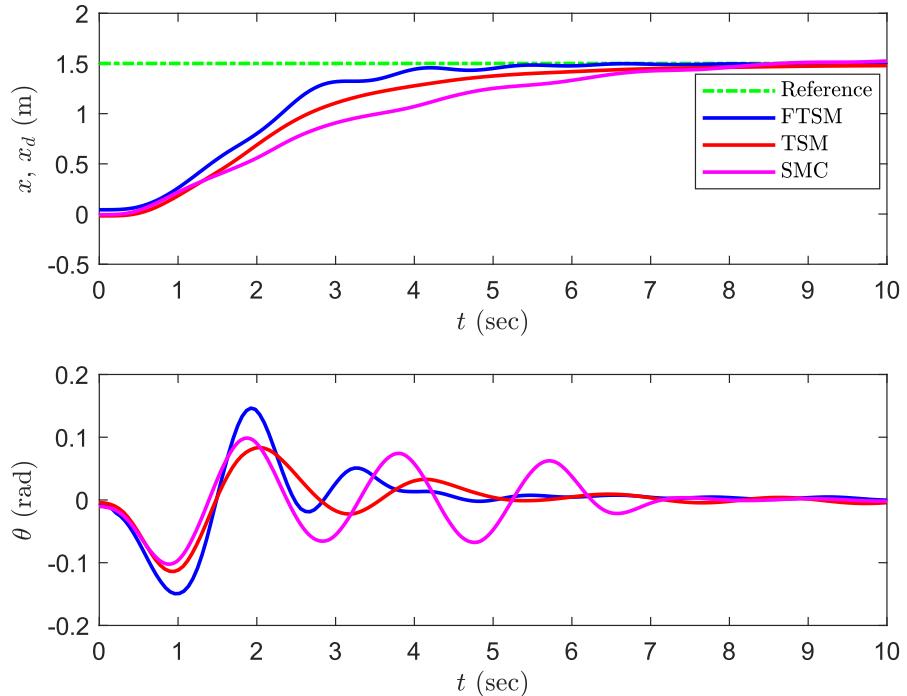


Figure 7.19 : Comparison with TSM and SMC.

7.5.3 Trajectory tracking

To further experimentally evaluate the control performance, we apply a trapezoidal reference for the cart velocity. The trajectory consists of three phases, namely, acceleration, constant velocity and deceleration. The desired velocity (\dot{x}_d) and po-

sition x_d using the trapezoidal trajectory for the cart velocity can be described as:

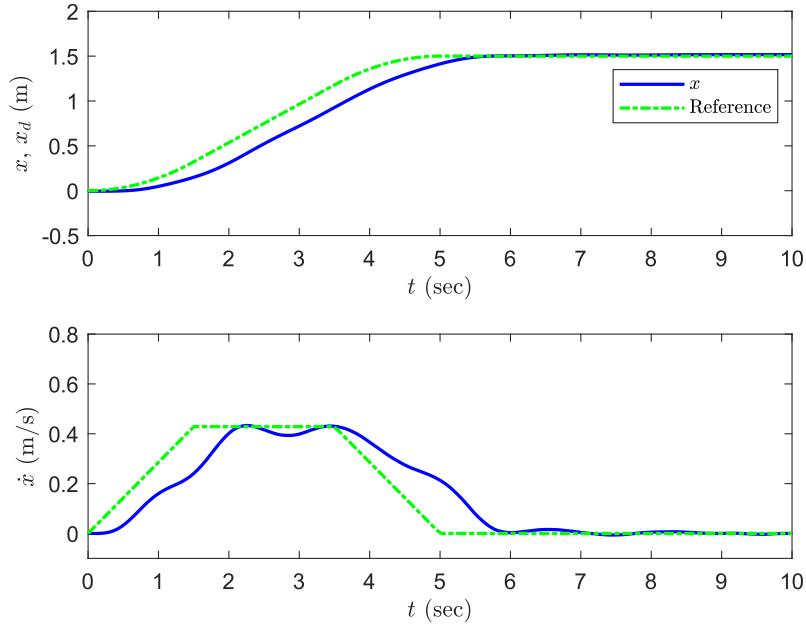


Figure 7.20 : Position and velocity trajectories of the cart.

$$\dot{x}_d = \begin{cases} at & 0 < t \leq t_1 \\ at_1 & t_1 < t \leq t_2 \\ at_1 - a(t - t_2) & t_2 < t \leq t_F, \end{cases} \quad (7.13)$$

and

$$x_d(t) = \begin{cases} a\frac{t^2}{2} & 0 < t \leq t_1 \\ a\frac{t_1^2}{2} + at_1(t - t_1) & t_1 \leq t \leq t_2 \\ a\frac{t_1^2}{2} + at_1(t_2 - t_1) - \frac{a}{2}(t^2 - t_2^2) & t_2 < t \leq t_F. \end{cases} \quad (7.14)$$

where a is the acceleration, t_1 is the time instant indicating the end of the acceleration phase, t_2 is the time instant at the beginning of the deceleration phase, and t_F is the final time period. The parameters of the reference trajectory for the experiment

are shown in Table 2. The sampling period (T_s) of the controller is 1 ms, much smaller than the motor time constant.

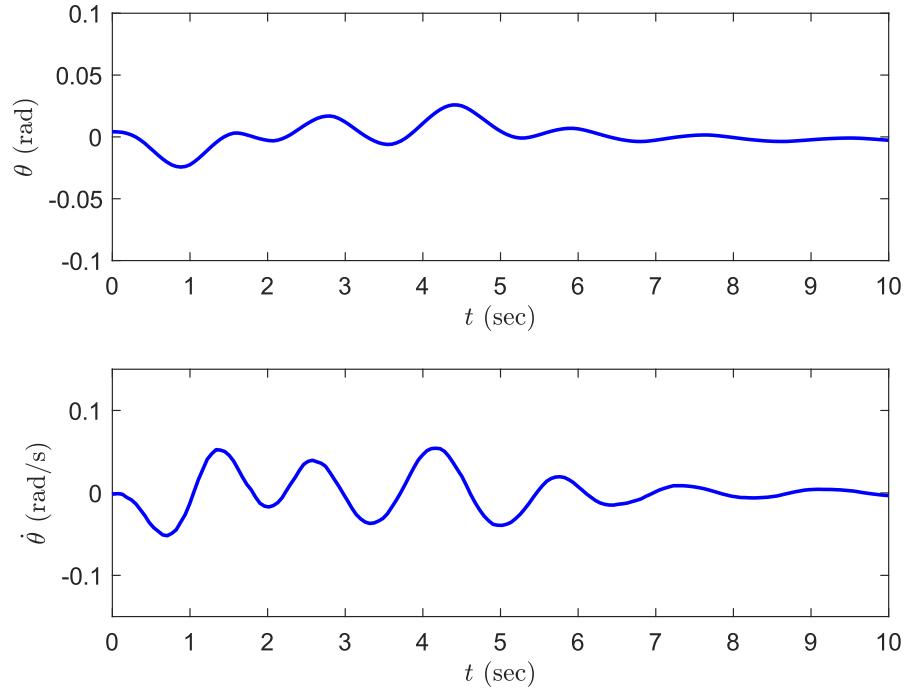


Figure 7.21 : Position and velocity trajectories of the swinging load.

Tracking performance of the proposed controller is presented in the Fig. 7.20, wherein it is clear that the controller is able to track the reference position and reach the final desired location in about 5 sec. Payload swing angle and its velocity, as shown in the Fig. 7.21 , are less than 3 degrees and 30 (degrees/sec), respectively. To indicate feasibility of the proposed FTSM controller, Fig. 7.22 depicts the sliding surfaces and control input of the controller. The control voltage signal during tracking the trapezoidal velocity reference is between -2 and 4 V.

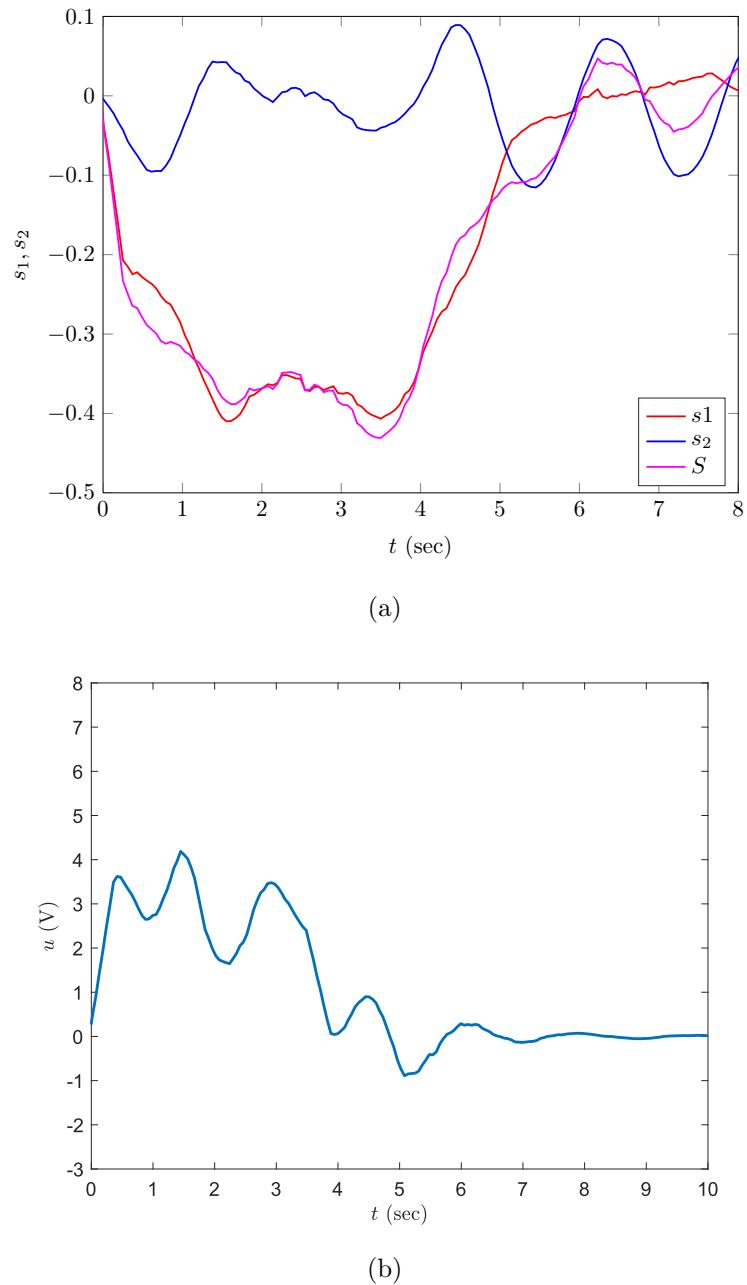


Figure 7.22 : Sliding functions and control input during tracking.

Robustness of the controller is again tested by varying the rope length by 20 % from its nominal value, as can be seen in Fig. 7.23 and 7.24, wherein the control performance of the position and velocity trajectories of the cart as well as the swinging load appears not to be affected by the length variations under the proposed controller.

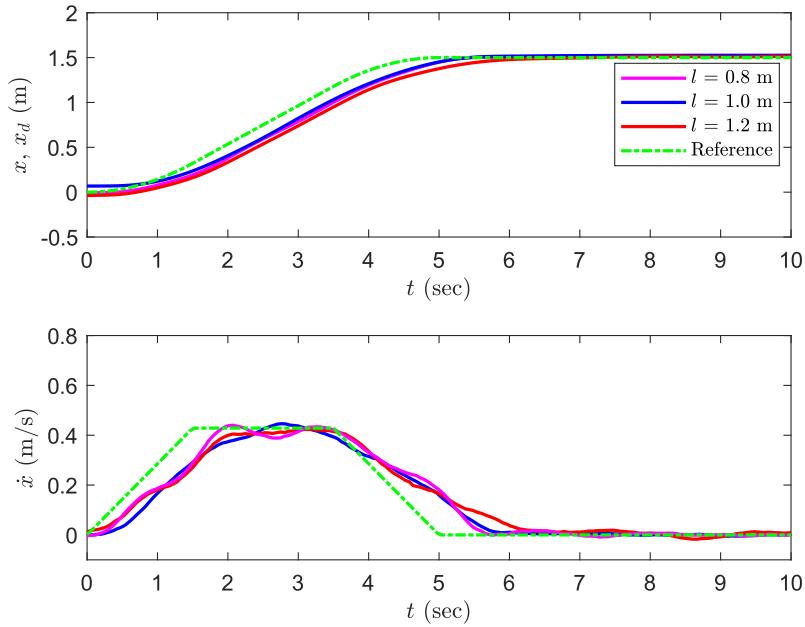


Figure 7.23 : Control performance of cart position with respect to varying rope length.

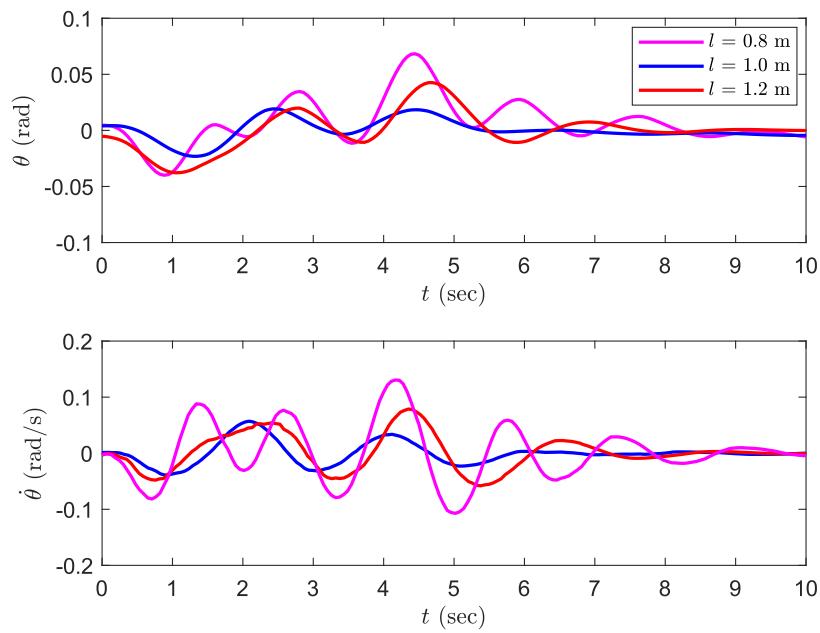


Figure 7.24 : Control performance of payload swing angle due to varying rope length.

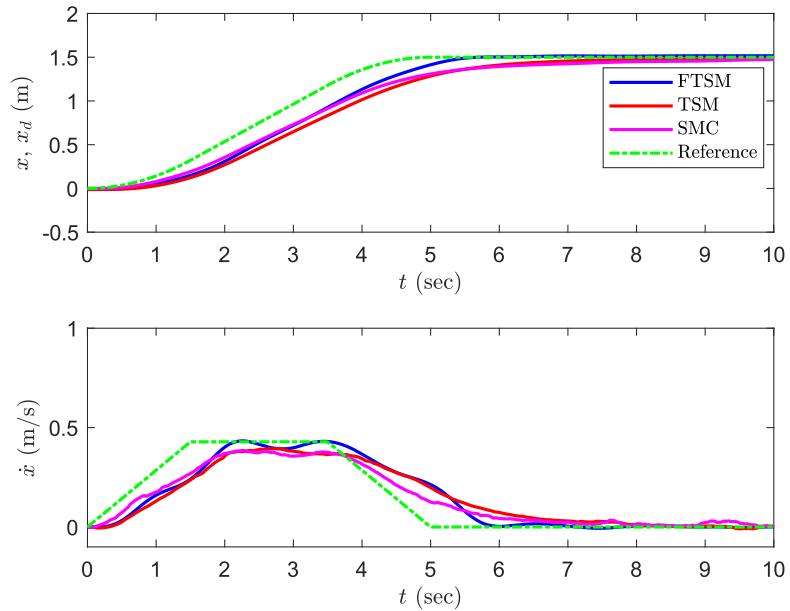


Figure 7.25 : Response of the cart for the proposed FTSM controller, TSM, and SMC.

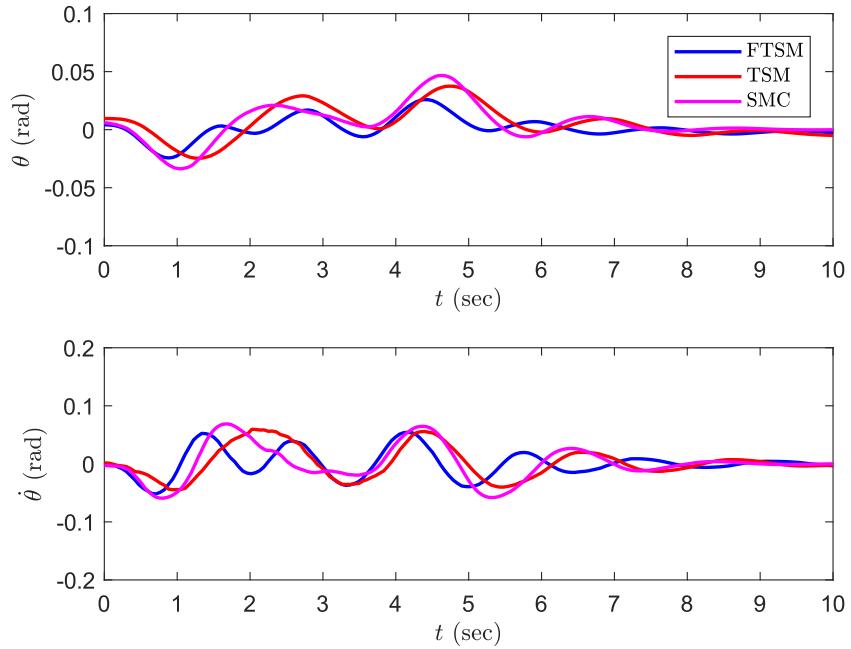
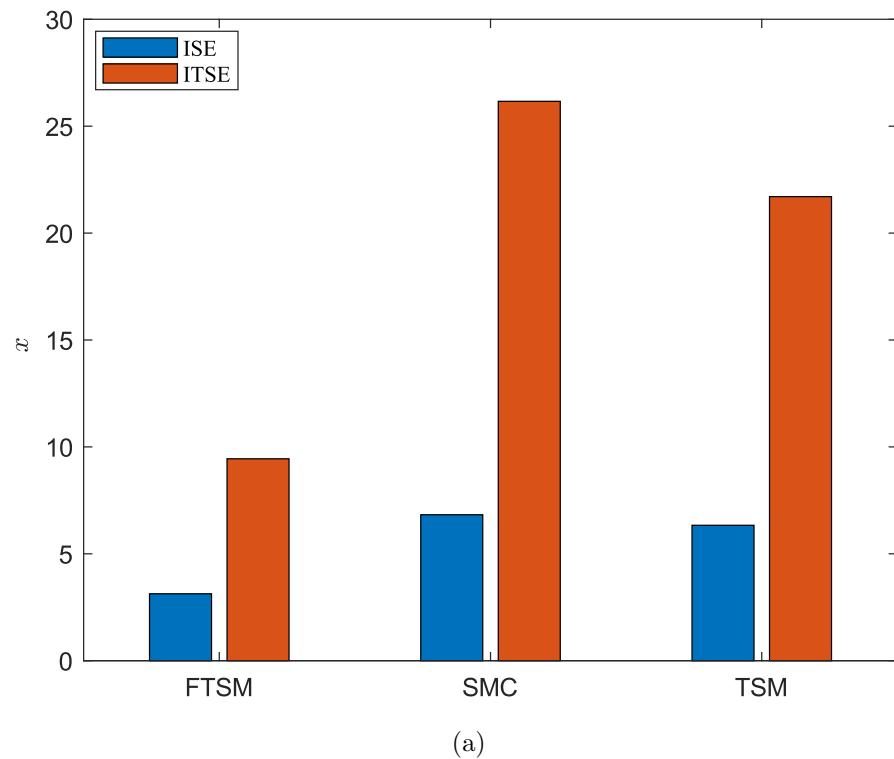
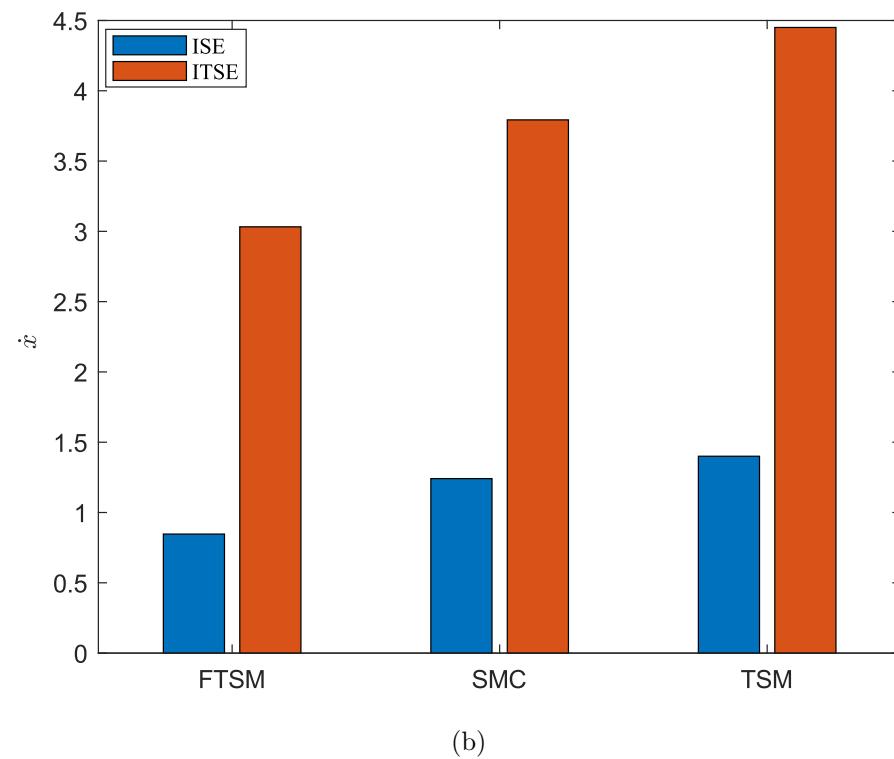


Figure 7.26 : Comparison of the swing angle for the proposed FTSM, TSM, and SMC.



(a)



(b)

Figure 7.27 : ISE and ITSE errors for a) cart's position and b) velocity tracking.

The merit of the proposed controller is also confirmed by comparison with the Terminal Sliding Mode(TSM) and Sliding Mode Control (SMC). As shown in Fig. 7.25 and 7.26, the cart's position with FTSM is the closest to the reference trajectory. The experimental results can also be evaluated in terms of performance indices such as the Integral of Squared Error (ISE) and Integral of Time Squared Error (ITSE). The results are summarised in the Fig. 7.27, where it can be seen that the tracking performance of the cart's position (x) with FTSM is much better than with TSM or SMC with respect to these indices.

7.6 Summary

The implementation of the proposed FTSM-based controller on underactuated gantry cranes has been presented in this chapter, along with the extensive simulation and experimental results. The results clearly indicate the advantage of the proposed method. Indeed, a comparison with SMC and TSM confirms that the proposed FSTM can provide faster convergence to the equilibrium while maintaining high performance against parameter variations and changing loads. Extensive tests of the proposed control on a scaled overhead gantry crane model have shown the high performance. Furthermore the results are in coincident with simulation results.

Chapter 8

Conclusion and Future Work

8.1 Thesis Contribution

This section highlights the key contributions of this thesis. They are enumerated as follows:

- The development of LS-SVR based identification method presented in Chapter 3. In the proposed method, the orthogonal projections of the block matrices are evaluated by using a multi-output LS-SVR regression has been formulated to be solved for by applying KKT conditions. Performance evaluation of the method has highlighted its advantage first through a numerically generated state space system and comparison with the standard identification methods. Furthermore, the evaluation of the proposed method in identification of the motion model of the target described in Chapter 6 revealed its outperformance compared to other methods.
- The development of a FTSM-based robust control method for second-order UMS. The proposed control scheme consists of hierarchical sliding surfaces that are obtained using the HSM framework. The lower layer surfaces are designed using FTSM manifolds, whereas, in the higher layer, those sliding functions are linearly combined. Stability of the system has been analyzed using the Lyapunov stability theory. Implementation results of the method on gantry cranes presented in Chapter 7 demonstrates its feasibility. Moreover, comprehensive comparison with other methods in both simulation and experiment has highlighted its merit in term of robustness against disturbances and

parametric uncertainties while having a faster response.

- The development of a FTSM-based controller for a mirror-based pointing systems which can have many practical applications. To design the controller, a decoupled discretized model of the device is first developed. Then, the controller is synthesized in discrete-time. Through extensive simulations, effectiveness of the proposed control system has been verified.

8.2 Future work

Based on the work done for this thesis, the following research directions have been envisioned for the future:

- The LS-SVR based identification algorithm is currently applied for offline processing or batch processing. However, some practical systems may require processing in real-time. Therefore, in the future, this research can be extended for the development of a recursive version of the proposed identification algorithm. The recursive algorithm should reduce the computation time to be executed online. Such algorithms can be particularly beneficial for those systems whose parameters change with time.
- The developed FTSM based control method for underactuated systems can be extended into an adaptive scheme. Specifically, the proposed control scheme guarantees the stability of lower sliding surfaces for crane dynamics. However, the same conclusion cannot be made for other underactuated systems. In order to address the issue, an adaptive scheme could be introduced into sliding function gains k_1 and k_2 to satisfy the condition $k_1 s_1 \neq k_2 s_2$, and guarantee that the closed-loop underactuated system is also stable.
- To improve performance of the prototype system presented in Chapter 6, a calibration process could be applied to estimate system parameters such as

the distance between cameras and pointing sensor. Specifically, the prototype consists of a vision camera, thermal camera, and RobotEye. As mentioned previously, the pointing device has a mirror which is placed directly above the thermal camera. The mirror has a point of rotation which is also known as the center point. The distance between this point and the reference axis of the camera frame is measured using normal tools. However, such measurements are prone to errors, which can affect the accuracy of the system, especially while applying TSA. Therefore, to improve the accuracy, a process calibration process could rectify the issue.

Bibliography

- Abdel-Rahman, E. and Nayfeh, A. (2003). Dynamics and control of cranes: A review. *Journal of Vibration and Control*, 9(7):863–908. doi: 10.1177/2F1077546303009007007.
- Bakker, B. and Heskes, T. (2003). Task clustering and gating for bayesian multitask learning. *Journal of Machine Learning Research*, 4(1):83–99. doi: 10.1109/jmlr.87.761053.
- Balabin, R. M. and Lomakina, E. I. (2011). Support vector machine regression (LS-SVM) an alternative to neural networks (ANN) for analytical chemistry? Comparison of nonlinear methods on near infrared (NIR) spectroscopy data. *Analyst*, 136(8):1703–1712. doi: 10.1039/c0an00387e.
- Bartolini, G., Pisano, A., and Usai, E. (2002). Second-order sliding-mode control of container cranes. *Automatica*, 38(10):1783–1790. doi: 10.1016/S0005-1098.
- Behera, A. K. and Bandyopadhyay, B. (2015). Steady-state behaviour of discretized terminal sliding mode. *Automatica*, 54:176–181. doi: 10.1016/j.automatica.2015.02.009.
- Borchani, H., Varando, G., Bielza, C., and Larrañaga, P. (2015). A survey on multi-output regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(5):216–233. doi: 10.1002/widm.1233.
- Chen, S., Xu, H., Liu, D., Hu, B., and Wang, H. (2014). A Vision of IoT: Applications, Challenges, and Opportunities With China Perspective. *IEEE Internet of Things Journal*, 1(4):349–359. doi: 10.1109/JIOT.2014.2337336.

Davila, J., Fridman, L., and Levant, A. (2005). Second-Order Sliding-Mode Observer for Mechanical Systems. *IEEE Transactions on Automatic Control*, 50(11):1785–1789. doi: 10.1109/TAC.2005.858636.

Delgado, C. J. M., dos Santos, P., and de Carvalho, J. L. (2006). A new insight to the matrices extraction in a MOESP type subspace identification algorithm. *International Journal of Systems Science*, 37(8):565–574. doi: 10.1080/00207720600784486.

Favoreel, W., De Moor, B., and Van Overschee, P. (2000). Subspace state space system identification for industrial processes. *Journal of Process Control*, 10(2-3):149–155. doi: 10.1016/S0959-1524(99)00030-X.

Feng, Y., Zheng, J., Yu, X., and Truong, N. V. (2009). Hybrid terminal sliding-mode observer design method for a permanent-magnet synchronous motor control system. *IEEE Transactions on Industrial Electronics*, 56(9):3424–3431. doi: 10.1109/TIE.2009.2025290.

Galias, Z. and Yu, X. (2007). Euler’s discretization of single input sliding-mode control systems. *IEEE Transactions on Automatic Control*, 52(9):1726–1730. doi: 10.1109/TAC.2007.904289.

Gao, W., Wang, Y., and Homaifa, A. (1995). Discrete-time variable structure control systems. *IEEE transactions on Industrial Electronics*, 42(2):117–122. doi: 10.1109/41.370376.

Garnier, H., Mensler, M., and Richard, A. (2003). Continuous-time model identification from sampled data: implementation issues and performance evaluation. *International journal of Control*, 76(13):1337–1357. doi: 10.1080/0020717031000149636.

- Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F. J., and Marín-Jiménez, M. J. (2014). Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292. doi: 10.1016/j.patcog.2014.01.005.
- Ha, Q. P., Nguyen, M. T., Li, J., and Kwok, N. M. (2013). Smart Structures with Current-Driven MR Dampers: Modelling and Second-Order Sliding Mode Control. *IEEE/ASME Transactions on Mechatronics*, 18(6):1702–1711. doi: 10.1109/TMECH.2013.2280282.
- Hoang, V. T., Phung, M. D., Dinh, T. H., and Ha, Q. P. (2018). Angle-Encoded Swarm Optimization for UAV Formation Path Planning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. doi: 10.1109/ijsr.2016.11.008.
- Huang, J., Guan, Z.-H., Matsuno, T., Fukuda, T., and Sekiyama, K. (2010). Sliding-mode velocity control of mobile-wheeled inverted-pendulum systems. *IEEE Transactions on Robotics*, 26(4):750–758. doi: 10.1109/TRO.2010.2053732.
- Huang, X., Yan, Y., and Huang, Z. (2017). Finite-time control of underactuated spacecraft hovering. *Control Engineering Practice*, 68(Supplement C):46–62. doi: 10.1016/j.conengprac.2017.08.006.
- Hui, L. and Li, J. (2009). Terminal sliding mode control for spacecraft formation flying. *IEEE Transactions on Aerospace and Electronic Systems*, 45(3):835–846. doi: 10.1109/TAES.2009.5259168.
- Jin, M., Lee, J., Chang, P. H., and Choi, C. (2009). Practical nonsingular terminal sliding-mode control of robot manipulators for high-accuracy tracking control. *IEEE Transactions on Industrial Electronics*, 56(9):3593–3601. doi: 10.1109/TIE.2009.2024097.

Kolar, B., Rams, H., and Schlacher, K. (2017). Time-optimal flatness based control of a gantry crane. *Control Engineering Practice*, 60(Supplement C):18–27. doi: 10.1016/j.conengprac.2016.11.008.

Kruger, U., Li, P., and Irwin, G. W. (2006). Identification of dynamic systems under closed-loop control. *International Journal of Systems Science*, 37(3):181–195. doi: 10.1080/00207720600566933.

Ku, W., Storer, R. H., and Georgakis, C. (1995). Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 30(1):179–196. doi: 10.1016/0169-7439(95)00076-3.

Lee, H.-H. (2004). A new design approach for the anti-swing trajectory control of overhead cranes with high-speed hoisting. *International Journal of Control*, 77(10):931–940. doi: 10.1080/00207170412331270550.

Lee, S. H., Park, J. B., and Choi, Y. H. (2009). Finite time control of nonlinear underactuated systems using terminal sliding surface. In *IEEE International Symposium on Industrial Electronics*, pages 626–631. doi: 10.1109/ISIE.2009.5220307.

Levant, A. (2003). Higher-order sliding modes, differentiation and output-feedback control. *International Journal of Control*, 76(9-10):924–941. doi: 10.1080/0020717031000099029.

Li, S., Du, H., and Yu, X. (2014). Discrete-time terminal sliding mode control systems based on Euler’s discretization. *IEEE Transactions on Automatic Control*, 59(2):546–552. doi: 10.1109/TAC.2013.2273267.

Li, S., Zhou, M., and Yu, X. (2013). Design and implementation of terminal sliding mode control method for PMSM speed regulation system. *IEEE Transactions on Industrial Informatics*, 9(4):1879–1891. doi:10.1109/TII.2012.2226896.

- Liu, J., Laghrouche, S., Harmouche, M., and Wack, M. (2014). Adaptive-gain second-order sliding mode observer design for switching power converters. *Control Engineering Practice*, 30(Supplement C):124–131. doi: 10.1016/j.conengprac.2013.10.012.
- Ljung, L. (1998). System identification. In *Signal Analysis and Prediction*, pages 163–173. Springer. doi: 10.1016/1998.05.045.
- Ma, B. L. (2007). Comment 'Design of a stable sliding-mode controller for a class of second-order underactuated systems'. *IET Control Theory and Applications*, 1(4):1186–1187. doi: 10.1049/iet-cta.2009.0345.
- Madani, T., Daachi, B., and K, D. (2016). Non-singular terminal sliding mode controller: Application to an actuated exoskeleton. *Mechatronics*, 33:136–145. doi:10.1016/j.mechatronics.2015.10.012.
- Moon, S. C., Lee, W. G., and Lee, S. G. (2013). Adaptive sliding mode control of overhead cranes with varying cable length. *Journal of Mechanical Science and Technology*, 27(3):885–893. doi: 10.1007/s12206-013-0204-x.
- Ngo, Q. and Hong, K. (2012). Adaptive sliding mode control of container cranes. *IET Control Theory and Applications*, 6(5):662–668. doi: 10.1049/iet-cta.2010.0764.
- Olfati-Saber, R. (2001). *Nonlinear control of underactuated mechanical systems with application to robotics and aerospace vehicles*. PhD thesis, Massachusetts Institute of Technology. doi: 10.1109/TRO.2010.2053732.
- Olson, E. (2011). AprilTag: A robust and flexible visual fiducial system. In *2011 IEEE International Conference on Robotics and Automation*, pages 3400–3407. IEEE. doi: 10.1109/ICRA.2011.5979561.

Park, K.-B. and Lee, J.-J. (1996). Comments on "A robust MIMO terminal sliding mode control scheme for rigid robotic manipulators". *IEEE Transactions on Automatic Control*, 41(5):761–762. doi: 10.1109/9.489220.

Pathak, K., Franch, J., and Agrawal, S. K. (2005). Velocity and position control of a wheeled inverted pendulum by partial feedback linearization. *IEEE Transactions on Robotics*, 21(3):505–513. doi:10.1109/TRO.2004.840905.

Pelckmans, K., Goethals, I., Brabanter, J. D., Suykens, J. A. K., and Moor, B. D. (2005). Componentwise least squares support vector machines. In *Support Vector Machines: Theory and Applications*, pages 77–98. Springer. doi: 10.1007/109846973.

Qin, S. J. (2006). An overview of subspace identification. *Computers & Chemical engineering*, 30(10):1502–1513. doi: 10.1016/j.compchemeng.2006.05.045.

Raja Ismail, R. M. T., That, N. D., and Ha, Q. P. (2015). Modelling and robust trajectory following for offshore container crane systems. *Automation in Construction*, 59:179–187. doi: 10.1016/j.autcon.2015.05.003.

Rajic, N., Brooks, C., Wang, J., Forrester, C., and Swanton, G. (2018). Thermoelastic Stress Analysis for Structural Assessment of Aerospace Composites. *21st International Conference on Composite Materials*, pages 123–128. doi:10.1177/1475921714548936.

Rajic, N. and Street, N. (2014). A performance comparison between cooled and uncooled infrared detectors for thermoelastic stress analysis. *Quantitative InfraRed Thermography Journal*, 11(2):207–221. doi: 10.1080/17686733.2014.962835.

Sakagami, T., Izumi, Y., and Kubo, S. (2010). Application of infrared thermography to structural integrity evaluation of steel bridges. *Journal of Modern Optics*, 57(18):1738–1746. doi: 10.1080/09500340.2010.511289.

Sarpturk, S. Z., Istefanopoulos, Y., and Kaynak, O. (1987). On the stability of discrete-time sliding mode control systems. *IEEE Transactions on Automatic Control*, 32(10):930–932. doi: 10.1109/TAC.1987.1104468.

Saux, V. L. and Doudard, C. (2017). Proposition of a compensated pixelwise calibration for photonic infrared cameras and comparison to classic calibration procedures: Case of thermoelastic stress analysis. *Infrared Physics & Technology*, 80:83–92. doi: 10.1016/j.infrared.2016.11.008.

Selmic, R. R. and Lewis, F. L. (2000). Deadzone compensation in motion control systems using neural networks. *IEEE Transactions on Automatic Control*, 45(4):602–613. doi: 10.1109/9.847098.

Shotorbani, A. M., Ajami, A., Zadeh, S. G., Aghababa, M. P., and Mahboubi, B. (2014). Robust terminal sliding mode power flow controller using unified power flow controller with adaptive observer and local measurement. *IET Generation, Transmission and Distribution*, 8(10):1712–1723. doi: 10.1049/iet-gtd.2013.0637.

Shtessel, Y., Edwards, C., Fridman, L., and Levant, A. (2014). *Sliding mode control and observation*, volume 10. Springer. doi: 10.1007/978-0-8176-4893-0.

Söderström, T. and Stoica, P. (1989). System identification. doi: 10.1016/S0959-1524(99)00030-X.

Sotomayor, O. A. Z., Park, S. W., and Garcia, C. (2003). Multivariable identification of an activated sludge process with subspace-based algorithms. *Control Engineering Practice*, 11(8):961–969. doi: 10.1016/S0967-0661(02)00210-1.

Sueishi, T., Oku, H., and Ishikawa, M. (2015). Mirror-based high-speed gaze controller calibration with optics and illumination control. In *IEEE International Conference on Intelligent Robots and Systems*, pages 3064–3070. doi: 10.1109/IROS.2015.7353800.

Sun, N., Fang, Y., Chen, H., and He, B. (2015). Adaptive nonlinear crane control with load hoisting/lowering and unknown parameters: design and experiments. *IEEE/ASME Transactions on Mechatronics*, 20(5):2107 – 2119. doi: 10.1109/TMECH.2014.2364308.

Thibeault, N. and Smith, R. (2001). Fundamental limits in robustness and performance for unstable, underactuated systems. *IEEE Transactions on Automatic Control*, 46(8):1265–1268. doi: 10.1109/9.940931.

Thissen, U., Pepers, M., Üstün, B., Melssen, W. J., and Buydens, L. M. C. (2004). Comparing support vector machines to PLS for spectral regression applications. *Chemometrics and Intelligent Laboratory Systems*, 73(2):169–179. doi: 10.1016/j.chemolab.2004.01.002.

Van Overschee, P. and De Moor, B. (1994). N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30(1):75–93. doi: 10.1016/0005-1098(94)90230-5.

Van Overschee, P. and De Moor, B. (1995). A unifying theorem for three subspace system identification algorithms. *Automatica*, 31(12):1853–1864. doi: 10.1016/0005-1098(95)00072-0.

Van Overschee, P. and De Moor, B. L. (1996). *Subspace identification for linear systems: Theory, Implementation, and Applications*. Springer Science & Business Media. doi: 10.1007/978-1-4613-0465-4.

Vapnik, V. (2013). *The nature of statistical learning theory*. Taylor & Francis Group. doi: 10.1080/00401706.1996.10484565.

Vázquez, C., Collado, J., and Fridman, L. (2013). Control of a parametrically excited crane: A vector Lyapunov approach. *IEEE Transactions on Control Systems Technology*, 21(6):2332 – 2340. doi: 10.1109/TCST.2012.2233739.

- Verhaegen, M. and Dewilde, P. (1992). Subspace model identification part 1. The output-error state-space model identification class of algorithms. *International Journal of Control*, 56(5):1187–1210. doi: 10.1080/00207179208934363.
- Wang, J. and Qin, S. J. (2002). A new subspace identification approach based on principal component analysis. *Journal of Process Control*, 12(8):841–855. doi: 10.1016/S0959-1524(02)00016-1.
- Wang, W., Yi, J., Zhao, D., and Liu, D. (2004). Design of a stable sliding-mode controller for a class of second-order underactuated systems. *IEE Proceedings-Control Theory and Applications*, 151(6):683–690. doi:10.1109/9.362847.
- Whitley, D. (1994). A genetic algorithm tutorial. *Statistics and computing*, 4(2):65–85. doi: 10.1007/BF00175354.
- Wong, A. K., Rajic, N., and Nguyen, Q. (2014). Seeing stresses through the thermoelectric lens-a retrospective and prospective from an Australian viewpoint. *International Journal for Experimental Mechanics*, 51(1):1–15. doi: 10.1111/str.12116.
- Wood, D. and Bishop, M. (2012). A novel approach to 3D laser scanning. In *Proceedings of Australasian Conference on Robotics and Automation (ACRA)*. doi: 10.1016/2012.01.005.
- Wu, P., Pan, H. P., Ren, J., Yang, C. J., and Others (2015). A New Subspace Identification Approach Based on Principal Component Analysis and Noise Estimation. *Industrial & Engineering Chemistry Research*, 54(18):5106–5114. doi: 10.1021/ie504824a.
- Xu, B. C. and Liu, X. L. (2012). Identification algorithm based on the approximate least absolute deviation criteria. *International Journal of Automation and Computing*, 9(5):501–505. doi: 10.1007/s11633-012-0673-x.

- Xu, R. and Özgüner, Ü. (2008). Sliding mode control of a class of underactuated systems. *Automatica*, 44(1):233–241. doi: 10.1016/j.automatica.2007.05.014.
- Xu, S., An, X., Qiao, X., Zhu, L., and Li, L. (2013). Multi-output least-squares support vector regression machines. *Pattern Recognition Letters*, 34(9):1078–1084. doi: 10.1016/j.patrec.2013.01.015.
- Yao, B. and Tomizuka, M. (1996). Smooth robust adaptive sliding mode control of manipulators with guaranteed transient performance. *Journal of Dynamic Systems, Measurement, and Control*, 118(4):764–775. doi: 10.1115/1.2802355.
- Young, K. D., Utkin, V. I., and Ozguner, U. (1999). A control engineer’s guide to sliding mode control. In *IEEE Transactions on Control Systems Technology*, volume 7, pages 328–342. IEEE. doi: 10.1109/87.761053.
- Yu, S., Yu, X., Shirinzadeh, B., and Man, Z. (2005). Continuous finite-time control for robotic manipulators with terminal sliding mode. *Automatica*, 41(11):1957–1964. doi: 10.1016/j.automatica.2005.07.001.
- Yu, X. and Zhihong, M. (1996). On finite time mechanism: Terminal sliding modes. In *Variable Structure Systems, 1996. VSS’96.*, pages 164–167. doi: 10.1109/VSS.1996.578596.
- Yu, X. and Zhihong, M. (1998). Multi-input uncertain linear systems with terminal sliding-mode control. *Automatica*, 34(3):389–392. doi: 10.1016/S0005-1098.
- Yu, X. and Zhihong, M. (2002). Fast terminal sliding-mode control design for non-linear dynamical systems. *IEEE Transaction on Circuits and Systems^2I: Fundamental Theory and Applications*, 49(2). doi: 10.1016/S0005-1098.
- Zak, M. (1988). Terminal attractors for addressable memory in neural networks. *Physics Letters A*, 133(1-2):18–22. doi:10.1107/8.362847.

Zhang, W., Liu, X., Ding, Y., and Shi, D. (2012). Multi-output LS-SVR machine in extended feature space. In *2012 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications (CIMSA)*, pages 130–134. IEEE. doi: 10.1109/CIMSA.2012.6269600.

Zheng, Y., Jing, Y.-w., and Yang, G.-h. (2006). Design of approximation law for discrete-time variable structure control systems. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 4969–4973. IEEE. doi: 10.1109/CDC.2006.377116.

Zhihong, M., Paplinski, A. P., and Wu, H. R. (1994). A robust MIMO terminal sliding mode control scheme for rigid robotic manipulators. *IEEE Transactions on Automatic Control*, 39(12):2464–2469. doi: 10.1109/9.362847.

Zou, A.-M., Kumar, K. D., Hou, Z.-G., and Liu, X. (2011). Finite-time attitude tracking control for spacecraft using terminal sliding mode and chebyshev neural network. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 41(4):950–963. doi: 10.1109/TSMCB.2010.2101592.