



Erasmus Mundus



Master Thesis

ERASMUS MUNDUS MASTER of Mechanical Engineering

Mr. NGUYEN HUU NGHIA

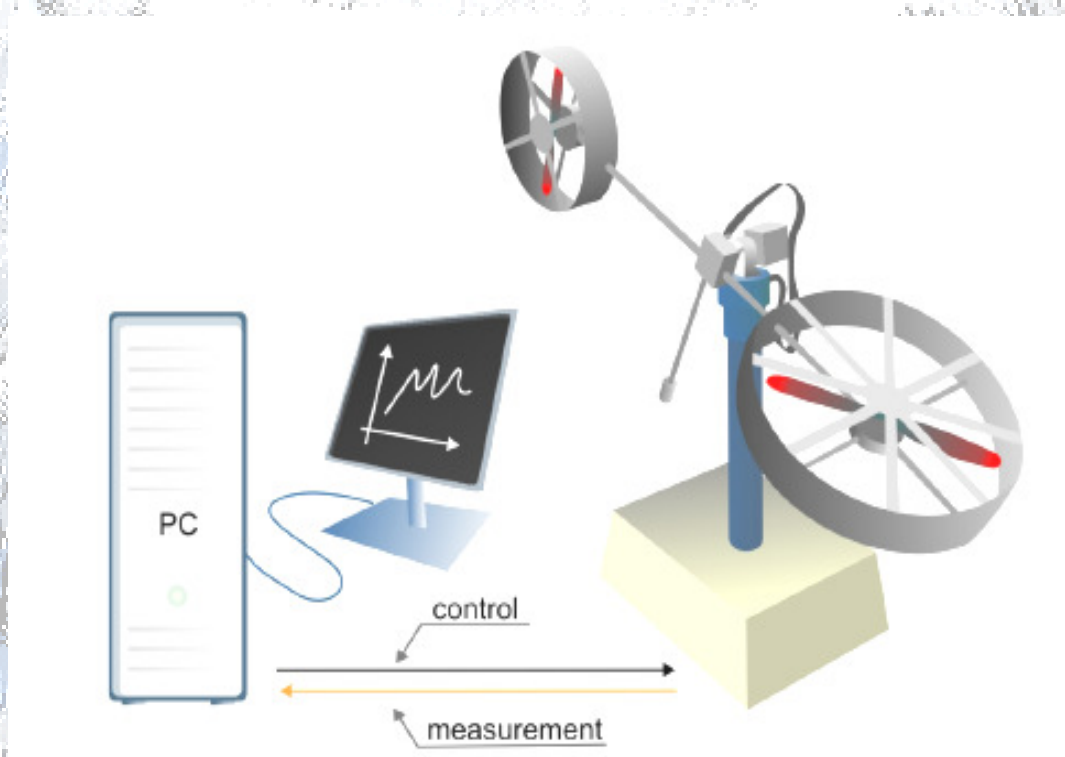
2005-2007
(First course)



POLYTECHNICAL UNIVERSITY OF CATALONIA
BARCELONA

Master Thesis
ERASMUS MUNDUS Master of Mechanical Engineering

MODELLING, SIMULATION, AND CALIBRATION OF TWIN ROTOR MIMO SYSTEM



Author: Nguyen, Huu Nghia
Advisor: Prof. Puig Cayuela, Vicenç

June 2007

Abstract

Helicopter is an aircraft that plays important role in transporting products, people...in society nowadays. It is complex mechanical equipment that corresponds many fields such as fluid mechanics, mechanics, control...Design it probably easy but fabrication and control it are not simple problem.

The aim of research is to obtain a simulation and control model for the setup that has the principle of function like a real helicopter in laboratory in Automatic Control Department in Technical University Catalonia – Barcelona. This setup names Twin Rotor Multi Inputs - Multi Outputs System (TRMS) is manufactured by the Feedback Instruments Limited Company. It serves as a guide for the control tasks and provides useful information about the physical behavior of the system. It is also useful setup for study and practice of students to have a clearer look.

On the main originalities of the present master thesis is the use on a control oriented model based on the use of a model that has linear structure but parameters varying with the operating point. This type of model is known as Linear Parameter Varying model (shortly, LPV model). Two procedures to obtain such a model are proposed. One based of rearranging the non-linear equations in such a way that the LPV parameters appear linearly. The second is based on linearizing the non-linear model around different operating points and the interpolation the parameters between them.

Finally, the LPV model for the TRMS system obtained using either of the procedures described above can be calibrated using standard parameter estimation algorithms available in the Identification Toolbox in MATLAB.

Acknowledgment

The thesis was done in Automatic Control Department of Technical University of Catalonia (UPC) of Barcelona - Spain. The work was supervised by Prof. Puig Cayuela Vicenç.

I would like to express my thanks and gratitude to Prof. Puig Cayuela Vicenç for continuous support, the helpful advice, and the valuable guidance throughout the work.

I would like to express my thanks to Prof. Jean-Claude Boyer, coordinator of Erasmus Mundus Master of Mechanical Engineering (EMMME), created a very good environment for us to study in INSA of Lyon - France as well as Prof. Ana Barjau, coordinator of EMMME in Spain, helped us very much during the courses.

I thank all the professors, members in two institutes (INSA of Lyon, UPC of Barcelona) helped us during studying such as teaching, organization, and registration...for the courses. My thanks are also to the staffs who taught and encouraged my study in Department of Mechanical Engineering in Cantho University-Vietnam.

I would like to commemorate Prof. Jean-Paul Long, responsible of the EMMME in INSA, has just forever gone to another world. He taught and deeply conveyed the knowledge to us during the courses.

Finally, I would like to express my special thanks to my family and friends for the patience and encouragement.

Barcelona, June 2007.

Nguyen Huu Nghia

Table of contents

Abstract	i
Acknowledgment	ii
Table of contents	iv
List of figures	v
List of tables	vi
1. INTRODUCTION	1
1.1 Motivation.....	1
1.2 Project objective	5
1.3 Content of chapters	6
2. MODELLING THE SYSTEM	7
2.1 Description of the system	7
2.2 Characteristics of the motors	9
2.2.1 Main motor	10
2.2.2 Tail rotor	10
2.3 Equations of system	11
2.3.1 Physical model	11
2.3.2 Mathematical model	12
3. SIMULATOR OF TRMS SYSTEM	19
3.1 Simulink TRMS system	19
3.2 Some basic concepts of mathematical models for control.....	21
3.2.1 State.....	22
3.2.2 State variables.....	22
3.2.3 State vvector	22
3.2.4 State space	22
3.2.5 State pace equation.....	22
3.3 Linear parameter varying (LPV) model for controlling the TRMS system	24
4. CALIBRATION OF TRMS SYSTEM CONTROL MODEL	31
4.1 Introduction	31
4.2 Calibration of LPV model using Matlab (pem command)	31
4.2.1 Pem and mathematical model	31
4.2.2 Implementation method	33
4.2.3 The results of calibration.....	37
4.3 Calibration of a LPV model coming from linearization toolbox	45
5. CONCLUSION AND FUTURE WORK	54
A. APPENDIX	55
References	67

List of figures

Figure 1.1 Model of aerodynamic force	1
Figure 1.2 Helicopter UH-1	2
Figure 1.3 Main rotor Blade system.....	3
Figure 1.4 Tail rotor blade	3
Figure 2.1 The twin rotor MIMO system.....	7
Figure 2.2 Aerodynamic model of the TRMS.....	8
Figure 2.3 Position sensor.....	9
Figure 2.4 Measuring of main rotor characteristics	10
Figure 2.5 Measuring of tail rotor characteristics	11
Figure 2.6 Schematic diagram of TRMS.....	12
Figure 2.7 Side view of TRMS.....	13
Figure 2.8 Top view of TRMS.....	16
Figure 3.1 Simulink in Matlab of TRMS	20
Figure 3.2 Subsystem of TRMS.....	21
Figure 3.3 Simulator of TRMS	27
Figure 3.4 Horizontal signal outputs	28
Figure 3.5 Vertical signal outputs	28
Figure 3.6 Horizontal position of LPV2 and Simulink model.....	29
Figure 3.7 Vertical position of LPV2 and simulink model	29
Figure 4.1 Function Fcn	33
Figure 4.2 Calculating coefficients in state space	36
Figure 4.3 Operating points of the system	37
Figure 4.4 The 1 st operating point calibration.....	38
Figure 4.5 The 2 nd operating point calibration.....	39
Figure 4.6 The 3 rd operating point calibration	40
Figure 4.7 the 4 th operating point calibration.....	41
Figure 4.8 Estimated coefficients of matrix A.....	43
Figure 4.9 Estimated coefficients of matrix B.....	44
Figure 4.10 Linearization toolbox window.....	46
Figure 4.11 The 1 st operating point calibration by linearization tool.....	47
Figure 4.12 The 2 nd operating point calibration by linearization tool.....	48
Figure 4.13 The 3 rd operating point calibration by linearization tool	49
Figure 4.14 The 4 th operating point calibration by linearization tool.....	50
Figure 4.15 Estimated coefficients of matrix A by linearization	52
Figure 4.16 Estimated coefficients of matrix B by linearization	53

List of tables

Table 2.1 TRMS model parameters in vertical axis.....	15
Table 2.2 TRMS model parameters in horizontal axis	18
Table 4.1 Result of the calibration	42
Table 4.2 Stable points.....	42
Table 4.3 Result of calibration by linearization tool.....	51

1. INTRODUCTION

1.1 Motivation

Helicopter is an aircraft which is lifted and propelled by one or more horizontal rotors consisting of two or more rotor blades. It has the ability to take off and land vertically and to maintain a steady hover in the air over a single point on the ground. This allows the helicopter to land and to take off from, including heliports in the middle of busy cities and rugged terrain in remote areas. The helicopter is used for rescue, medical evacuation and as an observation platform. Other operation that involve the use of helicopters are fire fighting, tours, as an aerial crane, logging, personnel, personnel transport, electronic news gathering, law enforcement, military and for pleasure. In reality, helicopters can land everywhere that only need the ground larger than 1.5 times the helicopters' blades.

Helicopter like aircraft can fly due to aerodynamic force or Zhukovski force. That is the different pressure between top surface and bottom surface of the blade when the air flows cover it.

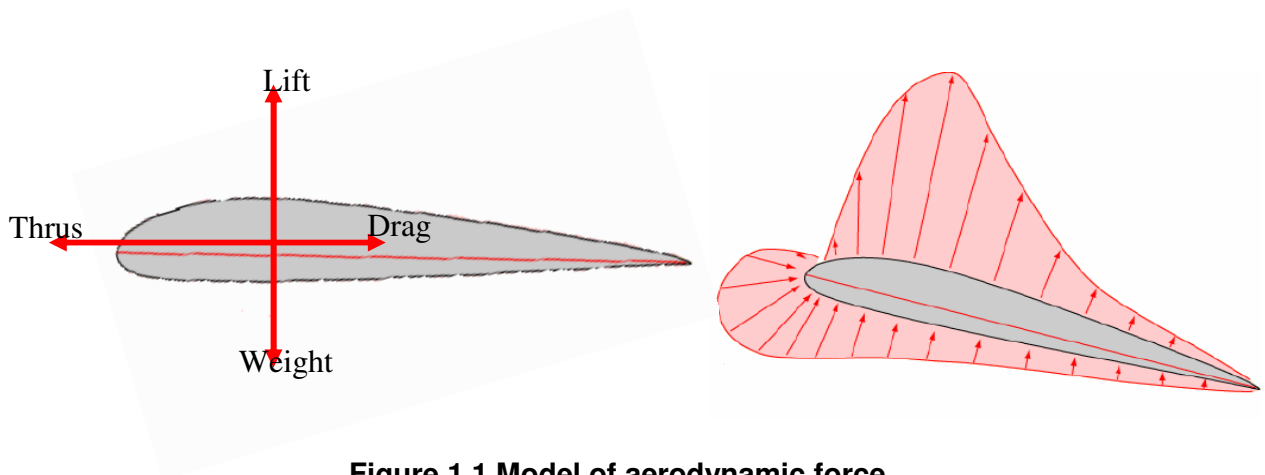


Figure 1.1 Model of aerodynamic force.

The air flows around the blade that has lift and simultaneously appears drag. At the bottom of the blade has pressure higher than the top, recently, there is a force from the bottom to the top and is perpendicular with the blade. This different depends on blade's profile, angle of attack-angle of flowing air (the acute angle measured between the chord of an airfoil and the relative wind) and rotation of the blades. So the blades rotate with certain speed that is enough for helicopter flying due to aerodynamic force, it means this force higher than helicopter's weight. The shape and the angle of attack move through the air will determine how much lift force is created. After the helicopter lifted off the ground, the pilot can tilt the blades, causing the helicopter to tip forward or backward or sideward.

There are many kinds of helicopters, nowadays; the popular kind is helicopter's model that has a lift blade and tail blade.



Figure 1.2 Helicopter UH-1

The blades of the helicopter are airfoils with a very high aspect ratio (length to chord). The angle of incidence is adjusted by means of the control from pilots.

The main rotor of the helicopter may have two, three, four, five or six blades, depending upon the design. The main rotor blades are hinged to the rotor head in such a manner that they have limited movement up and down and also they can change the pitch (angle of incidence). The controls for the main rotor are called Collective and Cyclic Controls.

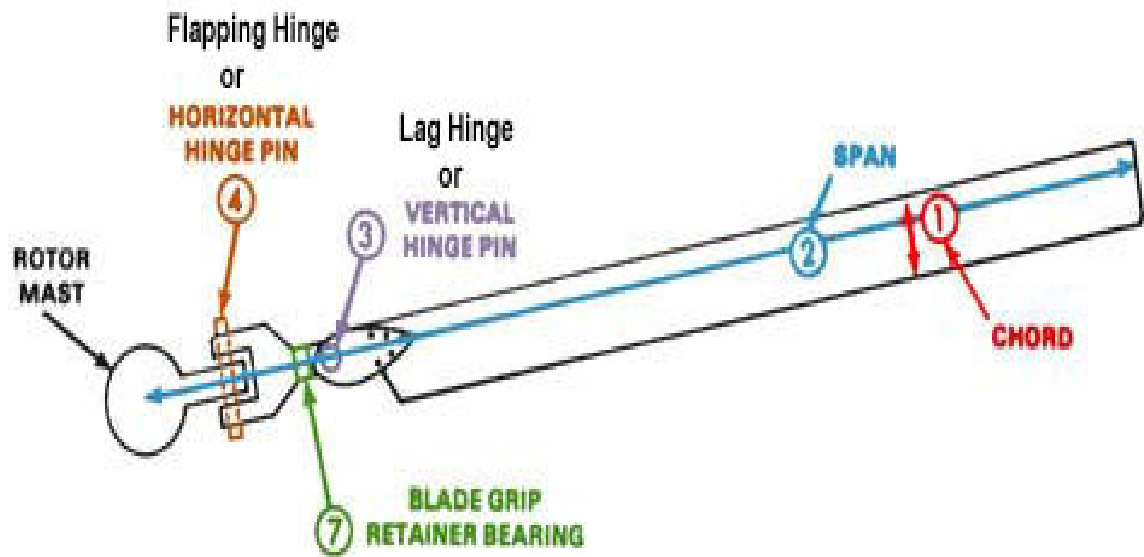


Figure 1.3 Main Rotor Blade System

The tail rotor is small blades may have two or four blades and mounted on the tail of the helicopter, it rotates in the vertical plane. The tail rotor is controlled by the rudder pedals. Its pitch can be changed as required to turn the helicopter in the direction desired.



Figure 1.4 Tail Rotor Blade

When the blades lift, the body of helicopter also rotates around the blade at contrast direction. So avoiding this phenomena, it has a tail blade at vertical direction, blow the air in horizontal direction. Created moment of tail blade can eliminate with moment cross-action of the body. Besides that, it plays role in changing the direction's helicopter into horizontal plane.

This model is popular about 95% using it today because of simple technology, high stability, and high reliability.

1.2 Project objective

This project is based on a two degree of freedom helicopter equipment (Twin Rotor MIMO System developed by Feedback) available at the laboratories of the Advanced Control Systems Research in the Automatic Control Department (ESAI) of Technical University of Catalonia (UPC). This model is a good multivariable control benchmark widely used in the literature. It allows illustrating the control of helicopter mechanics with two degrees of freedom that rotates around two directions. In order to design a controller a control-oriented model is required. On the other hand, to validate such controller before going to the real system, an accurate model in simulation will be used. To obtain both models (control and simulation oriented) mechanical physical laws should be applied. Some of the parameters of this model can be obtained from the manual of this equipment provided by the manufacturer, while some others should be obtained by experiments such as: magnitudes of physical propeller, length, mass, inertia, coefficients of friction, and impulse force...

The objective of this project is:

- to develop, calibrate and validate a simulation model
- to develop, calibrate and validate a control model

As a starting point, the model provided in the manual and some additional references will be used.

1.3 Content of chapters

Chapter 1 is brief introduction of structure, functional principle of the real helicopter and objective this project.

Chapter 2 consists of description of system, find and calculate its movements by experimental data.

Chapter 3 implements the simulation of the system and the model LPV.

Chapter 4 calibrates the system and technical identification.

Chapter 5 Conclusion and future work.

Appendix

2. MODELLING THE SYSTEM

2.1 Description of system



Figure 2.1 The Twin Rotor MIMO system

The Twin Rotor MIMO System (TRMS) is a laboratory setup that is designed for control experiments. In certain aspects its behaviour resembles that of a helicopter. As Figure 2.1 the TRMS mechanical unit has two rotors placed on a beam together with a counterbalance whose arm with a weight at its end is fixed to the beam at the pivot and it determines a stable equilibrium position.

The TRMS consists of a beam pivoted on its base in such a way that it can rotate freely both in the horizontal and vertical planes. Either the horizontal or the vertical degree of freedom can be restricted to 1 degree of freedom using nylon screws found near pivot point. At both ends of the beam there are rotors (the main and tail rotors) driven by DC motors. This device is a multivariable, nonlinear and strongly coupled system, with degrees of freedom on the pitch and yaw angle denoted by θ_v , θ_h .

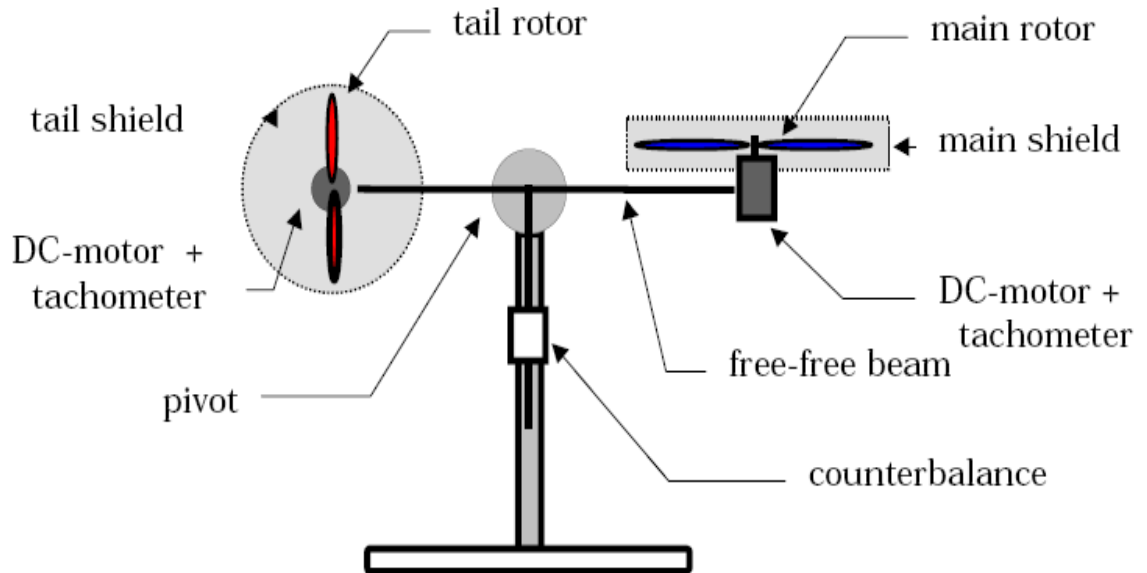


Figure 2.2 Aerodynamic model of the TRMS.

The state of the beam is described by four process variables: horizontal and vertical angles measured by position sensors fitted at the pivot, and two corresponding angular velocities. The tachogenerators are used to measure the angular velocities of rotors.

In a real helicopter, the aerodynamic force is controlled by changing the angle of attack but in this model the aerodynamic forces are controlled by varying the speed of rotors, the angle of attack is fixed.

A change in the voltage value results in a change of rotation speed of the propeller which results in a change of the corresponding position of the beam.

Optical encoders: the light beams emitted by two light sources (A and B) go through two rings of slits on the disc. The slits have a phase difference, so that the electric outputs of the receivers (A and B) are rectangular waves with a phase difference. The sign of the phase difference allows the direction of rotation to be determined.

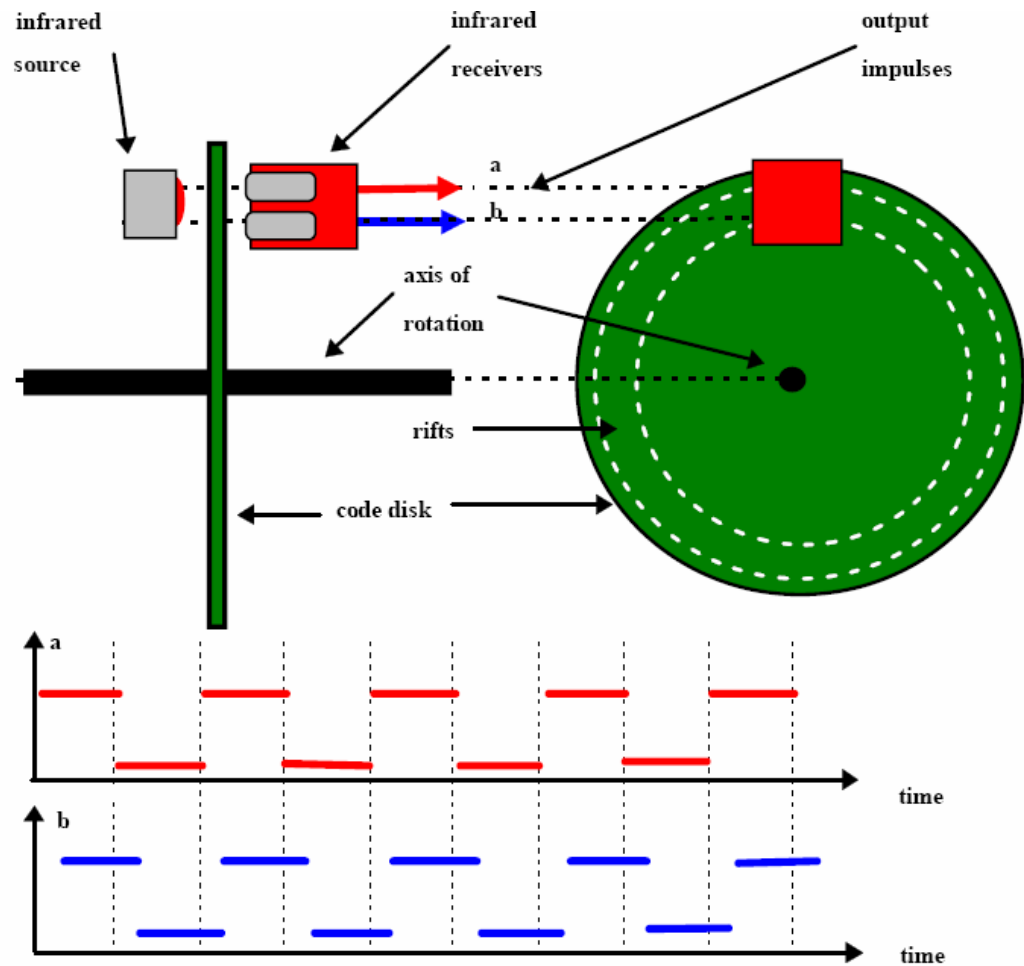


Figure 2.3 Position sensor

The control signal flows from the computer through the D/A converter of the data acquisition board. The D/A outputs is wired to the power amplifier output which drives the DC motor. The power amplifier and encoder interface are located in the TRMS box. This box is equipped with two switches: the main power switch and cutting off switch the DC motor power.

2.2 Characteristics of the motors

It is necessary to identify the following non-linear functions:

- Two non-linear inputs characteristics determining dependence of DC-motor rotational speed on voltage:

$$W_m = P(u_{rr}), w_t = P(u_{tt})$$

- Two non-linear characteristics determining dependence of propeller thrust on DC-motor rotational speeds:

$$F_h = F_h(w_t), F_v = F_v(w_m)$$

The static characteristics of the propellers should be measured in the case when not delivered with the equipment documentation or if the propellers were changed by user. In this case a proper electronic balance with voltage output [0-10V] is needed. The recommended range of input force is from 0 to 2 N.

2.2.1 Main motor

Having the main motor's characteristics is done by experiment. Making the measurements is correct, first block the beam so that it can rotate around the vertical axis. Place the electronic balance under the beam in such a way that it is pulled vertically up by the propeller. To balance the beam in the horizontal position attaches the beam. Connect the voltage output of the electronic balance to A/D input No.3 of the PLC-812PG or RT-DAC data acquisition board.

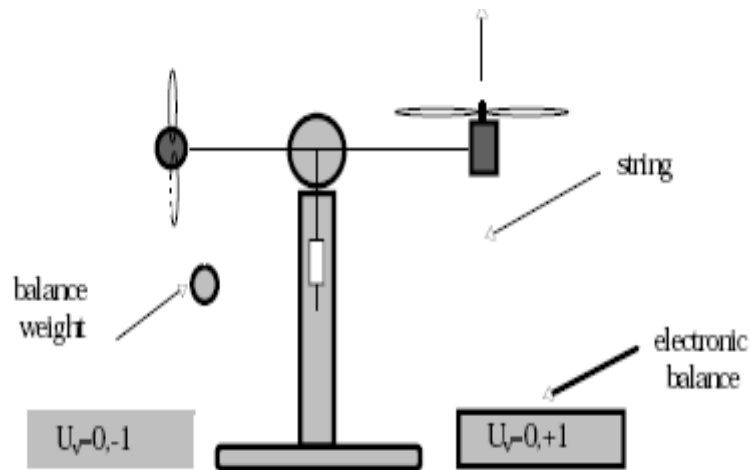


Figure 2.4 Measuring of main rotor characteristics

The result we obtain:

$$w_m(u_{rr}) = 90.99u_{rr}^6 + 599.73u_{rr}^5 - 129.26u_{rr}^4 - 1238.64u_{rr}^3 + 63.45u_{rr}^2 + 1283.41u_{rr}$$

$$F_v(w_m) = -3.48 \cdot 10^{-12} w_m^5 + 1.09 \cdot 10^{-9} w_m^4 + 4.123 \cdot 10^{-6} w_m^3 - 1.632 \cdot 10^{-4} w_m^2 + 9.544 \cdot 10^{-2} w_m$$

2.2.2 Tail rotor

To balance the beam in the horizontal position attaches the beam. Connect the voltage output of the electronic balance to A/D input No.3 of the PLC-812PG or RT-DAC data acquisition board.

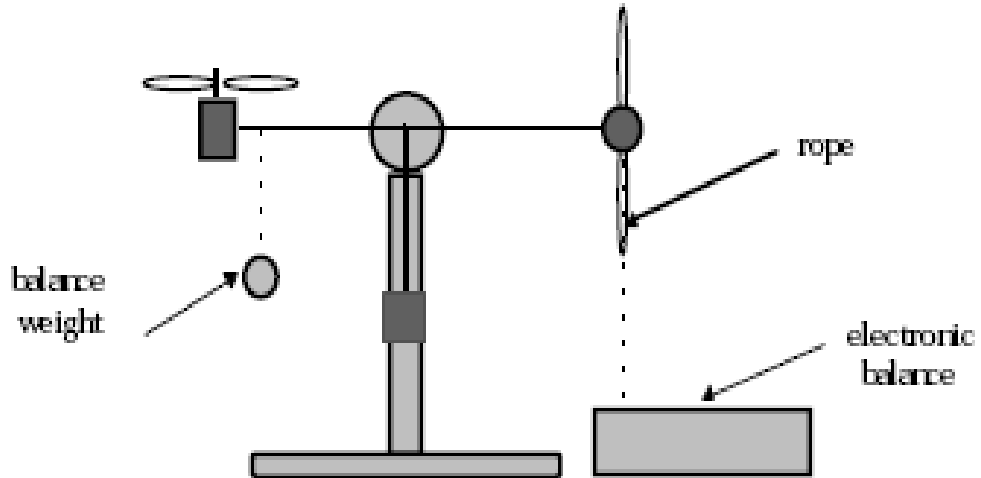


Figure 2.5 Measuring of tail rotor characteristic

The result after measuring

$$w_t(u_u) = 2020u_u^5 - 194.69u_u^5 - 4283.15u_u^3 + 262.27u_u^2 + 3796.83u_u$$

$$F_h(w_t) = -3.10^{-14} w_t^5 - 1.595.10^{-11} w_t^4 + 2.511.10^{-7} w_t^3 - 1.808.10^{-4} w_t^2 + 0.8080w_t$$

The rotational speed and the forces of the two rotors have the polynomial of the voltage. They are used in the subsystem and are the non-linear input characteristics.

2.3 Equations of system

2.3.1 Physical model

The static characteristics of the DC motors with propellers are non-linear function which connects between a linear dynamic system and static non-linearity and the input voltage is limited to the range +/-10 volts. The linear part is in the form of first order transfer functions $G_h=1/(T_hS+1)$ and $G_v=1/(T_vS+1)$. The non-linear function u_{tt} and u_{tr} are characteristics of the DC motors. The non-linear relations between the rotor's velocity and the resulting aerodynamic force can be approximated the quadratic functions:

$$F_h = \text{sign}(w_t)k_h \cdot w_t^2$$

$$F_v = \text{sign}(w_m)k_v \cdot w_m^2$$

k_h, k_v are positive constants.

Figure 2.6 presents the characteristic of the system. There are two propellers driven by DC motors at the end of the pivot. The articulated joint allows the beam to rotate in such a way that its ends move on spherical surfaces.

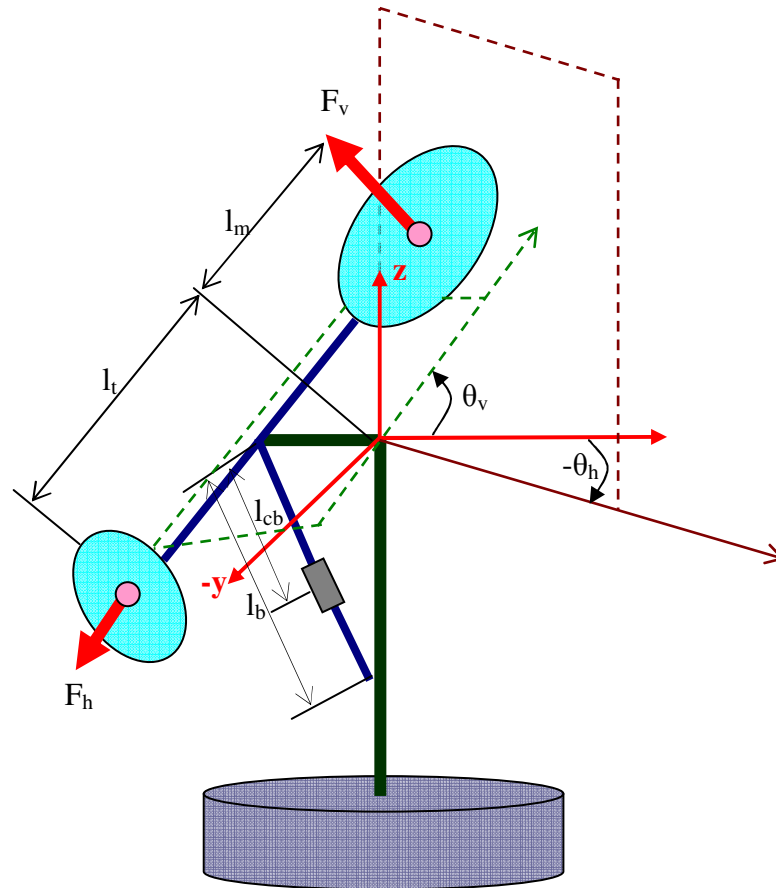


Figure 2.6 Schematic diagram of TRMS

The physical model is developed under some simplifying assumptions: the dynamics of the propeller subsystem is first order differential equations; friction of the system is of the viscous type and propeller-air subsystem accordance with the postulates of flow theory.

2.3.2 Mathematical model

Solving the system is considered the rotation of the beam in the vertical plane around the horizontal axis. Applying the Newton's second law of motion we obtain:

$$M_v = J_v \frac{d^2 \theta_v}{dt^2}$$

$$M_v = \sum_{i=1}^4 M_{vi}, J_v = \sum_{i=1}^8 J_{vi}$$

To determine these elements on above equation, consider the figure below and these parameters are described in the Table 1:

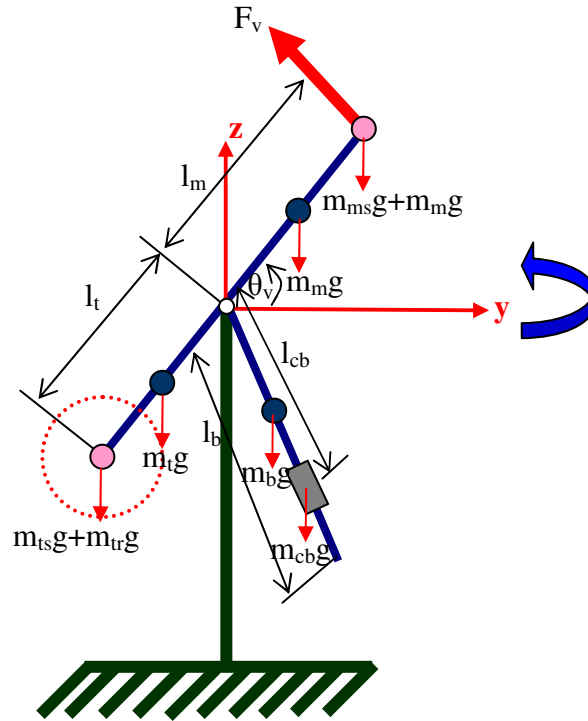


Figure 2.7 Side view of TRMS.

Moments of gravity forces (M_{v1}) applied to the beam and making it rotates around horizontal axis, consider the solution shown in Figure 2.7.

$$M_{v1} = \left\{ \left[\left(\frac{m_t}{2} + m_{tr} + m_{ts} \right) l_t - \left(\frac{m_m}{2} + m_{mr} + m_{ms} \right) l_m \right] \cos \theta_v - \left(\frac{m_b}{2} l_b + m_{cb} l_{cb} \right) \sin \theta_v \right\}$$

It can be expressed as

$$M_{v1} = g \{ [A-B] \cos \theta_v - C \sin \theta_v \}$$

Where:

$$A = \left(\frac{m_t}{2} + m_{tr} + m_{ts} \right) l_t$$

$$B = \left(\frac{m_m}{2} + m_{mr} + m_{ms} \right) l_m$$

$$C = \left(\frac{m_b}{2} l_b + m_{cb} l_{cb} \right)$$

Moments of propulsive forces (M_{v2}) applied to the beam

$$M_{v2} = I_m F_v(\omega_m)$$

Moment of the centrifugal forces (M_{v3}) corresponding to the motion of the beam around the vertical axis.

$$M_{v3} = -\omega_h^2 \left\{ \left(\frac{m_t}{2} + m_{tr} + m_{ts} \right) l_t + \left(\frac{m_m}{2} + m_{mr} + m_{ms} \right) + \left(\frac{m_b}{2} l_b + m_{cb} l_{cb} \right) \right\} \sin \theta_v \cos \theta_v$$

Or in the compact form:

$$M_{v3} = -\omega_h^2 (A + B + C) \sin \theta_v \cos \theta_v$$

$$\text{Where } \omega_h = \frac{d\theta_h}{dt} \quad (1)$$

Moment of friction (M_{v4}) depends on the angular velocity of the beam around the horizontal axis.

$$M_{v4} = -\omega_v k_v$$

$$\text{Where } \omega_v = \frac{d\theta_v}{dt} \quad (2)$$

According to fig 2.7 we can calculate components of the moment of inertia relative to the horizontal axis.

$$J_{mr} = m_{mr} l_m^2$$

$$J_m = m_m \frac{l_m^2}{3}$$

$$J_{cb} = m_{cb} l_{cb}^2$$

$$J_b = m_b \frac{l_b^2}{3}$$

$$J_{tr} = m_{tr} l_t^2$$

$$J_t = m_t \frac{l_t^2}{3}$$

$$J_{ms} = \frac{m_{ms}}{2} r_{ms}^2 + m_{ms} l_m^2$$

$$J_{ts} = m_{ts} r_{ts}^2 + m_{ts} l_t^2$$

Table 2.1 TRMS model parameters in vertical axis

Symbol	Parameter	Value	Unit
M_v	Total moment of force in vertical plane	$f(\theta_v)$	Nm
M_{v1}	Moments of gravity force	$f(\theta_v)$	Nm
M_{v2}	Moments of propulsive force	$f(\theta_v)$	Nm
M_{v3}	Moment of the centrifugal force	$f(\theta_v)$	Nm
M_{v4}	Moment of friction	$f(\theta_v)$	Nm
J_v	Sum of moments of inertia relative to the horizontal axis	0.055864	$\text{kg} \cdot \text{m}^2$
J_{mr}	Moments of inertia of the main DC-motor with main rotor	0.013132	$\text{kg} \cdot \text{m}^2$
J_m	Moment of inertia of main part of the beam	0.000278	$\text{kg} \cdot \text{m}^2$
J_{cb}	Moment of inertia of the counter weight	0.001149	$\text{kg} \cdot \text{m}^2$
J_b	Moment of inertia of the counter-weight beam	0.000495	$\text{kg} \cdot \text{m}^2$
J_{tr}	Moment of inertia of the tail motor with the tail rotor	0.012875	$\text{kg} \cdot \text{m}^2$
J_t	Moment of inertia of the tail part of the beam	0.000322	$\text{kg} \cdot \text{m}^2$
J_{ms}	Moment of inertia of the main shield	0.015622	$\text{kg} \cdot \text{m}^2$
J_{ts}	Moment of inertia of the tail shield	0.011962	$\text{kg} \cdot \text{m}^2$
θ_v	The pitch angle of the beam		rad
θ_h	The yaw angle of the beam		rad
ω_v	The angular velocity of the beam around the vertical axis	$f(\theta_v)$	rad/s
ω_h	The angular velocity of the beam around the horizontal axis	$f(\theta_h)$	rad/s
k_v	Constant	0.00545371	Nms/rad
k_h	Constant	0.055864	Nms/rad
w_m	The rotational velocity of main rotor	$f(u_{tt})$	rad/s
w_t	The rotational velocity of tail rotor	$f(u_{tr})$	rad/s
m_{mr}	Mass of the main DC-motor with main rotor	0.228	kg
m_m	Mass of the main DC-motor with main rotor	0.0145	kg
m_{tr}	Mass of the tail motor with the tail rotor	0.206	kg
m_t	Mass of the tail part of the beam	0.0155	kg
m_{cb}	Mass of the counter weight	0.068	kg
m_b	Mass of the counter-weight beam	0.022	kg
m_{ms}	Mass of the main shield	0.225	kg
m_{ts}	Mass of the tail shield	0.165	kg
l_m	The length of main part of the beam	0.24	m
l_t	The length of the tail part of the beam	0.25	m
l_b	The length of the counter-weight beam	0.26	m
l_{cb}	The distance between the counter-weight and the joint	0.13	m
g	Gravitational acceleration	9.81	m/s^2
r_{ms}	Radius of main shield	0.155	m
r_{ts}	Radius of tail shield	0.10	m

Similarly, we can describe the motion of the beam around vertical axis as figure below

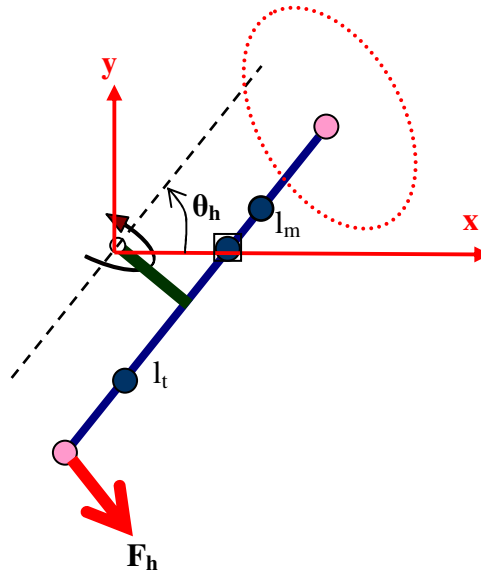


Figure 2.8 Top view of TRMS

From Newton's second law and the parameters in the Table 2:

$$M_h = J_h \frac{d^2 \theta_h}{dt^2}$$

$$M_h = \sum_{i=1}^3 M_{hi}, J_h = \sum_{i=1}^8 J_{hi}$$

Moments of forces (M_{h1}) applied to the beam and making it rotate around vertical axis

$$M_{h1} = l_m F_h(w_t) \cos \theta_v$$

Moment of friction (M_{h2}) depends on the angular velocity of the beam around the vertical axis.

$$M_{h2} = -\omega_h k_h$$

According to Figure 2.7 we can calculate components of the moment of inertia relative to the horizontal axis.

$$J_{h1} = \frac{m_m}{3} (l_m \cos \theta_v)^2$$

$$J_{h2} = \frac{m_t}{3} (l_t \cos \theta_v)^2$$

$$J_{h3} = \frac{m_b}{3} (l_b \sin \theta_v)^2$$

$$J_{h4} = m_{tr} (l_t \cos \theta_v)^2$$

$$J_{h5} = m_{mr} (l_m \cos \theta_v)^2$$

$$J_{h6} = m_{cb} (l_{cb} \sin \theta_v)^2$$

$$J_{h7} = \frac{m_{ts}}{2} r_{ts}^2 + m_{ts} (l_t \cos \theta_v)^2$$

$$J_{h8} = m_{ms} r_{ms}^2 + m_{tm} (l_m \cos \theta_v)^2$$

Or in compact form

$$J_h = D \cos^2 \theta_v + E \sin^2 \theta_v + F$$

Where D, E, F is constant:

$$D = \frac{m_b}{3} l_b^2 + m_{cb} l_{cb}^2$$

$$E = \left(\frac{m_m}{3} + m_{mr} + m_{ms} \right) l_m^2 + \left(\frac{m_t}{3} + m_{tr} + m_{ts} \right) l_t^2$$

$$F = m_{ms} r_{ms}^2 + \frac{m_{ts}}{2} r_{ts}^2$$

Using all above equations we find:

$$\frac{dS_v}{dt} = \frac{d\theta_v^2}{dt^2} = \frac{l_m F_v (w_m) - \omega_v k_v + g((A - B) \cos \theta_v - C \sin \theta_v) - \frac{1}{2} \omega_h^2 (A + B + C) \sin 2\theta_v}{J_v} \quad (3)$$

$$\frac{d\theta_v}{dt} = \omega_v \quad , \quad \omega_v = S_v + \frac{J_{tr} \omega_t}{J_v} \quad (4)$$

In horizontal axis we have:

$$\frac{dS_h}{dt} = \frac{d\theta_h^2}{dt^2} = \frac{l_t F_h(w_t) \cos \theta_v - \omega_h k_h}{J_h} \quad (5)$$

And

$$\frac{d\theta_h}{dt} = \omega_h \quad , \quad \omega_h = S_h + \frac{J_{mr} w_m \cos \theta_v}{J_h} = S_h + \frac{J_{mr} w_m \cos \theta_v}{D \sin^2 \theta_v + E \cos^2 \theta_v + F} \quad (6)$$

Table 2.2 TRMS model parameters in horizontal axis

Symbol	Parameters	Values	Units
M_h	Total moment of force in horizontal plane		Nm
M_{h1}	Moments of gravity force		Nm
M_{h2}	Moments of propulsive force		Nm
M_{h3}	Moment of the centrifugal force		Nm
J_h	Sum of moments of inertia relative to the horizontal axis		kg- m^2
J_{h1}	Moments of inertia of the main DC-motor with main rotor		kg- m^2
J_{h2}	Moment of inertia of main part of the beam		kg- m^2
J_{h3}	Moment of inertia of the counter weight		kg- m^2
J_{h4}	Moment of inertia of the counter-weight beam		kg- m^2
J_{h5}	Moment of inertia of the tail motor with the tail rotor		kg- m^2
J_{h6}	Moment of inertia of the tail part of the beam		kg- m^2
J_{h7}	Moment of inertia of the main shield		kg- m^2
J_{h8}	Moment of inertia of the tail shield		kg- m^2
k_c	Cross reaction momentum gain	-0.2	
T_0	Cross reaction momentum parameter	3.5	
T_p	Cross reaction momentum parameter	2	
T_h	Tail motor denominator parameter	1.1	
S_v	Angular momentum in vertical plane for the beam		
S_h	Angular momentum in horizontal plane for the beam		

3. SIMULATOR OF TRMS SYSTEM

In this chapter the implementation of the simulation model for the TRMS system is described using the mathematical model derived using physical laws presented in chapter 2.

The implementation of this simulation model will be done using the Matlab/Simulink environment.

The Matlab and Simulink environment are integrated into one entry, and thus we can analyse, calculate, simulate and revise our models in either environment at any point. We invoke Simulink within from Matlab.

To control this system we have to do simulator and estimate the system by using Matlab tools. This simulator consists of 2 parts: Simulink and LPV2 model.

3.1 Simulink TRMS system

The TRMS system uses the program Matlab-Simulink. As we know this system is a non-linear mechanical system. The mathematical model that we found in chapter 2:

$$\frac{dS_v}{dt} = \frac{d\theta_v^2}{dt^2} = \frac{l_m F_v(w_m) - \omega_v k_v + g((A - B) \cos \theta_v - C \sin \theta_v) - \frac{1}{2} \omega_h^2 (A + B + C) \sin 2\theta_v}{J_v}$$

$$\frac{d\theta_v}{dt} = \omega_v$$

$$\omega_v = S_v + \frac{J_r \omega_l}{J_v}$$

$$\frac{dS_h}{dt} = \frac{d\theta_h^2}{dt^2} = \frac{l_t F_h(w_t) \cos \theta_v - \omega_h k_h}{J_h}$$

$$\frac{d\theta_h}{dt} = \omega_h$$

$$\omega_h = S_h + \frac{J_{mr} w_m \cos \theta_v}{J_h} = S_h + \frac{J_{mr} w_m \cos \theta_v}{D \sin^2 \theta_v + E \cos^2 \theta_v + F}$$

From the equations by using block diagram to simulate this system with the inputs and outputs. Using block diagrams is a pictorial presentation of the functions performed by each

component and of the flow of signals. The inputs are the voltages u_t , u_r of the tail rotor and main rotor, the outputs are angular velocity and position of two directions. All the variables and dynamic equations are compacted in the subsystem.

The purpose of this Simulink shows us clearly about the movements of TRMS system and applies it to estimate the parameters using to control real helicopter. It means using the identification tool box in Matlab that we will do detailed in the next chapter. The outputs of the system are connected with the horizontal input and motor's speed to see how the system works.

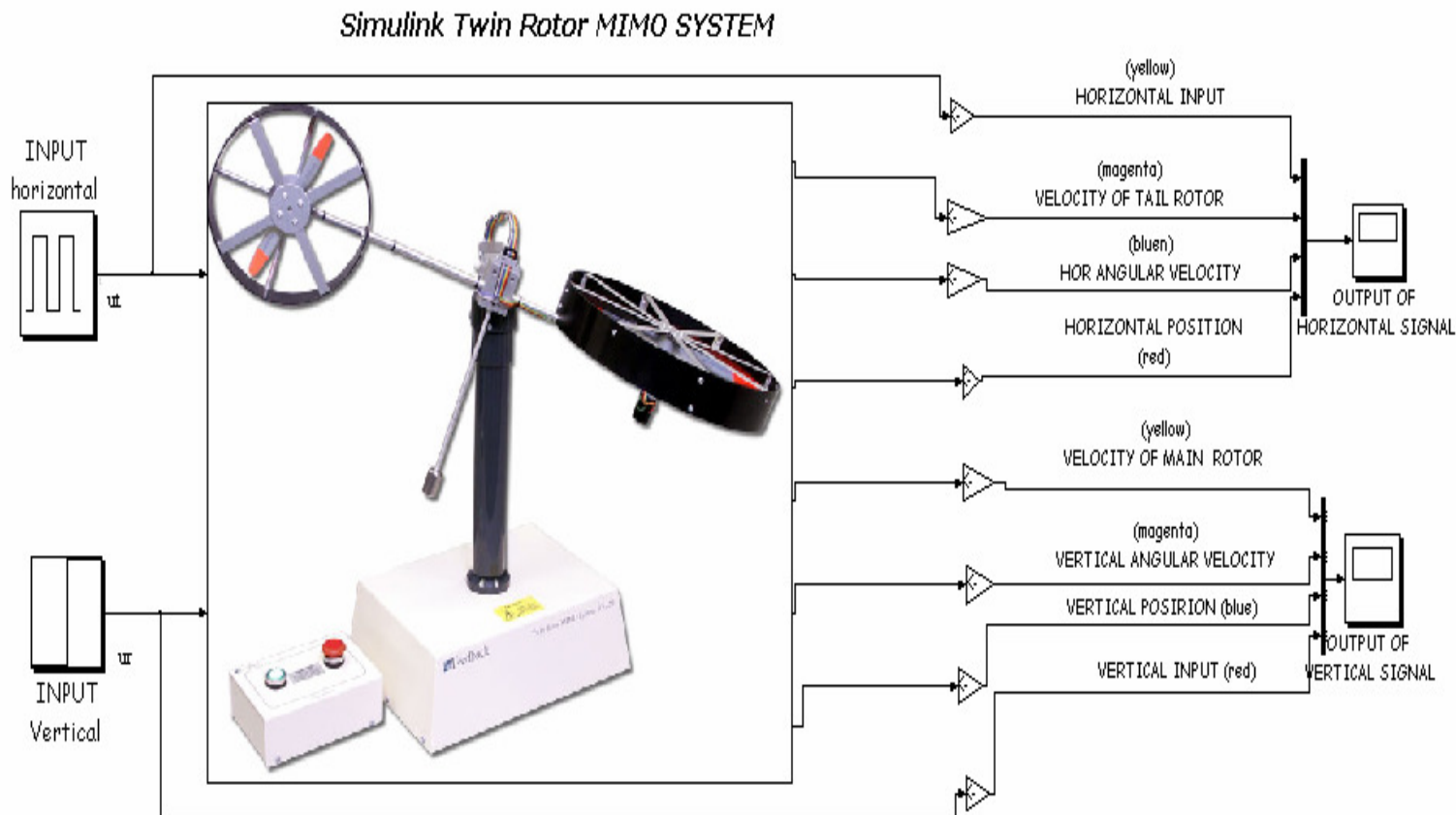


Figure 3.1 Simulink in Matlab of TRMS system

There are two horizontal and vertical inputs entering in subsystem and extracting the outputs by scope of Simulink. Here four outputs are gathered together to see the whole of system's behaviour.

All equations in the subsystem using values are from the model in laboratory and are done by experiments. The parameters are taken from the equipment to model its behaviour. Figure 3.2 following describes the subsystem. The inputs u_r , u_t voltages supply to tail rotor and main rotor. The forces of two rotors can change by controlling the inputs.

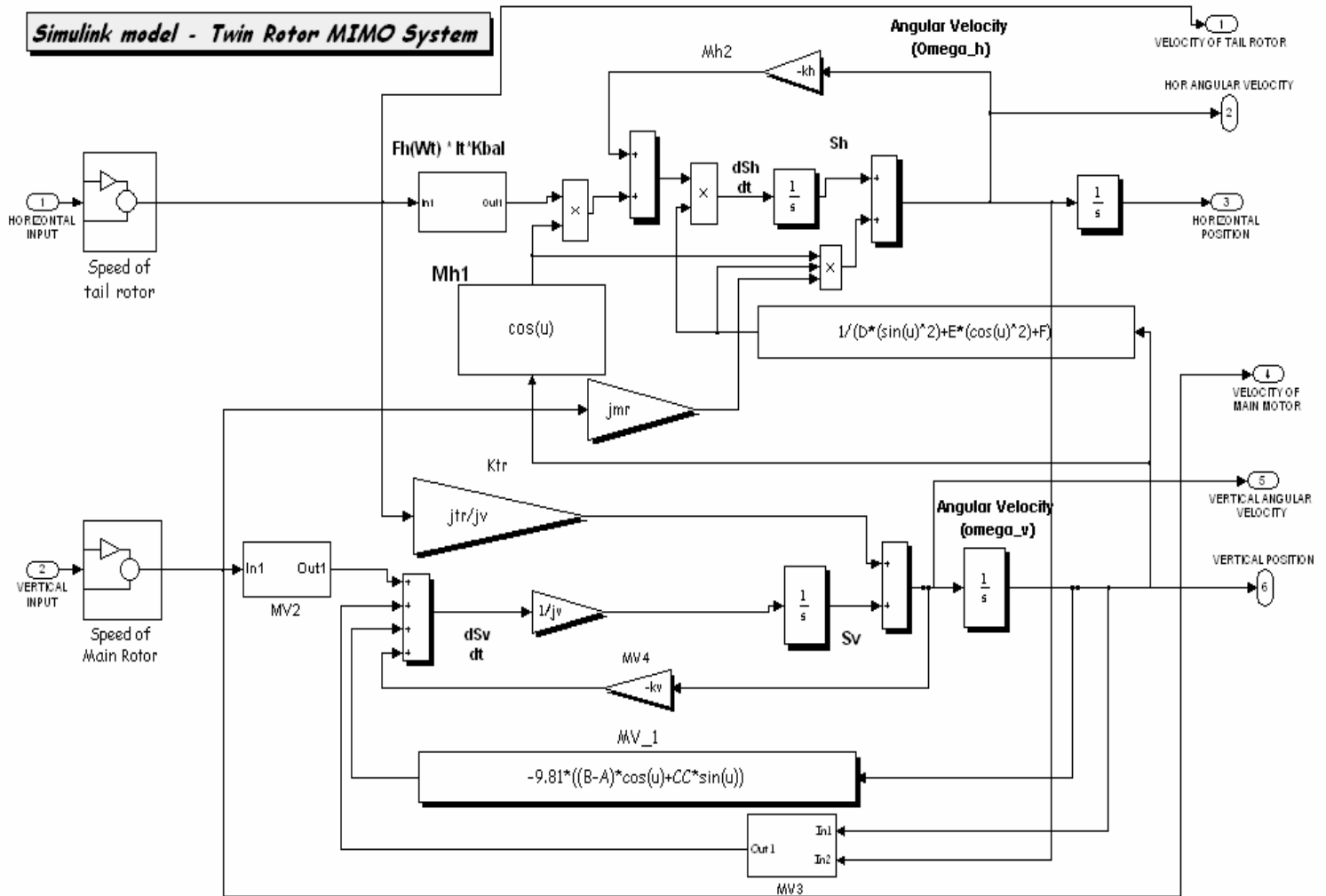


Figure 3.2 Subsystem of TRMS

The rotor's characteristics that we analysed to generate the forces is a polynomial depending on the signal inputs and it is presented in speed of tail rotor and main rotor subsystem. Base on the equations of TRMS we use the block diagram to have figure above.

3.2 Some basic concepts of mathematical models for control

To control firstly we are sure to work by modelling and identifying throughout mathematical model. A mathematical model of a system is defined as a set of equations that represents the dynamics of the system accurately or, at least, fairly well. The dynamics of many systems, whether they are mechanical, electrical, thermal, economic, biological, and so on, may be obtained by using physical laws governing a particular system, for example, Newton's laws for mechanical systems...Deriving the reasonable mathematical model is the most important part of the entire analysis.

Mathematical models may assume many different forms. Depending on the particular system and particular circumstances, one mathematical model may be better suited than other

models. It is advantageous to use state-space representation. We will take into account some definitions in the mathematical model.

3.2.1. State

State of a dynamic system is the smallest set of the variable (called state variables) such that the knowledge of these variables at $t=t_0$ together with the knowledge of the input for $t \geq t_0$ completely determines the behaviour of the system for any time $t \geq t_0$.

The concept of the state is by no means limited to physical. It is applicable to biological systems, economic systems, and others.

3.2.2 State variables

The state variables of a dynamic system are the variables making up the smallest set of variables that determine the state of the dynamic system. If at least n variables x_1, x_2, \dots, x_n are needed to completely describe the behaviour of the dynamic system (so that once the input is given for $t \geq t_0$ and the initial state at $t = t_0$ is specified, the future state of the system is completely determined), then such n variables are a set of state variables.

The state variables need not be physically measurable and observable quantities. Variables that do not represent physical quantities and those that are neither measurable nor observable can be chosen as state variables.

3.2.3. State vector

If n state variables are needed to completely describe the behaviour of a given system, then these n state variables can be considered the n components of a vector x . Such a vector is called a state vector. A state vector is thus a vector. A state vector is thus a vector that determines uniquely the system $x(t)$ for any time $t \geq t_0$, once the state at $t=t_0$ is given and the input $u(t)$ for $t \geq t_0$ is specified.

3.2.4 State space

The n dimensional space whose coordinate axes consist of the x_1 axis, x_2 axis, ..., x_n axis is called a state space. Any state can be represented by a point in the state space.

3.2.5 State space equations

In state space analysis we are concerned with three types of variables that are involved in the modelling of the dynamic systems: input variables, output variables, and state variables.

The dynamic system must involve elements that memorize the values of the input for $t \geq t_0$. Since integrators in a continuous time control system serve as memory devices, the outputs of such integrators can be considered as the variables that define the internal state of the dynamic system. Thus the outputs of integrators serve as state variables. The number of state variables to completely define the dynamics of the system is equal to the number of integrators involved in the system.

Assume that a multiple-input-multiple-output system involves n integrators. Assume also there are r inputs $u_1(t), u_2(t), \dots, u_r(t)$ and m outputs $y_1(t), y_2(t), \dots, y_m(t)$. Define n outputs of the integrators as state variables: $x_1(t), x_2(t), \dots, x_n(t)$. Then the system may be described by

$$\begin{aligned} \dot{x}_1(t) &= f_1(x_1, x_2, \dots, x_m; u_1, u_2, \dots, u_r; t) \\ \dot{x}_2(t) &= f_2(x_1, x_2, \dots, x_m; u_1, u_2, \dots, u_r; t) \\ &\dots\dots\dots \\ \dot{x}_n(t) &= f_n(x_1, x_2, \dots, x_m; u_1, u_2, \dots, u_r; t) \end{aligned} \quad (4.1)$$

The outputs $y_1(t), y_2(t), \dots, y_m(t)$ of the system may be given by

$$\begin{aligned} y_1(t) &= g_1(x_1, x_2, \dots, x_m; u_1, u_2, \dots, u_r; t) \\ y_2(t) &= g_2(x_1, x_2, \dots, x_m; u_1, u_2, \dots, u_r; t) \\ &\dots\dots\dots \\ y_m(t) &= g_m(x_1, x_2, \dots, x_m; u_1, u_2, \dots, u_r; t) \end{aligned} \quad (4.2)$$

If we define

$$\begin{aligned} x(t) &= \begin{bmatrix} x_1(t) \\ x_2(t) \\ \dots \\ x_n(t) \end{bmatrix}, f(x, u, t) = \begin{bmatrix} f_1(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \\ f_2(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \\ \dots \\ f_n(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \end{bmatrix} \\ y(t) &= \begin{bmatrix} y_1(t) \\ y_2(t) \\ \dots \\ y_m(t) \end{bmatrix}, g(x, u, t) = \begin{bmatrix} g_1(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \\ g_2(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \\ \dots \\ g_m(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \end{bmatrix}, u(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \\ \dots \\ u_r(t) \end{bmatrix} \end{aligned} \quad (4.3)$$

Then from (4.1) and (4.2) become

$$\begin{aligned} \dot{x}(t) &= f(x, u, t) \\ y(t) &= g(x, u, t) \end{aligned} \quad (4.4)$$

The 1st equation (4.4) is the state equation and the second is the output equation. If they are linearized about the operating point state, then we have the following linearized state equation and output equation:

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t) \\ y(t) &= C(t)x(t) + D(t)u(t) \end{aligned} \quad (4.5)$$

Where $A(t)$ is called the state space matrix, $B(t)$ the input matrix, $C(t)$ the output matrix, and $D(t)$ the direct state transmission matrix.

3.3 Linear Parameter Varying (LPV) model for controlling the TRMS system

Most of the existing control techniques are based in linear models. However, as we know the above equations of the TRMS system are non-linear. To solve these we have a lot of methods such as: perturbation methods, harmonic balance method, and numerical integration methods.

- Perturbation methods: base on the hypothesis that the non-linearity is small, they allow to transform the system into several linear systems, for which an analytical solution is obtained.
- Harmonic balance method: the solution is expanded in a truncated Fourier series, transforming the problem into an algebraic one. It's a frequency domain method.
- Numerical integration methods: time domain methods where the time is discrete.

All these methods are useful for simulation but not for control. The idea in this master thesis is to use a model that is linear in the structure but preserves the non-linearity by obtaining how parameters vary with the operating point. This kind of model is known as a linear parameter varying (LPV) model.

An LPV model for the TRMS system can be obtained by rearranging the non-linear equations presented in chapter 2:

$$\begin{aligned}
 \frac{d\theta}{dt} &= \omega_h \\
 \frac{d\omega_h}{dt} &= a_{22}\omega_h + a_{24}\omega_v + a_{25}u_{tt} + a_{26}u_{rr} + b_{22}u_r \\
 \frac{d\tilde{\theta}}{dt} &= \omega_v \\
 \frac{d\omega_v}{dt} &= a_{42}\omega_h + a_{43}\tilde{\theta}_v + a_{44}\omega_v + a_{45}u_{tt} + a_{46}u_{rr} + b_{41}u_t \\
 \frac{du_{tt}}{dt} &= a_{55}u_{tt} + b_{51}u_t \\
 \frac{du_{rr}}{dt} &= a_{66}u_{rr} + b_{61}u_r
 \end{aligned} \tag{4.6}$$

Where

$$a_{22} = \frac{-k_h}{D \sin^2(\tilde{\theta}_v - \theta_0) + E \cos^2(\tilde{\theta}_v - \theta_0) + F}$$

$$a_{24} = \frac{((E - D) \cos^2(\tilde{\theta}_v - \theta_0) - (F + D)) J_{mr} \sin(\tilde{\theta}_v - \theta_0) w_m}{D \sin^2(\tilde{\theta}_v - \theta_0) + E \cos^2(\tilde{\theta}_v - \theta_0) + F}$$

$$a_{25} = \frac{l_t \cos(\tilde{\theta}_v - \theta_0) F_h(u_t)}{D \sin^2(\tilde{\theta}_v - \theta_0) + E \cos^2(\tilde{\theta}_v - \theta_0) + F}$$

$$a_{26} = -\frac{1}{\tau_{mr}} \frac{J_{mr} \cos(\tilde{\theta}_v - \theta_0)}{D \sin^2(\tilde{\theta}_v - \theta_0) + E \cos^2(\tilde{\theta}_v - \theta_0) + F}$$

$$a_{42} = \frac{-\frac{1}{2} (A + B + C) \sin(2(\tilde{\theta}_v - \theta_0)) w_h}{J_v}$$

$$a_{43} = \frac{g((A - B) \cos(\tilde{\theta}_v - \theta_0) - C \sin(\tilde{\theta}_v - \theta_0))}{(\tilde{\theta}_v - \theta_0) J_v}$$

$$a_{44} = -\frac{k_v}{J_v}$$

$$a_{45} = -\frac{1}{\tau_{tr}} \frac{J_{tr}}{J_v}$$

$$a_{46} = \frac{l_m F_v(u_{rr})}{J_v}$$

$$a_{55} = -\frac{1}{\tau_{tr}}$$

$$a_{66} = -\frac{1}{\tau_{mr}}$$

$$b_{22} = \frac{k_{mr}(u_r) J_{mr} \cos(\tilde{\theta}_v - \theta_0)}{\tau_{mr} J_h}$$

$$b_{41} = \frac{k_{tr}(u_t) J_{tr}}{\tau_{tr} J_v}$$

$$b_{51} = \frac{k_{tr}(u_t)}{\tau_{tr}}$$

$$b_{62} = \frac{k_{mr}(u_r)}{\tau_{mr}}$$

All equations of this model can be described by state space (in code file $w = \dot{\omega}$)

$$\begin{bmatrix} \dot{\theta}_h \\ \dot{w}_h \\ \dot{\tilde{\theta}}_v \\ \dot{w}_v \\ \dot{u}_{tt} \\ \dot{u}_{rr} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & a_{22} & 0 & a_{24} & a_{25} & a_{26} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} \\ 0 & 0 & 0 & 0 & a_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & a_{66} \end{bmatrix} \begin{bmatrix} \theta_h \\ w_h \\ \tilde{\theta}_v \\ w_v \\ u_{tt} \\ u_{rr} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & b_{22} \\ 0 & 0 \\ b_{41} & 0 \\ b_{51} & 0 \\ 0 & b_{62} \end{bmatrix} \begin{bmatrix} u_t \\ u_r \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$D = 0$$

The LPV model is implemented as an S-function (named LPV2, see Appendix) of MATLAB/SIMULINK. We will use this model to compare how well approximates the non-linear simulation is. The comparison between LPV2 model and the non-linear system TRMS using MATLAB/SIMULINK is presented in figure 3.3.

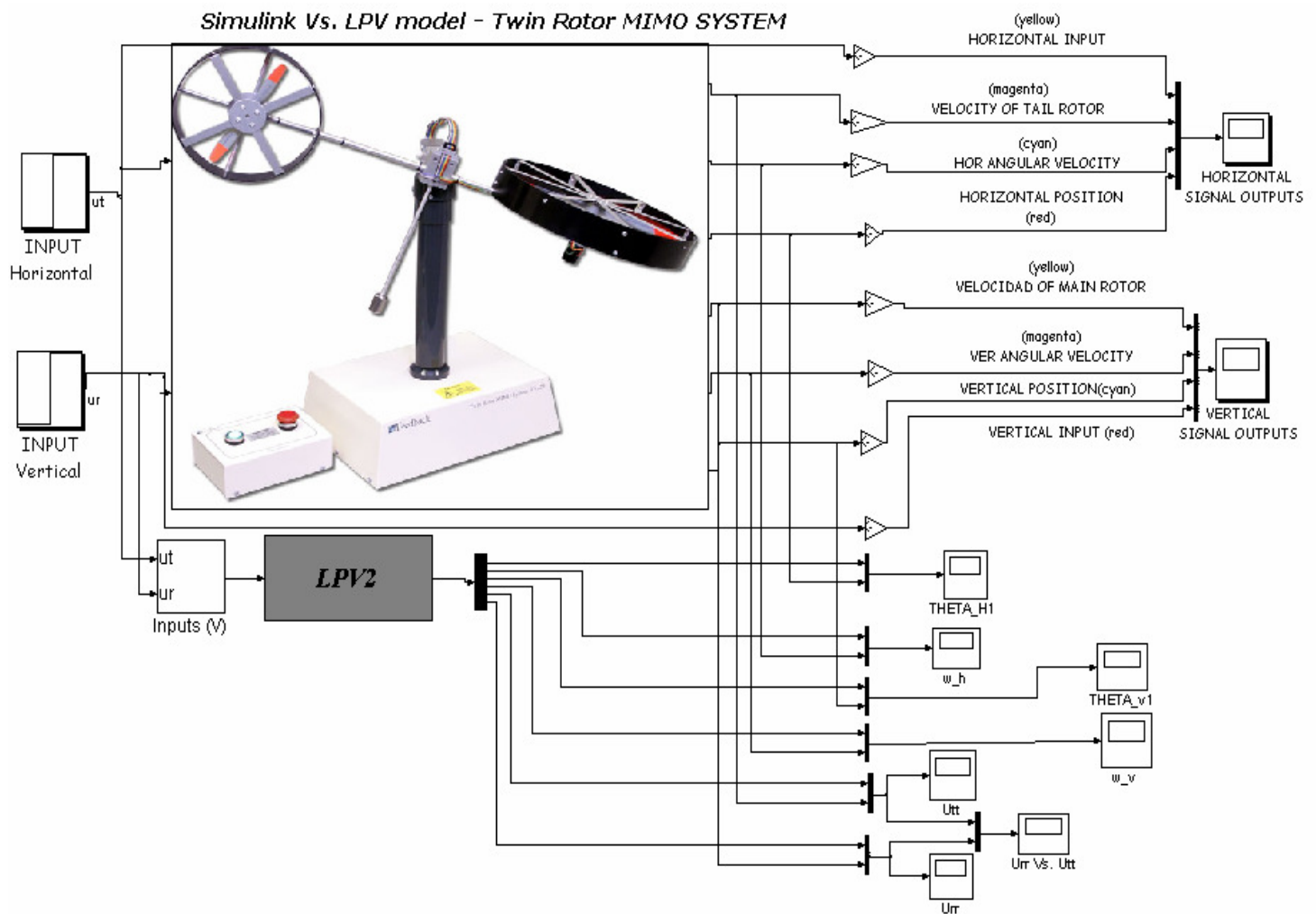


Figure 3.3 Simulator of TRMS

We can know the movements of two directions (horizontal and vertical) in our model by plotting the graphs from scope block (horizontal and vertical signal outputs). From the inputs runs the Simulink and we obtained the results.

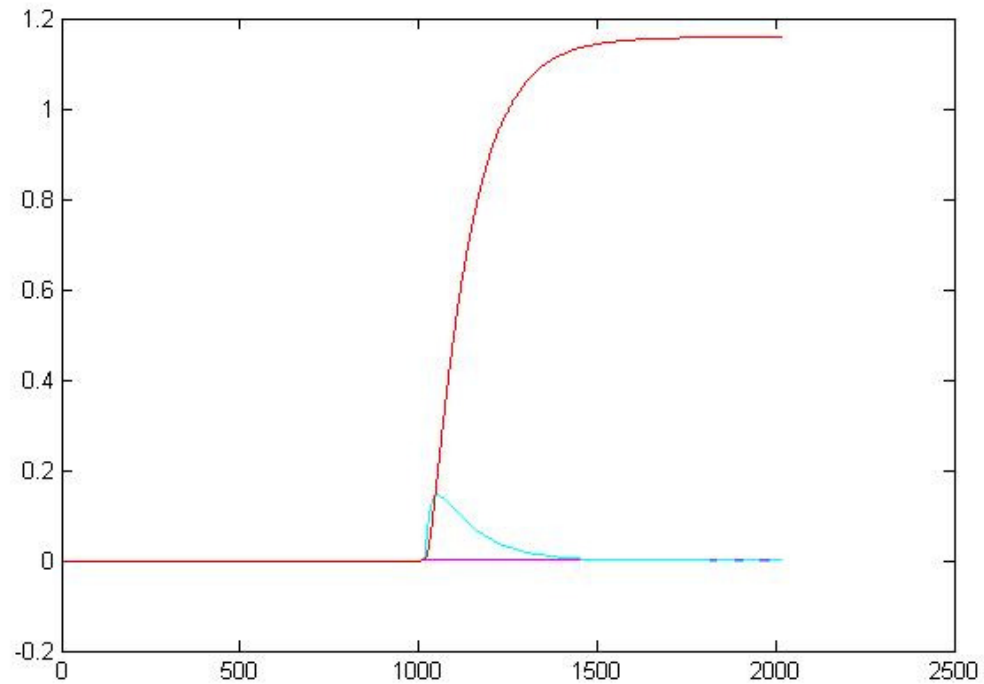


Figure 3.4 Horizontal signal outputs

The signal outputs in horizontal direction show us behaviour of the system. From that we can know our system function at certain range that correspond the inputs (red colour) and will be stable after.

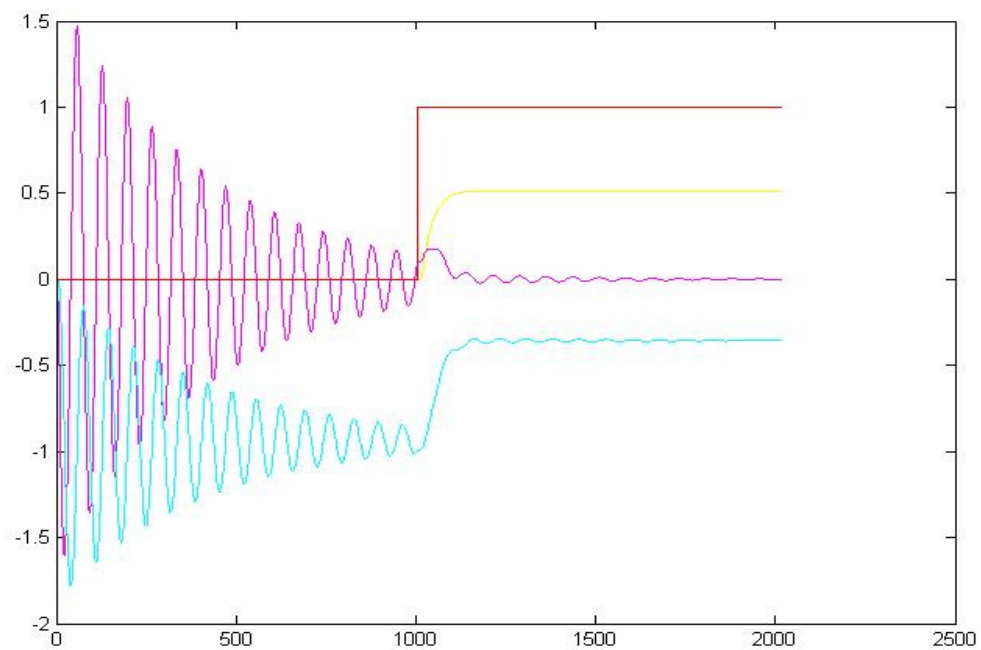


Figure 3.5 Vertical signal outputs

Vertical position (cyan colour) is the movement of the beam in vertical plan. Like horizontal position our system will function around some positions and keep stably at certain points.

The LPV2 model is applied in our system by using state space method that is compared with the non-linear model as figure below.

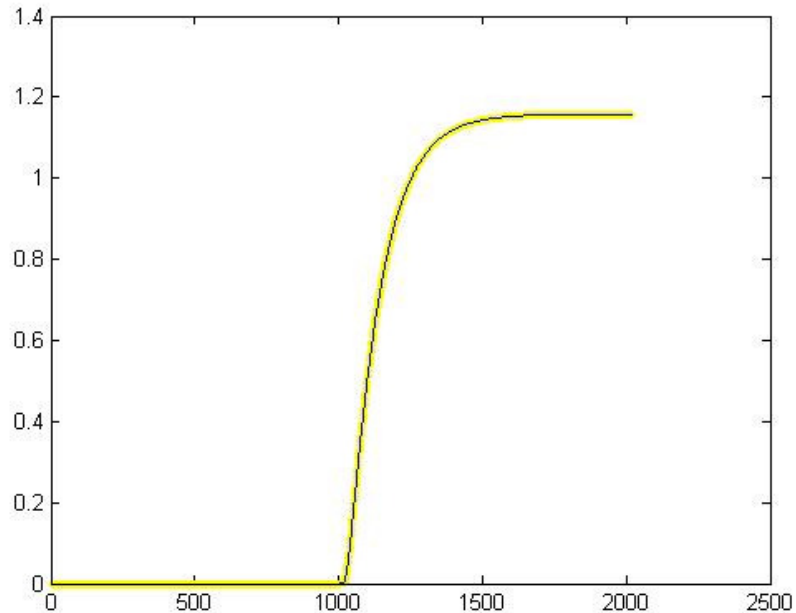


Figure 3.6 Horizontal position of LPV2 and Simulink model

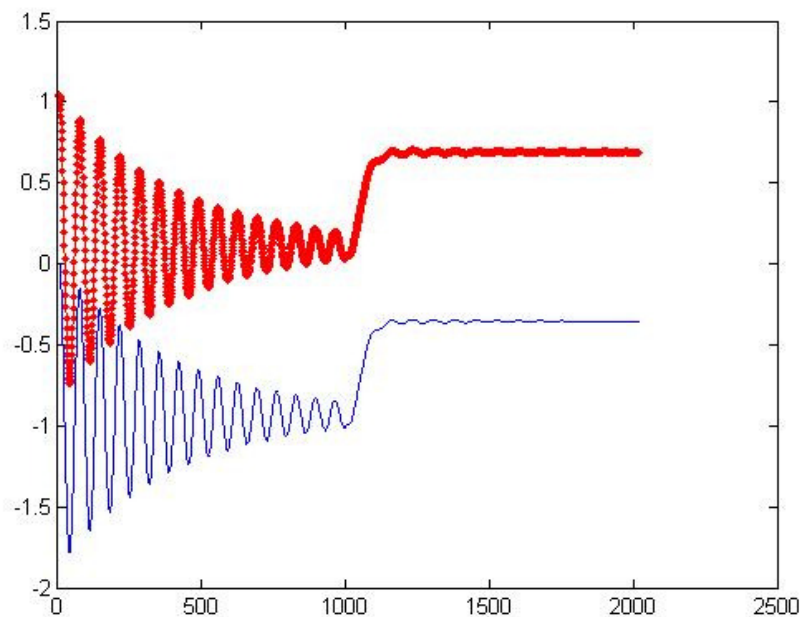


Figure 3.7 Vertical position of LPV2 (red) and Simulink model

In figure 3 there are different values of vertical position because in model LPV2 that divided by 0 so avoiding this problem we change the coordinates. To obtain exactly the vertical horizontal between them we can offset again to get the same result.

Once the simulator for the non-linear and LPV models is finished, we have all the tools to simulate their behaviour. To control the real system we need to calibrate both models using real data. This step will be done in next chapter.

4. CALIBRATION OF TRMS SYSTEM CONTROL MODEL

4.1. Introduction

Calibration of the TRMS systems control model is done by using the identification toolbox in Matlab. Two different control models will be considered:

- the linear parameter varying (LPV) model presented in chapter 3.
- a linear model around a given operating point obtained using linearization

The aim of this chapter is to tune the calibration algorithm using data coming from the helicopter simulation generated using SIMULINK model. Once the calibration algorithm has been validated, it will be applied using real data.

4.2. Calibration of LPV model using MATLAB (pem command)

4.2.1. Pem and mathematical model

To calibrate the LPV model a set of experiments at different operating points will be performed in the TRMS system. Using data coming from these experiments and parameter estimation algorithms from Identification Toolbox in MATLAB parameters will be estimated. Once the parameters for each operating point have been obtained the variation of their value can be interpolated obtaining the LPV model.

The parameter estimation algorithm used in MATLAB is the Pem (prediction-error minimization) algorithm. This algorithm allows to estimate low-order continuous-time process models, linear state-space, and polynomial models. Pem also refine initial parameter estimates for all linear and nonlinear parametric models.

Pem uses optimization to minimize the cost function that is a function of weighted sum of squares of the errors, defined as follows for scalar outputs:

$$V_N(G, H) = \sum_{t=1}^N e^2(t)$$

Where $e(t)$ is the difference between the measured output and the predicted output of the model. For a linear model, this error is defined by the following equation:

$$e(t) = H^{-1}(q)[y(t) - G(q)u(t)]$$

$e(t)$ is a vector and the cost function $V_N(G, H)$ is a scalar value. The subscript N indicates that the cost function is a function of the number of data samples and becomes more accurate for larger values of N. $G(t)$ is a operator that takes the input to the output and captures the system dynamics or G is a transfer function between $u(t)$ and $y(t)$. H is an operator that

describes the properties of the additive output disturbance and is the noise model. For multioutput models, the previous equation is more complex.

The LPV model of TRMS is presented in chapter 3 with the equations rewriting (see detail in index):

$$\begin{bmatrix} \dot{\theta}_h \\ \dot{w}_h \\ \dot{\tilde{\theta}}_v \\ \dot{w}_v \\ \dot{u}_{tt} \\ \dot{u}_{rr} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & a_{22} & 0 & a_{24} & a_{25} & a_{26} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} \\ 0 & 0 & 0 & 0 & a_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & a_{66} \end{bmatrix} \begin{bmatrix} \theta_h \\ w_h \\ \tilde{\theta}_v \\ w_v \\ u_{tt} \\ u_{rr} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & b_{22} \\ 0 & 0 \\ b_{41} & 0 \\ b_{51} & 0 \\ 0 & b_{62} \end{bmatrix} \begin{bmatrix} u_t \\ u_r \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$D = 0$$

Where:

$$a_{22} = \frac{-k_h}{D \sin^2(\tilde{\theta}_v - \theta_0) + E \cos^2(\tilde{\theta}_v - \theta_0) + F}$$

$$a_{24} = \frac{((E - D) \cos^2(\tilde{\theta}_v - \theta_0) - (F + D)) J_{mr} \sin(\tilde{\theta}_v - \theta_0) w_m}{D \sin^2(\tilde{\theta}_v - \theta_0) + E \cos^2(\tilde{\theta}_v - \theta_0) + F}$$

$$a_{25} = \frac{l_t \cos(\tilde{\theta}_v - \theta_0) F_h(u_{tt})}{D \sin^2(\tilde{\theta}_v - \theta_0) + E \cos^2(\tilde{\theta}_v - \theta_0) + F}$$

$$a_{26} = -\frac{1}{\tau_{mr}} \frac{J_{mr} \cos(\tilde{\theta}_v - \theta_0)}{D \sin^2(\tilde{\theta}_v - \theta_0) + E \cos^2(\tilde{\theta}_v - \theta_0) + F}$$

$$a_{42} = \frac{-\frac{1}{2} (A + B + C) \sin(2(\tilde{\theta}_v - \theta_0)) w_h}{J_v}$$

$$a_{43} = \frac{g((A - B) \cos(\tilde{\theta}_v - \theta_0) - C \sin(\tilde{\theta}_v - \theta_0))}{(\tilde{\theta}_v - \theta_0) J_v}$$

$$a_{44} = -\frac{k_v}{J_v}$$

$$a_{45} = -\frac{1}{\tau_{tr}} \frac{J_{tr}}{J_v}$$

$$a_{46} = \frac{l_m F_v(u_{rr})}{J_v}$$

$$a_{55} = -\frac{1}{\tau_{tr}}$$

$$a_{66} = -\frac{1}{\tau_{mr}}$$

$$b_{22} = \frac{k_{mr}(u_r) J_{mr} \cos(\tilde{\theta}_v - \theta_0)}{\tau_{mr} J_h}$$

$$b_{41} = \frac{k_{tr}(u_t) J_{tr}}{\tau_{tr} J_v}$$

$$b_{51} = \frac{k_{tr}(u_t)}{\tau_{tr}}$$

$$b_{62} = \frac{k_{mr}(u_r)}{\tau_{mr}}$$

From this analysis the structure of the system matrices can be inferred. This structural information will be used during the parameter estimation process using pem algorithm.

4.2.2. Implementation method

- In the first step

we need to calculate the coefficients of matrix A and B for calibrating the model by pem command in Matlab. From the Simulink and all equations of the coefficients (a_{22} , a_{24} ...) we use the block Fcn in Simulink Library Browser to get them.

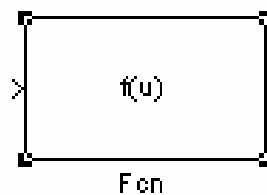


Figure 4.1 Function Fcn

All functions will be calculated easily by this block. They receive the inputs and calculate outputs so we can extract values into Matlab. The simulink connects the blocks for calculation in figure 4.2.

For example, the coefficient a_{22} (output) need to be calculated and the input here is θ . The function of a_{22} will be put in block and will give the values of a_{22} . For other coefficients, we can similarity calculate.

- The second step is estimating state space model with structured parameterization:

- Creating the data file of the model with name Trms (trms.mat). It contains of the input and output values from the workspace.
- Specifying the state space model structure, first define the A, B, C, D matrices in the Matlab workspace.

$$m=idss(A,B,C,D)$$

Where A, B, C specify the fixed values and the initial values for free parameters. We put the calculated values of the coefficients taking from Fcn blocks in matrices.

- After we create the nominal model structure, we must specify which parameters to estimate. To accomplish this, we must edit the structure of the following model properties: As, Bs, Cs, Ds. These structure matrices are properties of the nominal model we constructed and have the same sizes as A, B, C, D respectively. Set a NaN for the coefficient that we want to calibrate.

$$m.As = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & NaN & 0 & NaN & NaN & NaN \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & NaN & NaN & NaN & NaN & NaN \\ 0 & 0 & 0 & 0 & 0 & NaN \end{bmatrix}$$

$$m.Bs = \begin{bmatrix} 0 & 0 \\ 0 & NaN \\ 0 & 0 \\ NaN & 0 \\ NaN & 0 \\ 0 & NaN \end{bmatrix}$$

$$m.Cs = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$m.Ds=0$$

The estimation algorithm only estimates the parameters in A and B have NaN value in As, Bs. Finally, use pem to estimate the model.

$$m=pem(data,m)$$

Data is trms.mat that we constructed initially consisting the inputs and outputs.

In Matlab the calculated values are not really continuous (discrete time) so solving this problem we change the algorithm into discrete time for making suitable values to estimate with sampling time $\Delta t=0.05$.

Equation (4.6) is rewritten:

$$\begin{aligned}\frac{\theta(k+1) - \theta(k)}{\Delta t} &= \omega_h \\ \frac{\omega_h(k+1) - \omega_h(k)}{\Delta t} &= a_{22}\omega_h + a_{24}\omega_v + a_{25}u_{tt} + a_{26}u_{rr} + b_{22}u_r \\ \frac{\tilde{\theta}(k+1) - \tilde{\theta}(k)}{\Delta t} &= \omega_v \\ \frac{\omega_v(k+1) - \omega_v(k)}{\Delta t} &= a_{42}\omega_h + a_{43}\tilde{\theta}_v + a_{44}\omega_v + a_{45}u_{tt} + a_{46}u_{rr} + b_{41}u_t \\ \frac{u_{tt}(t+1) - u_{tt}(t)}{\Delta t} &= a_{55}u_{tt} + b_{51}u_t \\ \frac{u_{rr}(t+1) - u_{rr}(t)}{\Delta t} &= a_{66}u_{rr} + b_{61}u_r\end{aligned}$$

And the state space of the model now is:

$$\begin{bmatrix} \dot{\theta}_h \\ \dot{w}_h \\ \dot{\tilde{\theta}}_v \\ \dot{w}_v \\ \dot{u}_{tt} \\ \dot{u}_{rr} \end{bmatrix} = \begin{bmatrix} 1 & 0.05 & 0 & 0 & 0 & 0 \\ 0 & 1+0.05a_{22} & 0 & 0.05a_{24} & 0.05a_{25} & 0.05a_{26} \\ 0 & 0 & 1 & 0.05 & 0 & 0 \\ 0 & 0.05a_{42} & 0.05a_{43} & 1+0.05a_{44} & 0.05a_{45} & 0.05a_{46} \\ 0 & 0 & 0 & 0 & 1+0.05a_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1+0.05a_{66} \end{bmatrix} \begin{bmatrix} \theta_h \\ w_h \\ \tilde{\theta}_v \\ w_v \\ u_{tt} \\ u_{rr} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0.05b_{22} \\ 0 & 0 \\ 0.05b_{41} & 0 \\ 0.05b_{51} & 0 \\ 0 & 0.05b_{62} \end{bmatrix} \begin{bmatrix} u_t \\ u_r \end{bmatrix}$$

With this algorithm we can estimate our model.

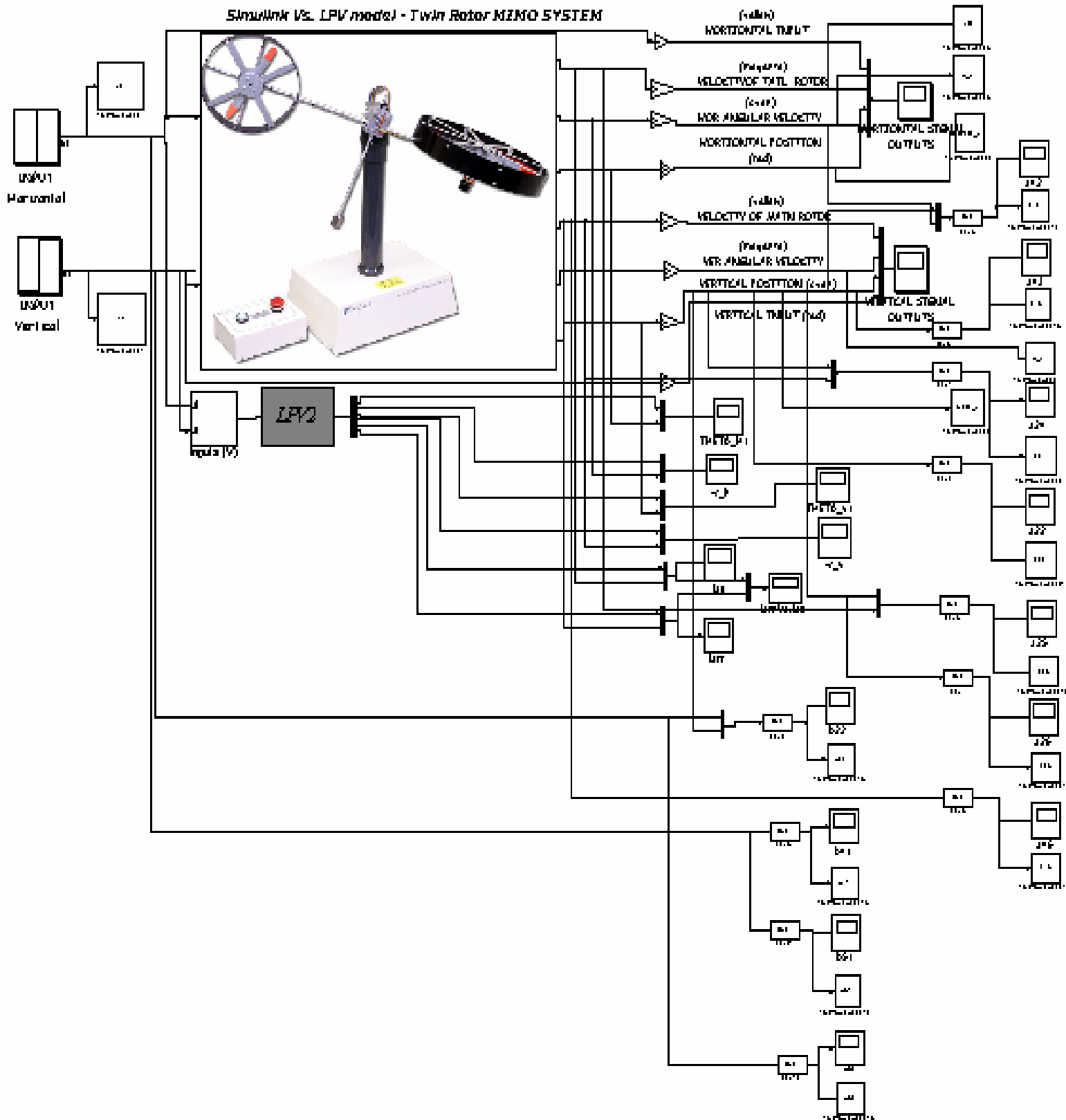


Figure 4.2 Calculating coefficients in state space

This non-linear system is solved by considering linear system around some operating points. In this case we consider 4 operating points to calibrate and find the behaviour of TRMS system.

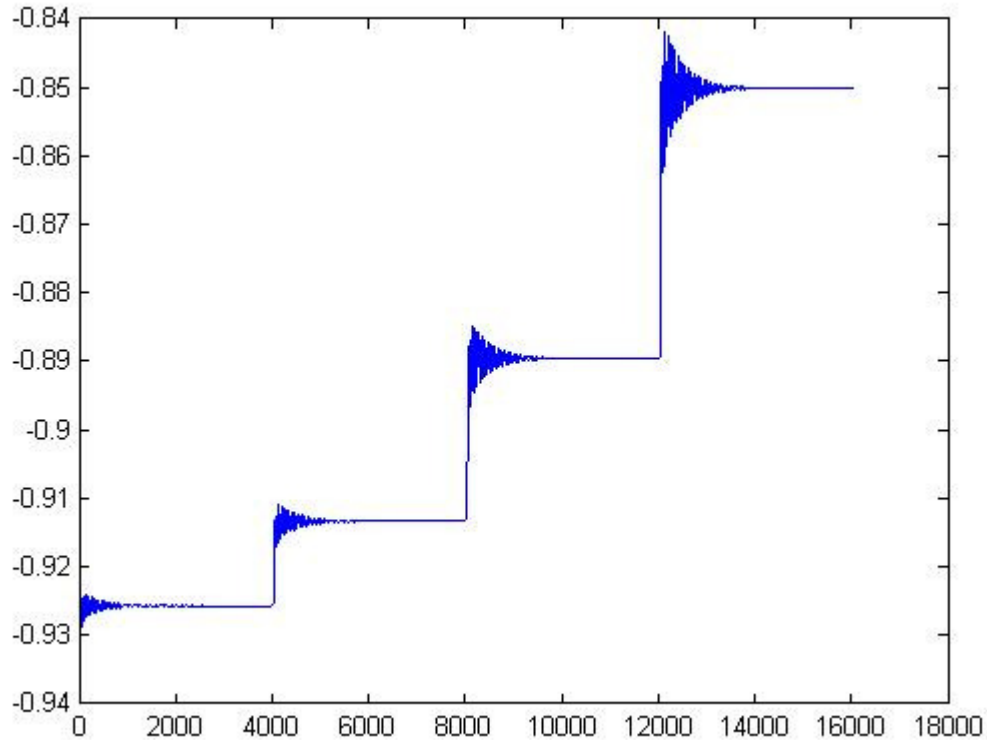


Figure 4.3 Operating points of the system

These steps are taken from 0.001-0.20 divided 4 steps of the vertical input. Now with all known values we will write the codes in Matlab and extract the result.

4.2.3. The results of calibration

Using the pem command extends the Matlab computation environment and let us fit linear and nonlinear mathematical models to input and output data from the dynamic systems. The calculated and estimated results are in the table 4.1.

- The first step 0.001-0.05

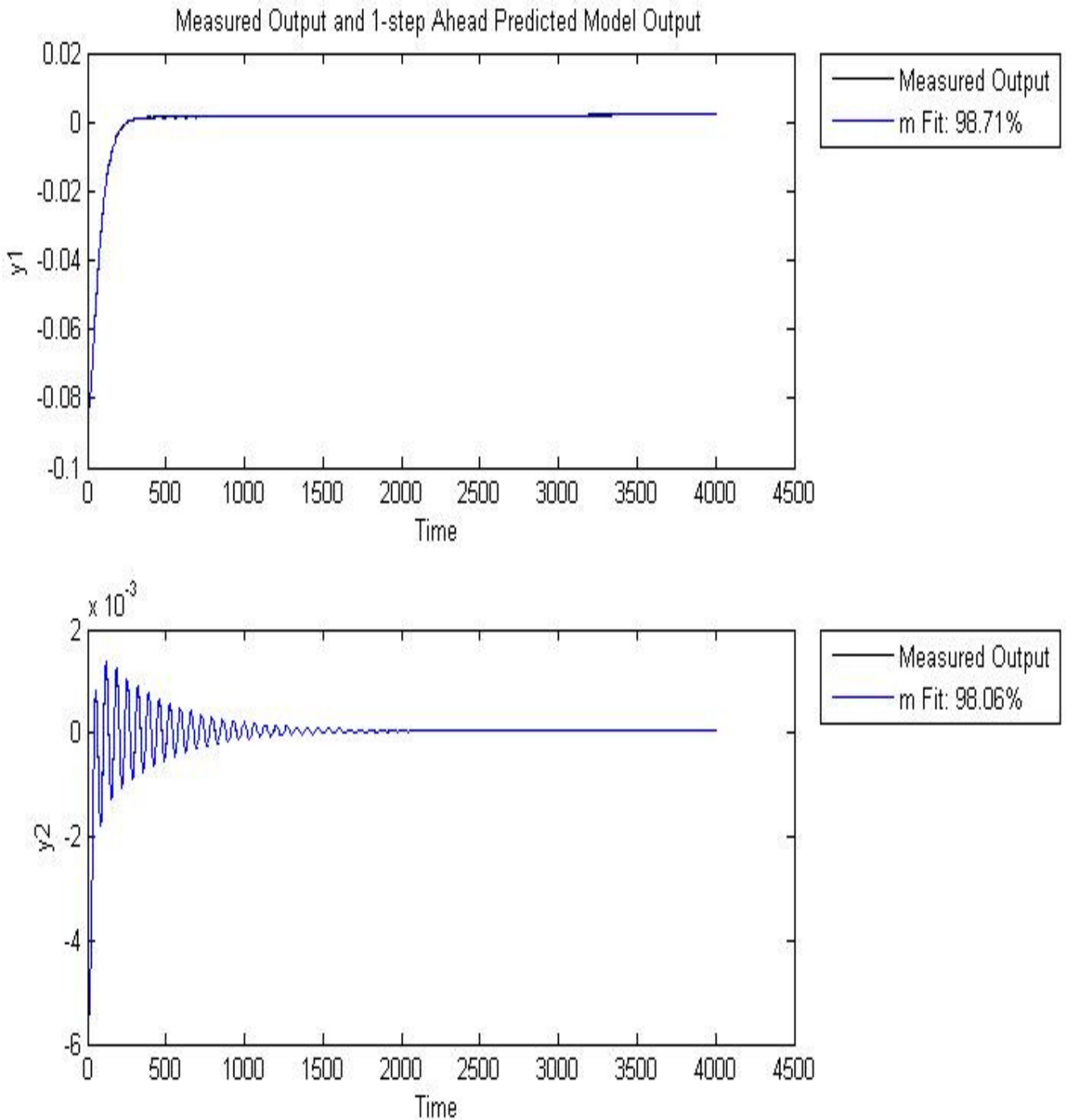


Figure 4.4 The 1st operating point calibration

The preceding plot shows that the model output is in reasonable agreement with the measured output: there is an agreement of 98.71% (horizontal position) and 98.06% (vertical position) percent between the model and the validation data.

- The second step 0.05-0.10

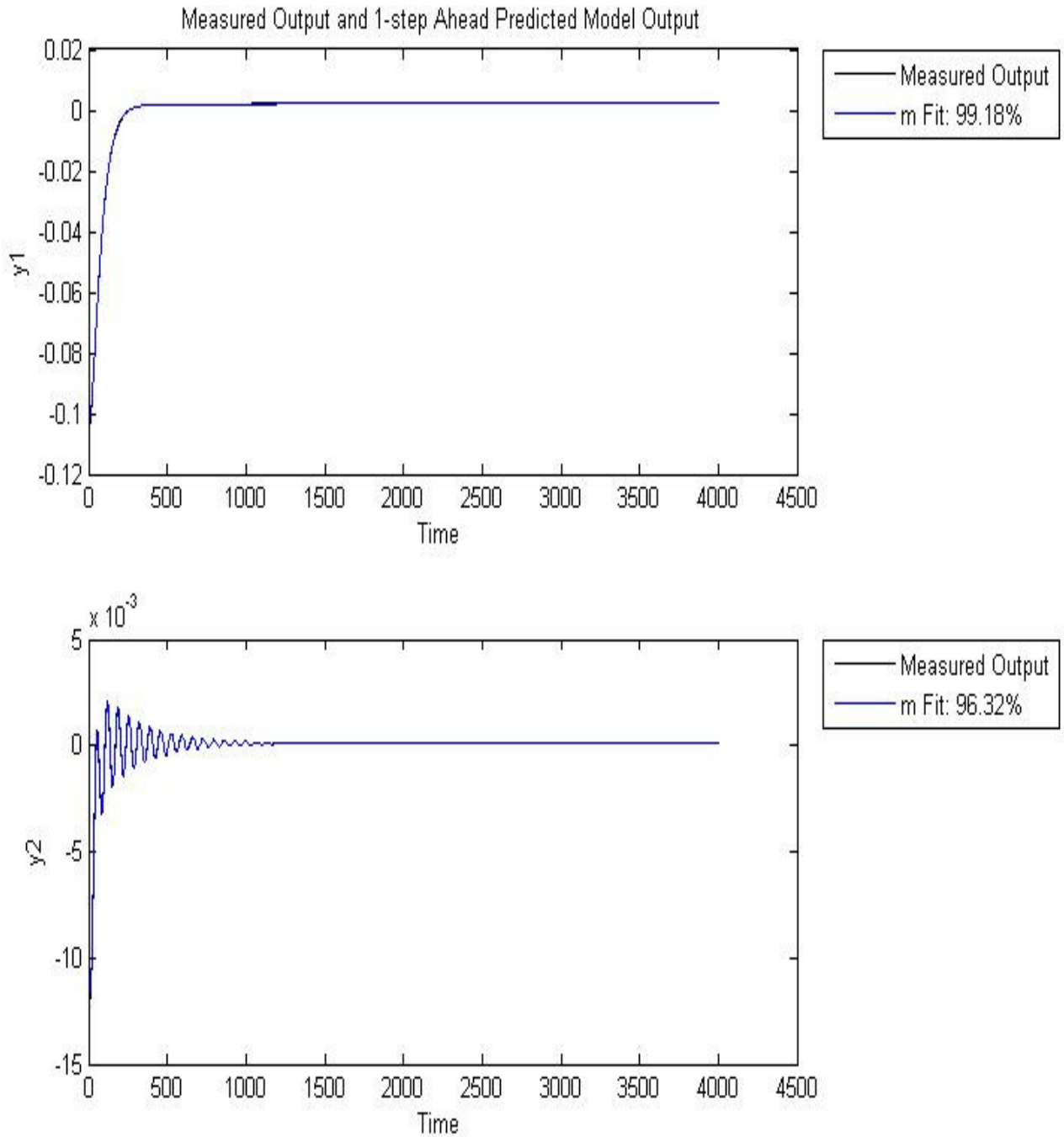


Figure 4.5 The 2nd operating point calibration

The preceding plot shows that the model output is in reasonable agreement with the measured output: there is an agreement of 99.18% (horizontal position) and 96.32% (vertical position) percent between the model and the validation data.

- The 3rd step 0.10-0.15

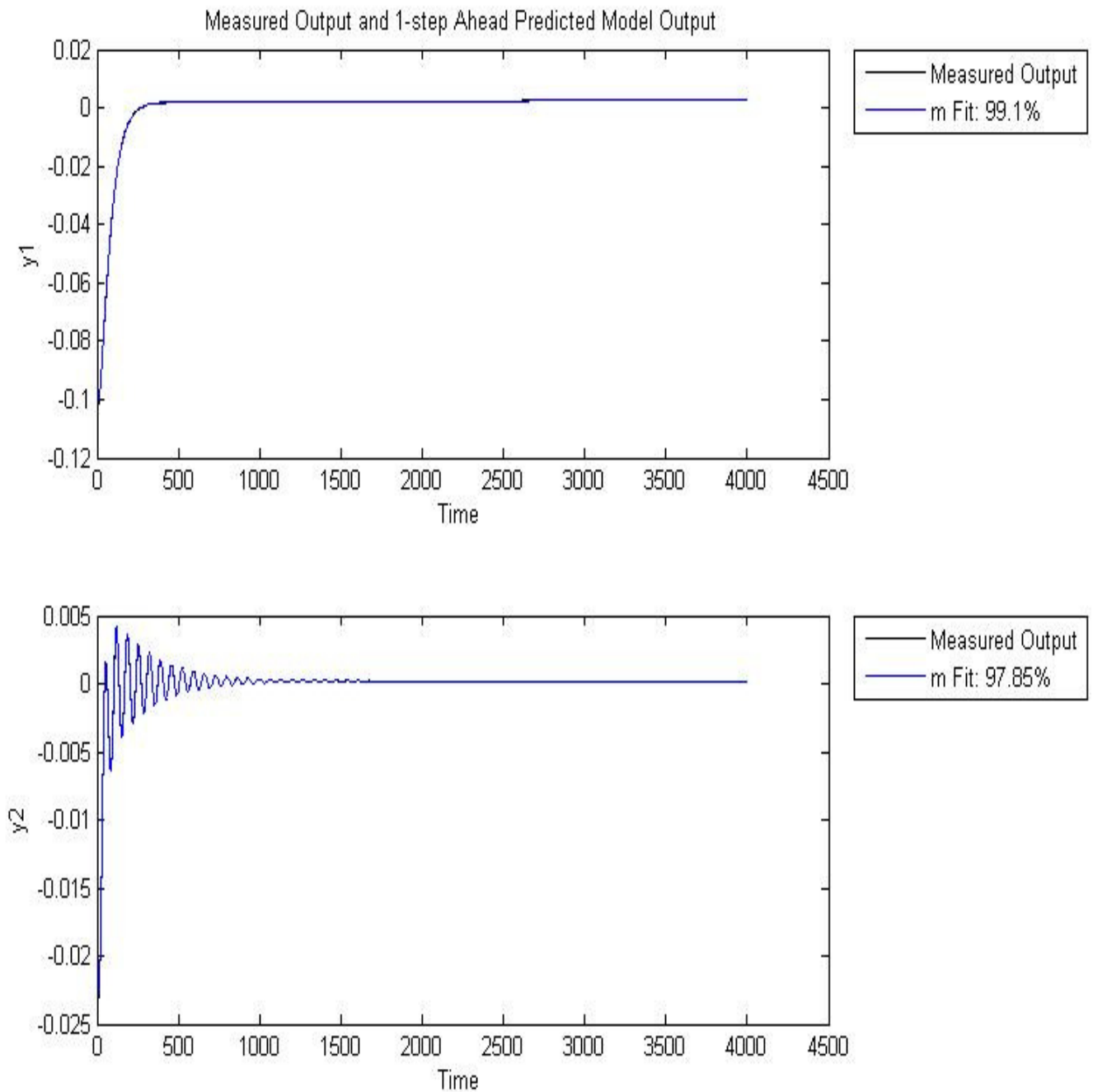


Figure 4.6 The 3rd operating point calibration

The preceding plot shows that the model output is in reasonable agreement with the measured output: there is an agreement of 98.1% (horizontal position) and 97.85% (vertical position) percent between the model and the validation data.

- The 4th step 0.15-0.20

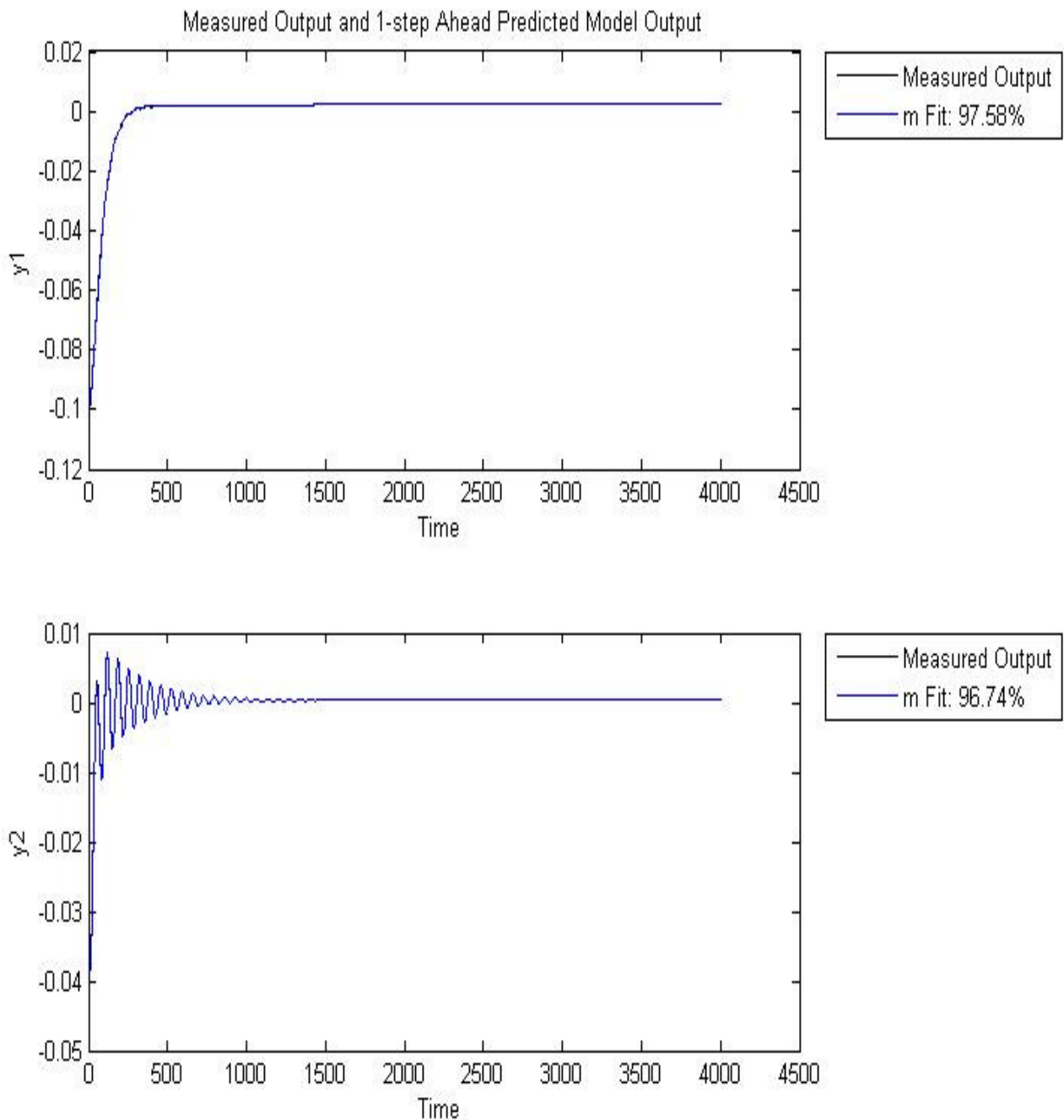


Figure 4.7 The 4th operating point calibration

The preceding plot shows that the model output is in reasonable agreement with the measured output: there is an agreement of 97.58% (horizontal position) and 96.74% (vertical position) percent between the model and the validation data.

Table 4.1 Result of the calibration

Matrices	A						B	
	State space	1	0.05	0	0	0	0	0
	0	$1+0.05a_{22}$	0	$0.05a_{24}$	$0.05a_{25}$	$0.05a_{26}$	0	$0.05b_{22}$
	0	0	1	0.05	0	0	0	0
	0	$0.05a_{42}$	$0.05a_{43}$	$1+0.05a_{44}$	$0.05a_{45}$	$0.05a_{46}$	$0.05b_{41}$	0
	0	0	0	0	$1+0.05a_{55}$	0	$0.05b_{51}$	0
	0	0	0	0	0	$1+0.05a_{66}$	0	$0.05b_{62}$
1 st Operating point	1	0.05	0	0	0	0	0	0
	0	0.98107	0	-0.00042	-86.718	-0.00032	0	0.096487
	0	0	1	0.05	0	0	0	0
	0	-0.00087	-0.05804	0.99511	-7.4081	0.00062	365.05	0
	0	0	0	0	0.68308	0	18.7	0
	0	0	0	0	0	0.96225	0	1.1429
2 nd Operating point	1	0.05	0	0	0	0	0	0
	0	0.98149	0	-0.00138	-88.728	-0.00014	0	0.19788
	0	0	1	0.05	0	0	0	0
	0	-0.00384	-0.05751	0.99508	-12.085	2.26E-03	1317.9	0
	0	0	0	0	0.61856	0	43.854	0
	0	0	0	0	0	0.96425	0	3.3504
3 rd Operating point	1	0.05	0	0	0	0	0	0
	0	0.98161	0	-0.00227	-166.99	-0.00021	0	0.75132
	0	0	1	0.05	0	0	0	0
	0	0.00034	-0.05620	0.99512	-43.589	5.33E-03	9138.2	0
	0	0	0	0	0.27619	0	160.9	0
	0	0	0	0	0	0.96544	0	5.5425
4 th Operating point	1	0.05	0	0	0	0	0	0
	0	0.98254	0	-0.00312	-224.16	-0.00018	0	1.275
	0	0	1	0.05	0	0	0	0
	0	0.58318	-0.02029	1.006	-226.94	1.14E-02	19774	0
	0	0	0	0	0.13892	0	243.16	0
	0	0	0	0	0	0.96667	0	7.665

Table 4.2 Stable points

	Stable points of the positions from simulation	
	θ_v	θ_h
1 st Operating point	-0.92572912253694	0.105774462
2 nd Operating point	-0.91345518281618	0.217876897
3 rd Operating point	-0.88973319573281	0.316503334
4 th Operating point	-0.85044590846964	0.387291297

We will plot some of coefficients in state space to see the parameters function around the operating point range.

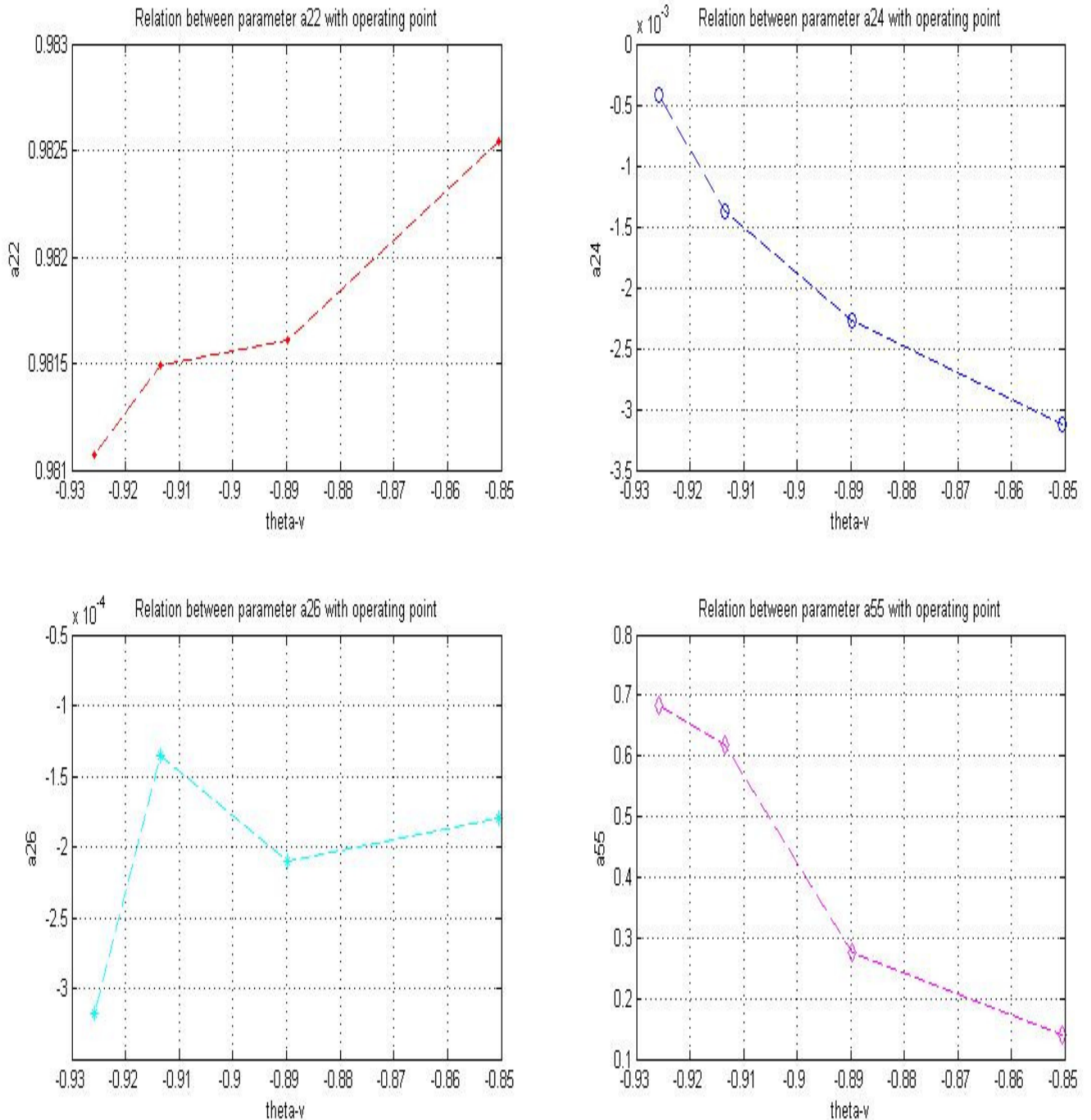


Figure 4.8 Estimated coefficients of matrix A

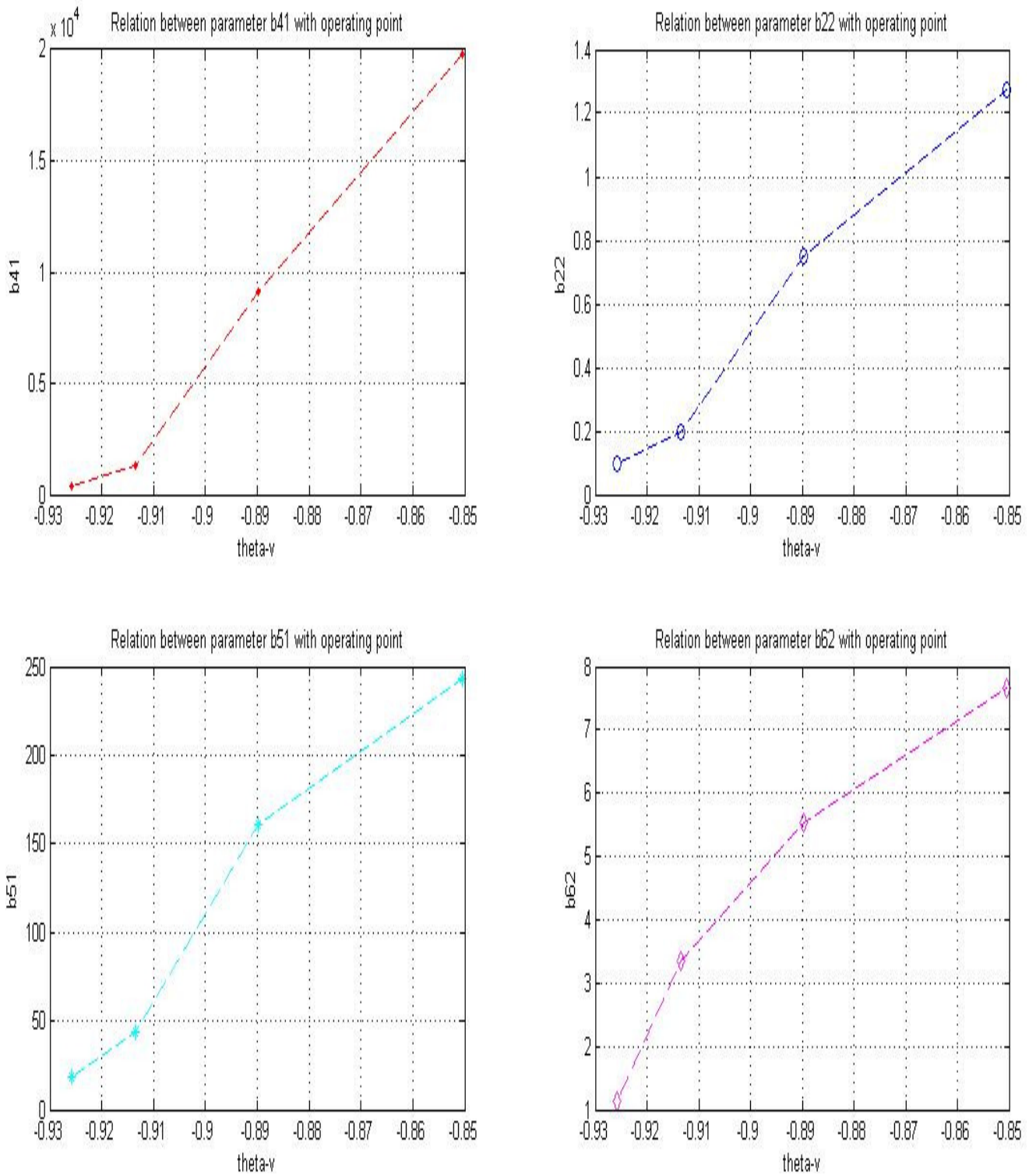


Figure 9.9 Estimated coefficients of matrix B

This model is non-linear that is applied to solve linear method but all coefficients are really non-linear and we can see from the graph they change around the operating points.

Repeating the calibration that is described so far using real data coming from the real TRMS system, physical parameters of the real system can be estimated.

4.3. Calibration of a LPV model coming from linearization toolbox

An alternative way to obtain an LPV model is to linearize the non-linear system around different operating points. This can be automatically done using the linearization toolbox in Matlab/Simulink.

The procedure is as follows: From the window Simulink we choose button tools and linear analysis on tool bar. Matlab will automatically extract the value of the coefficients that we used block to calculate above. After that we realize the calibration.

This operating point is at $t=180$ and it is estimated with 4 steps from 0.001-0.20. The results are in table 4.2.

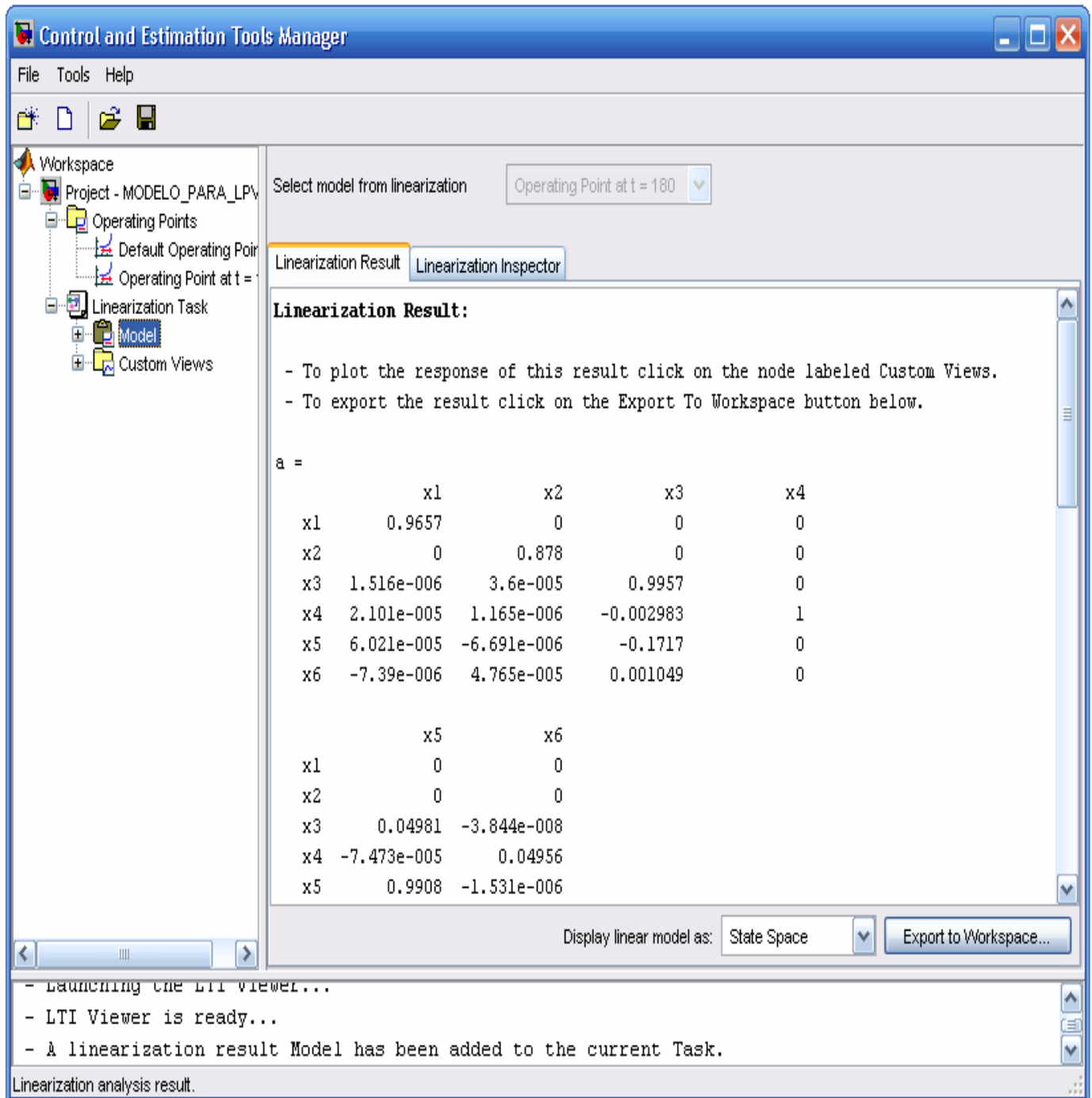


Figure 4.10 Linearization toolbox window

- The first step 0.001-0.05

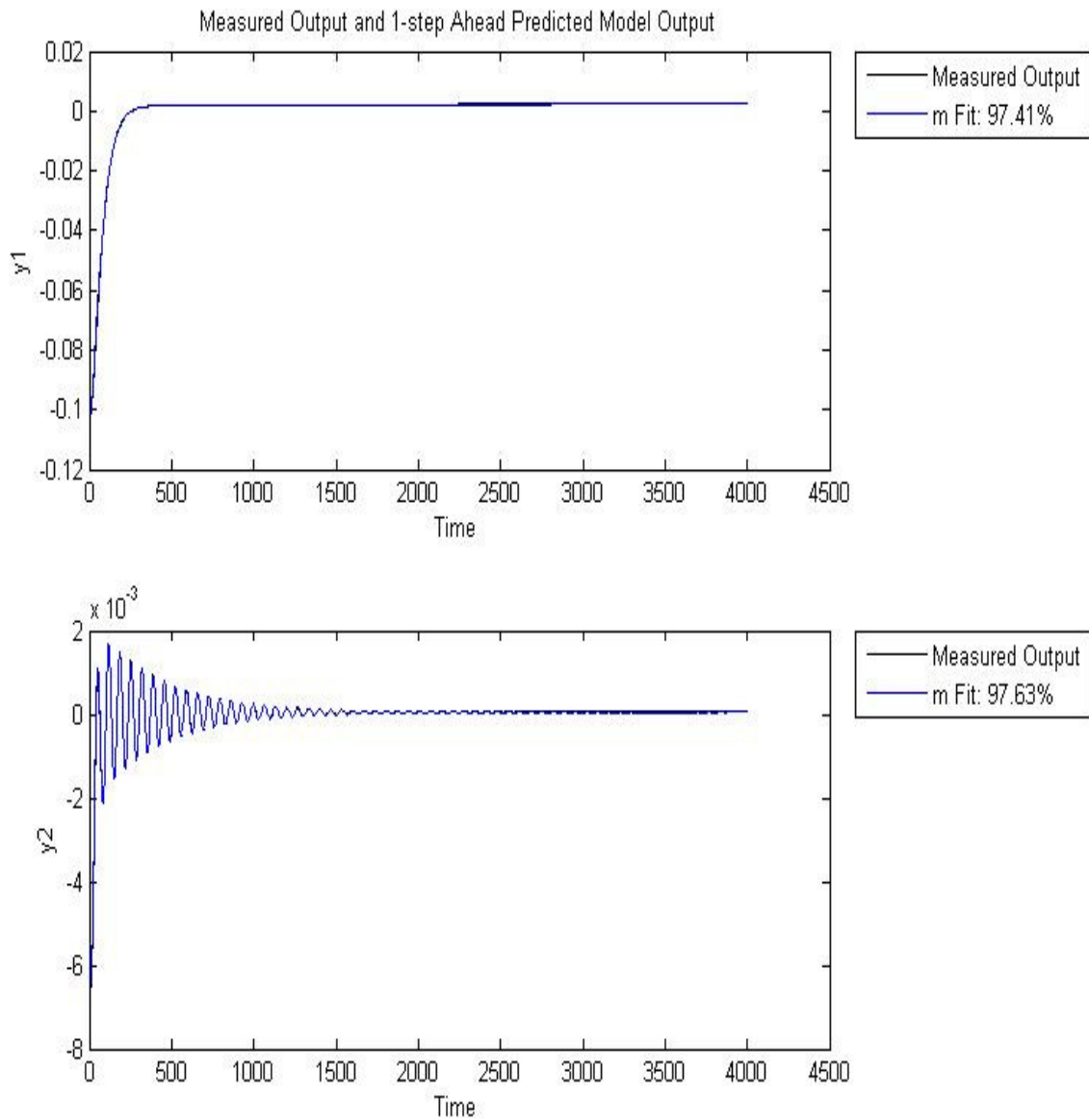


Figure 4.11 The 1st operating point calibration by linearization tool

The measured and estimated values are fit together. Horizontal position is 97.41% and vertical is 97.63%

- The second step 0.05.0.10

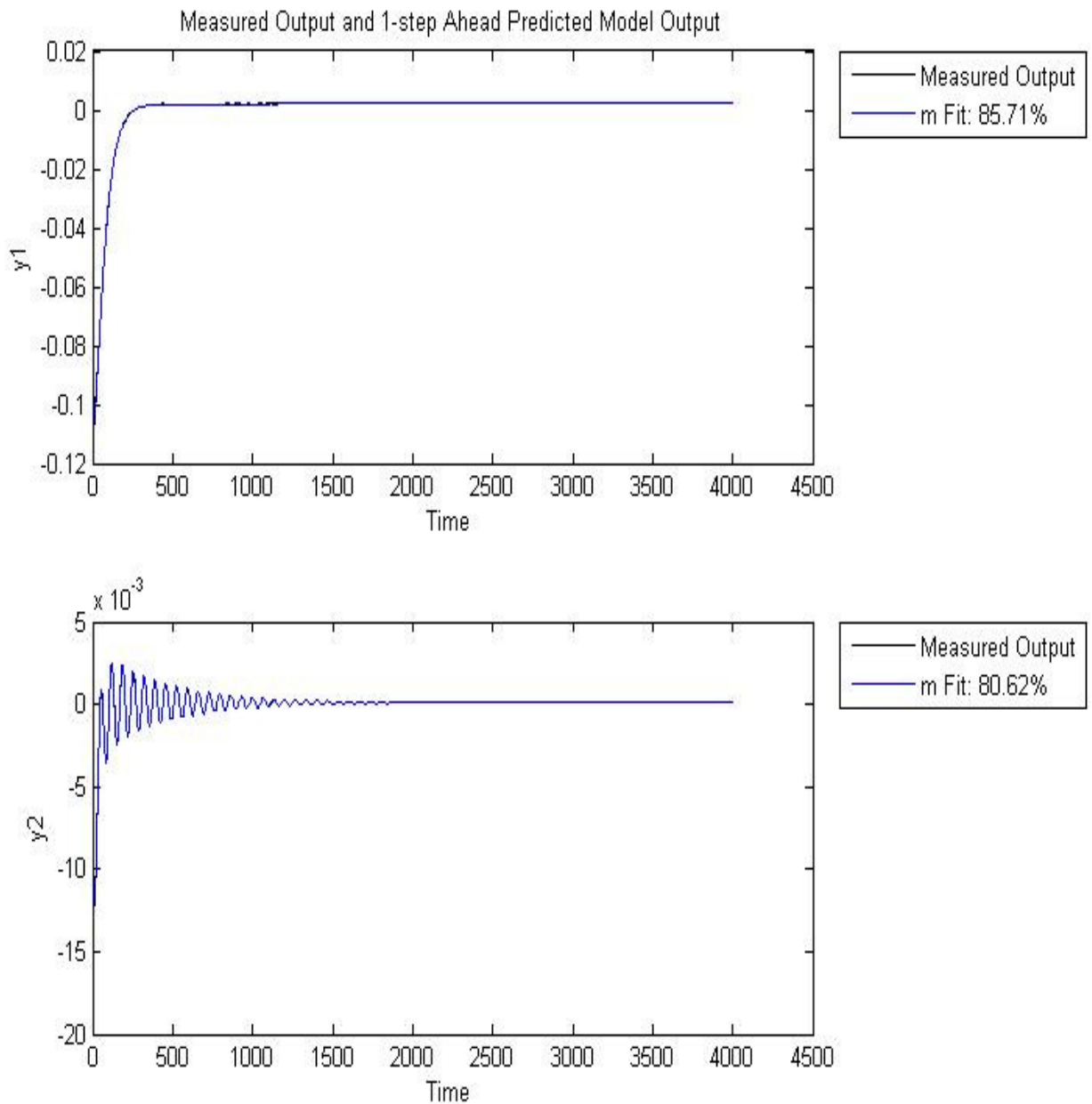


Figure 4.12 The 2nd operating point calibration by linearization tool

The measured and estimated values are fit together. Horizontal position is 85.71% and vertical is 80.62%

- The 3rd step

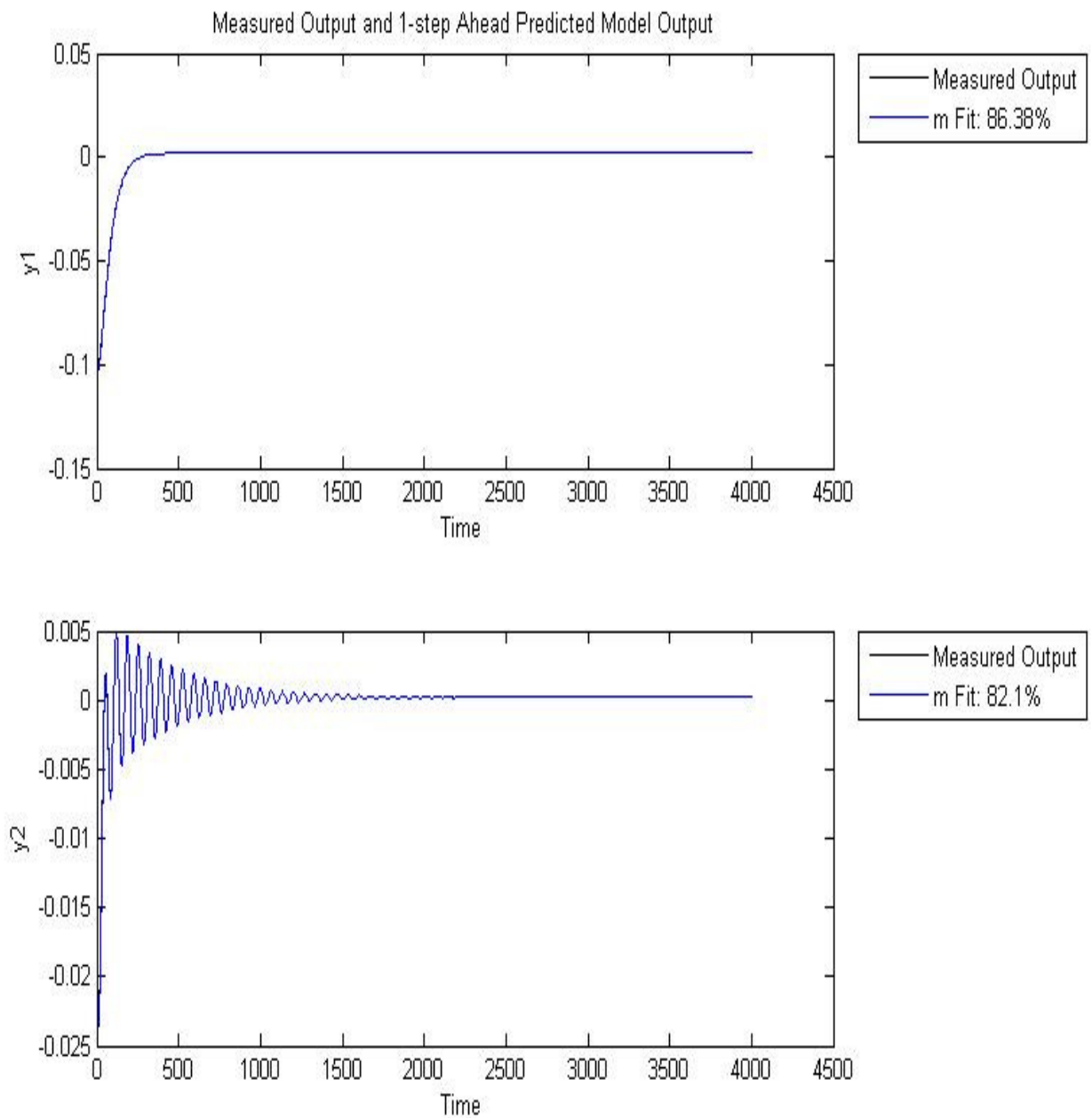


Figure 4.13 The 3rd operating point calibration by linearization tool

The measured and estimated values are fit together. Horizontal position is 85.38% and vertical is 82.1%

- The 4th step

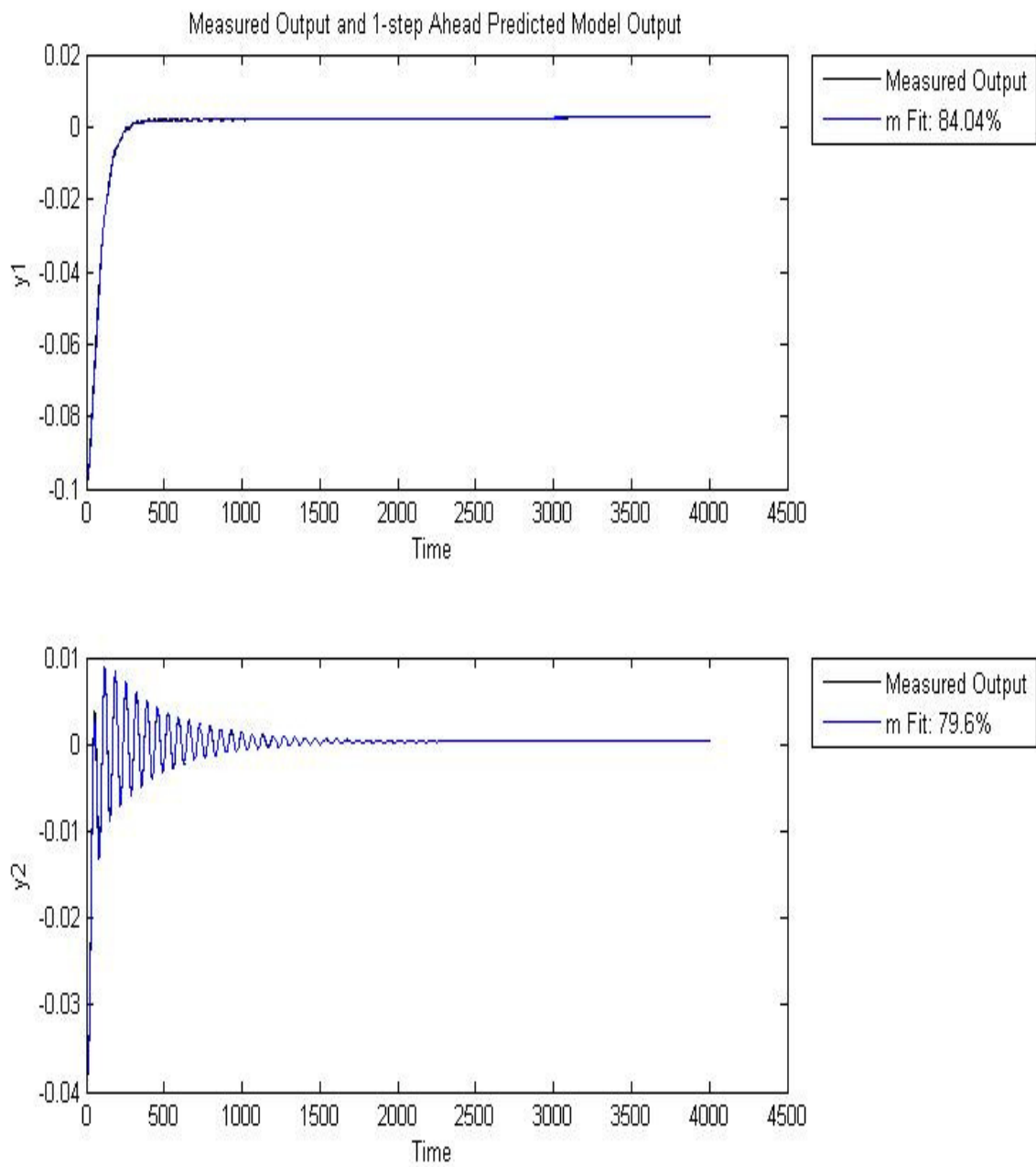


Figure 4.14 The 4th operating point calibration by linearization tool

The measured and estimated values are fit together. Horizontal position is 84.04% and vertical is 79.6% .

Table 4.3 Result of the calibration by linearization tool

Matrices	A						B	
	a_{11}	0	0	0	0	0	b_{11}	0
State space	0	a_{22}	0	0	0	0	0	b_{22}
	a_{31}	a_{32}	a_{33}	0	a_{35}	a_{36}	b_{31}	b_{32}
	a_{41}	a_{42}	a_{43}	1	a_{45}	a_{46}	b_{41}	b_{42}
	a_{51}	a_{52}	a_{53}	0	a_{55}	a_{56}	b_{51}	b_{52}
	a_{61}	a_{62}	a_{63}	0	a_{65}	a_{66}	b_{61}	b_{62}
	1 st Operating point	0.93131	0	0	0	0	0	62.92100
0		0.86675	0	0	0	0	0	178
-0.02421		0.00004	0.99713	0	0.04997	0.00139	0.00001	0.00742
0.08978		-0.00001	-0.01459	1	-0.00176	0.02426	0.00069	0.06726
-0.00053		0.00000	-0.17206	0	0.99018	0.00197	0.00050	-0.00054
0.00439		0.00007	0.00372	0	0.00907	0.97706	-0.00026	0.00472
2 nd Operating point	0.96724	0.00000	0.00000	0	0.00000	0.00000	61.85000	0.00000
	0.00000	0.87336	0.00000	0	0.00000	0.00000	0.00000	178
	-0.01037	-0.00017	0.99570	0	0.05156	0.00617	0.00002	0.39679
	0.00508	0.00209	-0.02035	1	0.00341	0.05420	0.00068	0.31079
	0.02716	0.00053	-0.17077	0	0.99236	-0.01346	0.00102	-0.00054
	0.01428	-0.00078	0.00072	0	-0.01673	0.97959	-0.00025	0.00469
3 rd Operating point	0.97349	0.00000	0.00000	0	0.00000	0.00000	59.88100	0.00000
	0.00000	0.95854	0.00000	0	0.00000	0.00000	0.00000	178
	-0.00326	0.00011	0.99491	0	0.05075	0.00486	0.00003	0.04212
	0.00392	0.00043	-0.00496	1	-0.00237	0.04659	0.00066	0.64302
	0.00748	0.00008	-0.17044	0	0.99102	-0.04270	0.00185	-0.00054
	0.00980	-0.00019	0.00284	0	-0.00216	0.98521	-0.00023	0.00463
	0.94344	0.00000	0.00000	0	0.00000	0.00000	57.048	0
	0.00000	0.95742	0.00000	0	0.00000	0.00000	0.00000	178
	-0.36561	0.00021	0.99153	0	0.05207	0.01029	0.00004	-0.00832
	-1.21700	0.00053	-0.00694	1	-0.00079	0.03597	0.00002	0.25244
	0.68938	0.00007	-0.16397	0	0.99448	-0.05769	0.00284	-0.00052
	-0.38194	-0.00022	0.00155	0	-0.00001	0.98611	-0.00177	0.00446

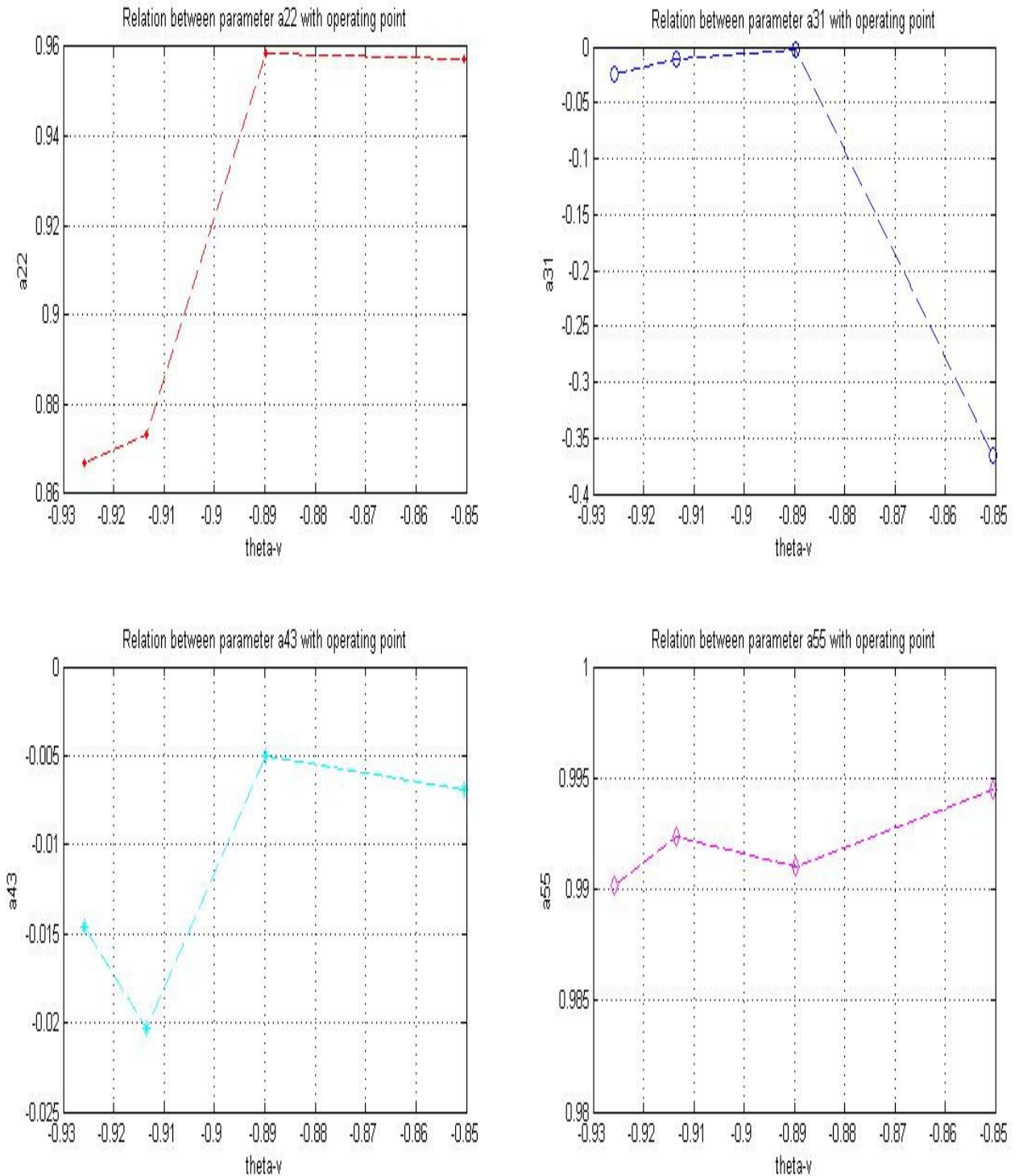


Figure 4.15 Estimated coefficients of matrix A by linearization

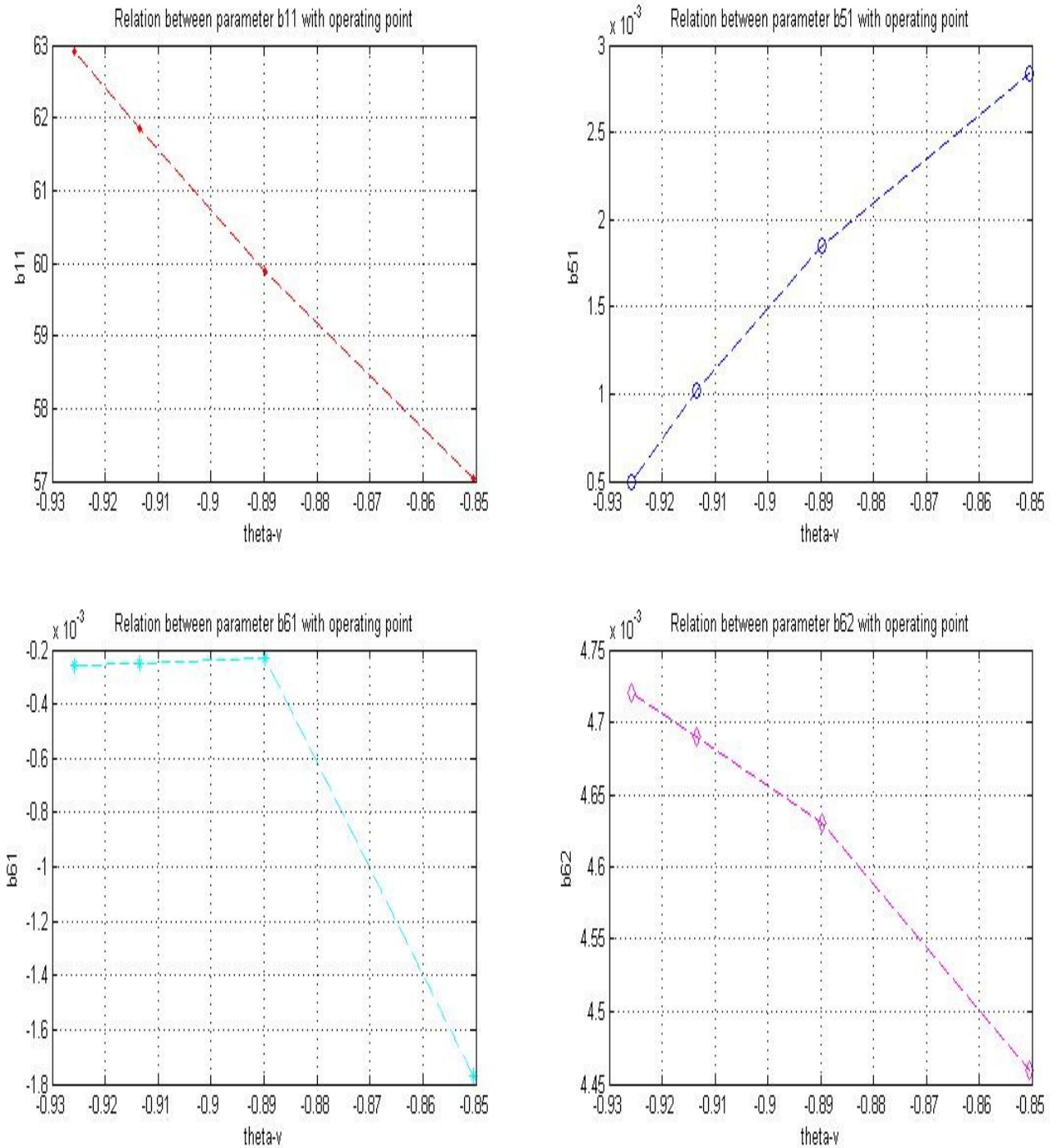


Figure 4.16 Estimated coefficients of matrix B by linearization

The above results can be defined this model that is a good model and can continue to do next task (implement with real data and control...)

5. CONCLUSIONS AND RECOMMENDATIONS FUTURE WORK

Twin Rotor MIMO system has a principal of function like a helicopter that has main rotor and tail rotor. Main rotor creates a lift force to lift the helicopter and tail rotor changes its direction but the structure is not the same. This model has been simplified to research in students' study but it assures the similarity between them.

This work analyses the mathematical model by using physical laws and simulates the system with Simulink – Matlab. This is an illustration to know the behavior of dynamic system clearly because of changing and making correct it easier.

The result of calibration of the LPV model for TRMS system is over 90% between the model and validation data so we can say this model is good. Next step is to repeat the calibration process using real data. With the non-linear dynamic system, finding its solution is not simple work and it should be assured well to apply by testing and calculating many ways. This case we can use linearization method by describing the state space to solve it. The calibration is used by syntax command and linearization tool. The fitted values using linearization tool is smaller than syntax command because it just validates around one operating point when moving out this point is not good.

The future work is:

- Implementation with the real data that is measured by experiments with TRMS equipment to know exactly the movements and control.
- Implementation of a control algorithm (for example, Model Predictive Control), that makes use of the LPV model proposed in the master thesis once calibrated using real data.
- Test and validation of the control algorithm in the real TRMS system.

STATE SPACE OF LPV MODEL OF TWIN ROTOR MIMO SYSTEM

1. $\frac{d\theta_h}{dt} = \omega_h$ we have $\tilde{\theta}_v = \theta_0 + \theta_v \rightarrow \tilde{\theta}_v - \theta_0 = \theta_v$ and new state is θ_v with initial condition $\theta_v=0.5225$

Where $\omega_h = \dot{S}_h + \frac{J_{mr} w_m \cos(\tilde{\theta}_v - \theta_0)}{J_h}$ derivative to obtain

$$\frac{d\omega_h}{dt} = \dot{S}_h + \frac{d}{dt} \left[\frac{J_{mr} \cos(\tilde{\theta}_v - \theta_0) w_m}{D \sin^2(\tilde{\theta}_v - \theta_0) + E \cos^2(\tilde{\theta}_v - \theta_0) + F} \right], \text{ then}$$

$$\frac{d\omega_h}{dt} = \dot{S}_h + \frac{d}{d\theta_v} \left[\frac{J_{mr} \cos(\tilde{\theta}_v - \theta_0) w_m}{D \sin^2(\tilde{\theta}_v - \theta_0) + E \cos^2(\tilde{\theta}_v - \theta_0) + F} \right] \dot{\theta}_v + \frac{d}{dw_m} \left[\frac{J_{mr} \cos(\tilde{\theta}_v - \theta_0) w_m}{D \sin^2(\tilde{\theta}_v - \theta_0) + E \cos^2(\tilde{\theta}_v - \theta_0) + F} \right] \dot{w}_m \text{ Replace } \dot{S}_h \text{ and}$$

$$\dot{w}_m = \frac{1}{\tau_{mr} (-w_m + k_{rm}(u_r) u_r)} \text{ with } w_m = u_{rr}$$

$$\begin{aligned} \frac{d\omega_h}{dt} = & \frac{l_t \cos(\tilde{\theta}_v - \theta_0) F_h(u_{tt})}{D \sin^2(\tilde{\theta}_v - \theta_0) + E \cos^2(\tilde{\theta}_v - \theta_0) + F} u_{tt} - \frac{k_h}{D \sin^2(\tilde{\theta}_v - \theta_0) + E \cos^2(\tilde{\theta}_v - \theta_0) + F} \omega_h + \frac{((E - D) \cos^2(\tilde{\theta}_v - \theta_0) - (F + D)) J_{mr} w_m \sin(\tilde{\theta}_v - \theta_0)}{(D \sin^2(\tilde{\theta}_v - \theta_0) + E \cos^2(\tilde{\theta}_v - \theta_0) + F)^2} w_v + \\ & + \frac{J_{mr} \cos(\tilde{\theta}_v - \theta_0)}{J_h} \left[\frac{1}{\tau_{mr}} (-u_{rr} + k_{rm}(u_r) u_r) \right] \end{aligned}$$

2.

$$\begin{aligned} \frac{d\omega_h}{dt} = & \frac{l_t \cos(\tilde{\theta}_v - \theta_0) F_h(u_{tt})}{D \sin^2(\tilde{\theta}_v - \theta_0) + E \cos^2(\tilde{\theta}_v - \theta_0) + F} u_{tt} - \frac{k_h}{D \sin^2(\tilde{\theta}_v - \theta_0) + E \cos^2(\tilde{\theta}_v - \theta_0) + F} \omega_h + \frac{((E - D) \cos^2(\tilde{\theta}_v - \theta_0) - (F + D)) J_{mr} w_m \sin(\tilde{\theta}_v - \theta_0)}{(D \sin^2(\tilde{\theta}_v - \theta_0) + E \cos^2(\tilde{\theta}_v - \theta_0) + F)^2} w_v - \\ & - \frac{J_{mr} \cos(\tilde{\theta}_v - \theta_0)}{J_h \tau_{mr}} u_{rr} + \frac{J_{mr} \cos(\tilde{\theta}_v - \theta_0) k_{rm}(u_r)}{J_h \tau_{mr}} u_r \end{aligned}$$

$$3. \frac{d\tilde{\theta}_v}{dt} = \omega_v \quad \text{where} \quad \dot{\omega}_v = \dot{S}_v + \frac{J_{tr}}{J_v} \dot{u}_{tt}$$

4.

$$\frac{d\omega_v}{dt} = \frac{l_m F_v(u_{rr})}{J_v} u_{rr} - \frac{k_v}{J_v} \omega_v + \frac{g((A-B)\cos(\tilde{\theta}_v - \theta_0) - C\sin(\tilde{\theta}_v - \theta_0))}{J_v \tilde{\theta}_v} \tilde{\theta}_v - \frac{1}{2J_v} (A+B+C)\omega_h^2 \sin(2(\tilde{\theta}_v - \theta_0)) + \frac{J_{tr}}{J_v} \left[\frac{1}{\tau_{tr}} (-u_{tt} + k_{tr}(u_t)u_t) \right]$$

$$5. \frac{du_{tt}}{dt} = -\frac{1}{\tau_{tr} u_{tt}} + \frac{k_{tr}(u_t)}{\tau_{tr}} u_t \quad \text{We consider the inputs are non-linear.}$$

$$6. \frac{du_{rr}}{dt} = -\frac{1}{\tau_{mr}} u_{rr} + \frac{k_{rm}(u_r)}{\tau_{mr}} u_r$$

*** by Maple:

$$xx := \frac{jmr \cdot wm \cdot \sin(\theta - \theta_0)}{D \sin(\theta - \theta_0)^2 + E \cos(\theta - \theta_0)^2 + F} - \frac{jmr \cdot wm \cdot \cos(\theta - \theta_0) (2D \sin(\theta - \theta_0) \cos(\theta - \theta_0) - 2E \cos(\theta - \theta_0) \sin(\theta - \theta_0))}{(D \sin(\theta - \theta_0)^2 + E \cos(\theta - \theta_0)^2 + F)^2}$$

1. IN LPV2 WE HAVE THE VALUES OF THE MODEL

g	= 9.81
lb	= 0.26
lcb	= 0.13
lm	= 0.24
lt	= 0.25
mb	= 0.022
mcb	= 0.068
mm	= 0.0145
mnr	= 0.228
mms	= 0.225
mt	= 0.0155
mtr	= 0.206
mts	= 0.165
rms	= 0.155
rts	= 0.1
taomr	= 1.432
taotr	= 0.3842

$$A = (mt/2 + mtr + mts) \cdot lt$$

$$B = (mm/2 + mnr + mms) \cdot lm$$

$$C = (mb/2) \cdot lb + mcb \cdot lcb$$

$$D = (mb/3 + mnr + mms) \cdot lm^2$$

$$E = (mm/3 + mnr + mms) \cdot lm^2 + (mt/3 + mtr + mts) \cdot lt^2$$

$$F = mms \cdot rms^2 + (mts/2) \cdot rts^2$$

$$jtr = 0.00001 \cdot 1.6543$$

$$kbal = 5 / (2.895 \cdot 2048)$$

$$jmr = 0.00001 \cdot 2.65$$

$$jv = 0.055846$$

$$kv = 0.00545371$$

$$kh = 0.0095$$

$$jh = D\sin(x)^2 + E\cos(x)^2 + F$$

2. DEFINITION THE PARAMETER TO OBTAIN LPV2 MODEL

We put the parameters equally

$$x(1) = \tilde{\theta}_h$$

$$x(2) = w_h$$

$$x(3) = \tilde{\theta}_v + \theta_0$$

$$x(4) = w_v$$

$$x(5) = u_{tt}$$

$$x(6) = u_{rr}$$

$$\frac{d\theta_h}{dt} = \omega_h$$

$$\frac{d\omega_h}{dt} = a_{22}\omega_h + a_{24}\omega_v + a_{25}u_{tt} + a_{26}u_{rr} + b_{22}u_r$$

$$\frac{d\tilde{\theta}_v}{dt} = \omega_v$$

$$\frac{d\omega_v}{dt} = a_{42}\omega_h + a_{43}\tilde{\theta}_v + a_{44}\omega_v + a_{45}u_{tt} + a_{46}u_{rr} + b_{41}u_t$$

$$\frac{du_{tt}}{dt} = a_{55}u_{tt} + b_{51}u_t$$

$$\frac{du_{rr}}{dt} = a_{66}u_{rr} + b_{61}u_r$$

STATE SPACE

$$\begin{bmatrix} \dot{\theta}_h \\ \dot{w}_h \\ \dot{\tilde{\theta}}_v \\ \dot{w}_v \\ \dot{u}_{tt} \\ \dot{u}_{rr} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & a_{22} & 0 & a_{24} & a_{25} & a_{26} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} \\ 0 & 0 & 0 & 0 & a_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & a_{66} \end{bmatrix} \begin{bmatrix} \theta_h \\ w_h \\ \tilde{\theta}_v \\ w_v \\ u_{tt} \\ u_{rr} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & b_{22} \\ 0 & 0 \\ b_{41} & 0 \\ b_{51} & 0 \\ 0 & b_{62} \end{bmatrix} \begin{bmatrix} u_t \\ u_r \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

```

%%                                     LPV2.m
%% _____ Programme LPV2
%% _____ From the non-linear equations
%% _____ Using S-FUNCTION

function [sys,x0,str,ts] = sfuntmpl(t,x,u,flag);
switch flag,
    %%%%%%%%%%%
    % Initialization %
    %%%%%%%%%%%
    case 0,
        [sys,x0,str,ts]=mdlInitializeSizes;
    %%%%%%%%%%%
    % Derivatives %
    %%%%%%%%%%%
    case 1,
        sys=mdlDerivatives(t,x,u);
    %%%%%%%%%%%
    % Update %
    %%%%%%%%%%%
    case 2,
        sys=mdlUpdate(t,x,u);
    %%%%%%%%%%%
    % Outputs %
    %%%%%%%%%%%
    case 3,
        sys=mdlOutputs(t,x,u);
    %%%%%%%%%%%
    % GetTimeOfNextVarHit %
    %%%%%%%%%%%
    case 4,
        sys=mdlGetTimeOfNextVarHit(t,x,u);
    %%%%%%%%%%%
    % Terminate %
    %%%%%%%%%%%
    case 9,
        sys=mdlTerminate(t,x,u);
    %%%%%%%%%%%
    % Unexpected flags %
    %%%%%%%%%%%
    otherwise
        error(['Unhandled flag = ',num2str(flag)]);
end
% end sfuntmpl
function [sys,x0,str,ts]=mdlInitializeSizes

%Here define the number of inputs, outputs and states

sizes = simsizes;
sizes.NumContStates = 6;    %No. State
sizes.NumDiscStates = 0;
sizes.NumOutputs = 6;    %No. outputs (this case is 6 states)
sizes.NumInputs = 2;    %No. inputs
sizes.DirFeedthrough = 1;

```

```

sizes.NumSampleTimes = 1; % at least one sample time is needed

sys = simsizes(sizes);

% Inicial parameters
% x0 = [];
THETA_0=0.5225; %initial condition 30°=0.5225
x0=[0 0 THETA_0 0 0 0];

% str is always an empty matrix
str = [];
% initialize the array of sample times
ts = [0 0];
% end mdlInitializeSizes

function sys=mdlDerivatives(t,x,u)

%initial variables
g=9.81;
lb=0.26;
lcb=0.13;
lm=0.24;
lt=0.25;
mb=0.022;
mcb=0.068;
mm=0.0145;
mmr=0.228;
mms=0.225;
mt=0.0155;
mtr=0.206;
mts=0.165;
rms=0.155;
rts=0.1;
taomr=1.432; %cte time of main rotor
taotr=0.3842; %cte time of tail rotor
THETA_0=0.5225;
A = (mt/2+mtr+mts)*lt; %0.09488;
B = (mm/2+mmr+mms)*lm; %0.11046;
C = (mb/2)*lb+mcb*lcb;
D = (mb/3*lb^2)+(mcb*lcb^2); %0.0016449;
E = (mm/3+mmr+mms)*lm^2 + (mt/3+mtr+mts)*lt^2;
F = mms*rms^2+(mts/2)*rts^2;

jtr=0.00001*1.6543;
jmr=0.00001*2.65;
jv= 0.055846;
kv= 0.00545371;
kh= 0.0095;
kbal = 5/(2.895*2048);

jh = D*sin(x(3)-THETA_0)^2+E*cos(x(3)-THETA_0)^2+F; %horizontal
inertia

% Polinomials

```

```

pwt = [2020 -194.69 -4283.15 262.27 3796.83 0]; %POLINOMIAL of
FUNCION of U_TT
pfh = [-.000000000000003 -.00000000001595 .00000025121766 -
.00018087473021 .08010909850208 0]; % polinomio en funcion de wt
pwm = [90.99 599.73 -129.26 -1238.64 +63.45 1283.41 0]; %POLINOMIO EN
FUNCION DE U_RR

pfv = [-.000000000000348 .00000000109079 .0000041231489 -
.0001632767956 .095447585 0]; %polinomio en funcion de wm

% Funciones
wt_utt = polyval(pwt,x(5));

%-----Characteristic the tail rotor(look MANUAL)

liminf=-1381;
limsup=1630;
wt=max(min(wt_utt,limsup),liminf);
fh = kbal*polyval(pfh,x(5));

wm_urr = polyval(pwm,x(6));%x(6)=urr

%-----Characteristic the main rotor(look MANUAL)

liminfp=-568;
limsupp=668;
wm=max(min(wm_urr,limsupp),liminfp);
fv = kbal*polyval(pfv,x(6));

% ESTADOS x = [th,wh,tv,wv,utt,urr] % urr=Wm = x(5); % utt=Wt =
x(6); %

%Consider non-linear inputs

krm_ur=90.99*(u(2))^6+599.73*(u(2))^5-129.26*(u(2))^4-
1238.64*(u(2))^3+63.45*(u(2))^2+1283.41*(u(2));
krt_ut=2020*u(1)^5-194.69*u(1)^4-
4283.15*u(1)^3+262.27*u(1)^2+3796.83*u(1);

% % Implement the differential equations of no-linear model

% % % sys(1) = x(2);

% % % sys(2) = lt*cos(x(3)-THETA_0)*(fh)/jh - (kh/jh)*x(2) + ((E-
D)*(cos(x(3)-THETA_0)^2)-(F+D))*jmr*(sin(x(3)-
THETA_0)/jh^2)*x(4)*x(6)+ ((jmr*cos(x(3)-THETA_0))/(jh*taomr)) * (-
x(6) + krm_ur);
% % % sys(3) = x(4);
% % % sys(4) = lm*(fv/jv) - (kv/jv)*x(4) + (g*((A-B)*cos(x(3)-
THETA_0)-C*sin(x(3)-THETA_0))/(jv) -
(0.5*(x(2)^2)*(A+B+C)*sin(2*(x(3)-THETA_0)))/(jv)+...
% % % (jtr/(jv*taotr))*( -x(5) + krt_ut);
% % % sys(5) = -1/taotr*(x(5)-krt_ut);
% % % sys(6) = -1/taomr*(x(6)-krm_ur);

```

```

% _____OBTAIN THE LPV2 MODEL ____
pwt = [2020 -194.69 -4283.15 262.27 3796.83];
pvh = [-.0000000000000003 -.00000000001595 .00000025121766 -
.00018087473021 .08010909850208];
pwm = [90.99 599.73 -129.26 -1238.64 +63.45 1283.41];
pvm = [-.0000000000000348 .00000000109079 .0000041231489 -
.0001632767956 .095447585];

sys(1) = [0 1 0 0 0 0]*x;
a22 = -kh/jh;
a24 = ((E-D)*(cos(x(3)-THETA_0)^2)-(F+D))*jmr*(sin(x(3)-
THETA_0)/jh^2)*x(6);
a25 = kbal*lt*cos(x(3)-THETA_0)*(polyval(pvh,x(5)))/jh;

a26 = -jmr*cos(x(3)-THETA_0)/(jh*taomr);
sys(2) = [0 a22 0 a24 a25 a26]*x+[0 ((jmr*cos(x(3)-
THETA_0))/(jh*taomr))*polyval(pwm,u(2))]*u;

sys(3) = [0 0 0 1 0 0]*x;
a42 = -0.5*x(2)*(A+B+C)*sin(2*(x(3)-THETA_0))/jv;
a43 = g*((A-B)*cos(x(3)-THETA_0)-C*sin(x(3)-THETA_0))/(jv*x(3)); %
*x(3)
a44 = -kv/jv;
a45 = -(jtr/(jv*taotr));
a46 = kbal*lm*polyval(pvm,x(6))/jv;

sys(4) = [0 a42 a43 a44 a45 a46]*x+
[((jtr*polyval(pwt,u(1)))/jv/taotr) 0]*u;
sys(5) = [0 0 0 0 -1/taotr 0 ]*x + [(1/taotr)*polyval(pwt,u(1)) 0
]*u;
sys(6) = [0 0 0 0 0 -1/taomr]*x + [ 0
(1/taomr)*polyval(pwm,u(2))]*u;

% end mdlDerivatives
function sys=mdlUpdate(t,x,u)
sys = [];
% end mdlUpdate
function sys=mdlOutputs(t,x,u)
lb =0.26;
lcb=0.13;
lm =0.236;
lt =0.25;
mb =0.022;
mcb=0.068;
mm =0.0145;
mmr=0.228;
mms=0.225;
mt =0.0155;
mtr=0.206;
mts=0.165;
rms=0.155;
rts=0.1;

```

```
D = (mb/3*lb^2)+(mcb*lcb^2);    %0.0016449;
E = (mm/3+mmr+mms)*lm^2 + (mt/3+mtr+mts)*lt^2;
F = mms*rms^2+(mts/2)*rts^2;
jh= D*sin(x(3))^2+E*cos(x(3))^2+F;

jtr=0.00001*1.6543;
jmr=0.00001*2.65;
jv = 0.055846;
THETA_0=0.5225;

% DECLARE THE OUTPUTS OF MODEL
sys(1) = x(1);
sys(2) = x(2);
sys(3) = x(3) + THETA_0;
sys(4) = x(4);
sys(5) = x(5);
sys(6) = x(6);

% end mdlOutputs
function sys=mdlGetTimeOfNextVarHit(t,x,u)
sampleTime = 1;    % Example, set the next hit to be one second
later.
sys = t + sampleTime;
% end mdlGetTimeOfNextVarHit
function sys=mdlTerminate(t,x,u)
sys = [];
% end mdlTerminate
```


% CALIRATION m.file

```

y1=theta_h;
y2=theta_v;
u1=ut;
u2=ur;
g=9.81;
lb=0.26;
lcb=0.13;
lm=0.24;
lt=0.25;
mb=0.022;
mcb=0.068;
mm=0.0145;
mmr=0.228;
mms=0.225;
mt=0.0155;
mtr=0.206;
mts=0.165;
rms=0.155;
rts=0.1;
taomr=1.432;
taotr=0.3842; THETA_0=0.5225;
jtr=0.00001*1.6543;
jmr=0.00001*2.65;
jv= 0.055846;
kv= 0.00545371;
kh= 0.0095;
kbal = 5/(2.895*2048);

% Creating a data file
trms=iddata([detrend(y1(4000:8000),'constant') detrend(y2(4000:8000),
'constant')],[detrend(u1(4000:8000),'constant')
detrend(u2(4000:8000),'constant')]);

a22n=mean(a22)
a24n=mean(a24)
a25n=mean(a25)
a26n=mean(a26)
a42n=mean(a42)
a43n=mean(a43)
a46n=mean(a46)
b22n=mean(b22)
b41n=mean(b41)
b51n=mean(b51)
b62n=mean(b62)
a44=-kv/jv;
a45=-jtr/(taotr*jv);
a55=-1/taotr;
a66=-1/taomr;

% Construct Matrices to calibrate

```

```
A=[1 0.05 0 0 0 0;0 1+0.05*a22 0 0.05*a24 0.05*a25 0.05*a26;0 0 1
0.05 0 0;0 0.05*a42 0.05*a43 1+0.05*a44 0.05*a45 0.05*a46;0 0 0 0
1+0.05*a55 0;0 0 0 0 0 1+0.05*a66];

B=[0 0;0 0.05*b22;0 0;0.05*b41 0;0.05*b51 0;0 0.05*b62];

C=[1 0 0 0 0 0;0 0 1 0 0 0];

D=zeros(2,2);

m=idss(A,B,C,D);

m.As=[1 0.05 0 0 0 0;0 NaN 0 NaN NaN NaN;0 0 1 0.05 0 0;0 NaN NaN NaN
NaN NaN;0 0 0 0 NaN 0;0 0 0 0 0 NaN];
m.Bs=[0 0;0 NaN;0 0;NaN 0;NaN 0;0 NaN];
m.Cs=m.c;
m.Ds=m.d;

%Calibration the model

m=pen(trms,m,'Focus','Simulation','DisturbanceModel','Estimate')
compare(trms,m,1)

-----end-----
```

References

- [1] ALVERTO BEMPORAD, MANFRED MORARI, N. LAWRENCE RICKER, Getting Stared with Model Predictive Control Toolbox 2, the Math Works Inc, 2007.
- [2] ALVERTO BEMPORAD, MANFRED MORARI, N. LAWRENCE RICKER, User's guide Model Predictive Control Toolbox 2, The Math Works Inc, 2007.
- [3] BRAMWELL, Helicopter Dynamics, 2nd edition, Butterworth-Heinemann, 2001.
- [4] Feedback Instruments Ltd, Twin Rotor Mimo System (33- 007- 4M5), park Road, E. Sussex, TN6 2QR UK, p 1.1-4.4
- [5] KATSUHIKO OGATA, Modern Control Engineering, 3rd edition, New Jersey - US, p. 70-76, 1997.
- [6] LENNART LJUNG, Getting started with system identification toolbox 7, the Math Works Inc, 2007.
- [7] LENNEART LJUNG, User's guide system identification toolbox 7, The Math Works Inc, 2007.
- [8] PEDRO LEAL RIBERIO, Course of non-linear dynamics for Erasmus Mundus Master, University of Porto, May 2007.
- [9] WIKIPEDIA, <http://en.wikipedia.org/wiki/Helicopter>.

