MOTION CONTROL OF AN OPEN CONTAINER WITH SLOSH CONSTRAINTS

KEDAR B KARNIK

Bachelor of Science in Mechatronics Engineering

Sardar Patel University

Gujarat, India May, 2004

submitted in partial fulfillment of requirements for the degree

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

at the

CLEVELAND STATE UNIVERSITY

Dec, 2008

This thesis has been approved
for the department of MECHANICAL ENGINEERING
and the College of Graduate Studies by:

Thesis Chairperson, Hanz Richter, Ph.D.

Department & Date

Jerzy T Sawicki, Ph.D.

Department & Date

Daniel Simon, Ph.D.

Department & Date

ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Hanz Richter, who provided essential support and assistance throughout my graduate career, and also for his guidance which immensely contributed towards the completion of this thesis. I would also like to thank Dr. Jerzy Sawicki and Dr. Simon for being in my thesis committee. Thanks are also due to my family and friends who have encouraged, supported and inspired me.

MOTION CONTROL OF AN OPEN CONTAINER WITH SLOSH CONSTRAINTS

KEDAR B KARNIK

ABSTRACT

General motion control of conveyor belts does not present difficulties. When the conveyor belt carries open containers filled with liquid, significant analysis needs to be carried out to design controllers. The objective of this thesis was to design a control system which will allow an open container filled with liquid to be transferred between two stations as fast as possible and without excessive slosh causing the liquid to spill out of the container. This control problem has applications to industrial processing facilities, where open containers are carried by a conveyor belt. The speed at which the open container can be transferred between stations has a direct impact on productivity.

The thesis involves determination of the plant (conveyor belt dynamics and the container filled with liquid) model using system identification techniques and examination of candidate control techniques. Simulation results have been shown to validate the approach.

TABLE OF CONTENTS

ABSTRACT iv

LIST OF FIGURES vii

I INTRODUCTION 1

1.1 Overview of Sloshing 1

1.2 Motivation 2

1.3 Literature Review 2

1.4 Objective 3

1.4.1 Composition of Thesis 4

II EXPERIMENTAL SETUP 5

2.1 Introduction 5

2.2 Components 7

2.2.1 Slide and Tank 7

2.2.2 LVDT 8

2.2.3 Magnetostrictive Sensor 8

2.2.4 HP Spectrum Analyser 10

2.2.5 Rapid Control Prototyping Hardware 10

III SYSTEM IDENTIFICATION 12

3.1 Physics of Slosh Motion 13

3.1.1 Related Work 13

3.2 Frequency Domain Tests 16

3.2.1 Frequency domain analysis using HP Spectrum Analyzer 16

3.2.2 Individual frequency tests 17

3.2.3 Frequency domain test using Chirp signal 17

3.3 Parametric model for the plant 18
3.4 Time Domain Tests 19
3.4.1 Step response analysis 19

IV SLIDING MODE CONTROL 23
4.1 Sliding Mode Design 24
4.1.1 Sliding Mode Control Regulator 26
4.2 Simulation and Results 29
4.2.1 Model Simulation 29
4.2.2 Real-Time digital SMC 30

V CONCLUSIONS AND FUTURE WORK 33

BIBLIOGRAPHY 35

A MATLAB PROGRAMS AND SIMULATION MODELS 38
1.1 Plant identification 38
1.2 Mathematical model for tank 44
1.3 Reduced order model 50
1.4 Sliding Mode Control design 55
1.5 Real Time plots 59
1.6 Simulink model for Digital SMC 65
1.7 Simulink model for masked subsystems 66

LIST OF FIGURES

2.1 Schematic Overview 6

2.2 Schematic Overview 6

2.3 Actual Experimental Setup 7

2.4 Magnetostrictive Level Sensor - Principle of operation 9

2.5 Level sensor calibration - Voltage (V) vs Depth (cm) 10

2.6 dSpace - Application Window 11

3.1 Frequency response for the tank - Input is Analog voltage, Output is Level sensor output 17

3.2 Input signal to slide position (mm) - modeled vs recorded output . . 20

3.3 Slide position to level sensor (volts) - modeled vs recorded output . . 20

3.4 Time-domain model prediction, Step response 22

3.5 Time-domain model prediction; Step response 22

4.1 Simulation Result, SMC 30

4.2 Realtime, Trial 1 - Model output from the realtime system and the experimental setup 31

4.3 Realtime, final - Model output from the realtime system and the experimental setup 32

1.1 Simulink model for digital SMC 65

1.2 Simulink model for masked subsystem 66

CHAPTER I

INTRODUCTION

This Chapter justifies the undertaken study and introduces the research work which is conducted in this thesis. The literature review pertinent to the subject is presented. At the end, the organization of the thesis is given.

1.1 Overview of Sloshing

Fluid motion in partially filled tanks may cause large structural loads if the period of tank motion is close to the natural period of fluid inside the tank. This phenomenon is called sloshing. Sloshing means any motion of a free liquid surface inside a container. The amplitude of the slosh, in general, depends on the nature, amplitude and frequency of the tank motion, liquid-fill depth, liquid properties and tank geometry [1]. Sloshing is a phenomenon that is observed in all liquid-filled containers and is of great engineering importance. Motion-induced sloshing has been a classical control problem. It was first encountered in the control of guided missiles. The dynamic forces resulting from the motion of these fluid-filled missiles can be substantial and result in instabilities. The first analytical solution to such a problem was addressed by Westergard [6]. Liquid sloshing is a severe problem in transportation ships [3]. Lately

movement of open containers carrying liquids has been investigated by Schmidt [3].

1.2 Motivation

The motivation for the research was the cost associated with the transport of liquid filled containers. Designing a control to reduce the time to move the container between two stations, would mean greater production capacity and hence lower per container production cost. If the motion can be controlled such that sloshing is minimum then the containers could be filled to a greater depth and hence reduce the packaging cost as well.

One of the major issues in the packing industry is the machine downtime due to maintenance. Liquid is bound to spill while packaging which results in frequent maintenance of the machine. This research also aims at reducing the downtime by restricting the spilling of the liquids during packaging. This increases the production capacity.

1.3 Literature Review

The following paragraph lists several researchers' work in relation to this thesis. Studies have been conducted to identify the non-linear behavior of the liquid sloshing. Mathematical models of sloshing have been identified using finite element analysis.

Akyildz used a numerical algorithm based of the volume of fluid technique to identify the non-linearity and damping characteristic of sloshing [1]. He utilized a moving co-ordinate system to model the complex boundary conditions. These methods were used to compute the pressure variation due to sloshing in a partially filled tank. It was concluded that there are two major kinds of pressures, impulsive and non-impulsive, and the severity of sloshing depends on the tank geometry, depth

of the liquid and excitation frequencies.

Grundelius used the method of determining the acceleration profile to control the motion of the moving tank [3]. This control approach is typically open-loop. In many cases, a liquid level sensor cannot be installed in each moving container. This precludes the application of entire family of state-feedback based controllers, including observer-based approaches. In absence of liquid sensing, an optimal input profile approach may be used to reduce the excitation of sloshing dynamics. A linear second order oscillator was used to model the slosh. The minimum energy approach was used to calculate the acceleration profile (applied in open-loop fashion) which did not satisfy the time constraint due to poor plant model estimation. This approach relies on accurate mathematical model of the liquid container and cannot accommodate disturbances or model errors. An assumption is also made in [3] that the transport mechanism is (slide, conveyor belt) stiff enough to be unaffected by fluid motion. This thesis, therefore, does not make this assumption and relies on modeling the plant using system identification techniques. It assumes and that the liquid level at the edge of the tank and its rate can be measured in real-time. The availability of liquid level measurements enables the application of host of control approaches, including a high level of robustness which are amenable to constraint handling.

1.4 Objective

This thesis is aimed at designing a control system for a moving container. The motion is constrained due to slosh effect of the liquid inside it. Conventional methods to determine the slosh model were either finite element methods or various fluid mechanics methods which could be used in the absence of liquid level sensing. The mathematical model could be used to solve the minimum time and energy problem. This approach relies on accurate mathematical model of the plant. In this thesis a

mathematical model for the plant is derived using system identification technique. This would eliminate the uncertainties associated with the mathematically modeling using various universal laws. The non-linearity of the slosh is another factor which could be taken care of using nonlinear system identification. In this thesis, however, we use a linear dynamic model, which is shown to be sufficiently accurate [7].

The primary task in achieving the objective was extensive collection of real time data for frequency and time domain analysis. This data was also used in determining the parametric model for the plant using *MATLAB*.

The objective was fulfilled by determining an accurate 4th order model for the plant and devising a control using Sliding Mode Control (SMC) methods which can be tuned to meet transfer time specification under slosh constraints.

1.4.1 Composition of Thesis

The organization of the thesis is in the following manner. The experimental setup is introduced in Chapter 2. A schematic diagram and picture of the set up is shown. Each component is discussed in detail. Fundamentals and steps for system identification have been dealt with in Chapter 3. Various plots and experimental methods have been presented for identification of the plant.

Chapter 4 considers the the closed loop control when constraints are imposed. It talk about the SMC technique. Finally results are summarized with conclusions of the research work conducted for the thesis being provided along with the scope for future work in Chapter 5.

CHAPTER II

EXPERIMENTAL SETUP

2.1 Introduction

The experimental setup is shown in Figure 2.1. The figure shows a slide (representing a conveyor belt) for the motion driven by analog signal. A liquid (water for experimental purpose) filled container is mounted on the slide. These two comprise the cascaded plant for the experiment. The Linear Variable Differential Transformer (LVDT) is attached to the container to measure the position of the container at any given point. Later, the LVDT was replaced by an encoder to record the position with greater accuracy and range. The dynamics between the analog input voltage and slide position were obtained by standard system identification techniques. The Level sensor is a float which slides along a stem, minimizing the frictional effects in the sensor. A dSpace/Simulink realtime interface is used to measure signals from various sensors, evaluate the control signal and send it to slide actuator. Figure 2.2 shows the schematic of the experimental setup and the flow of signals. Figure 2.3 shows the actual picture of the experimental setup.

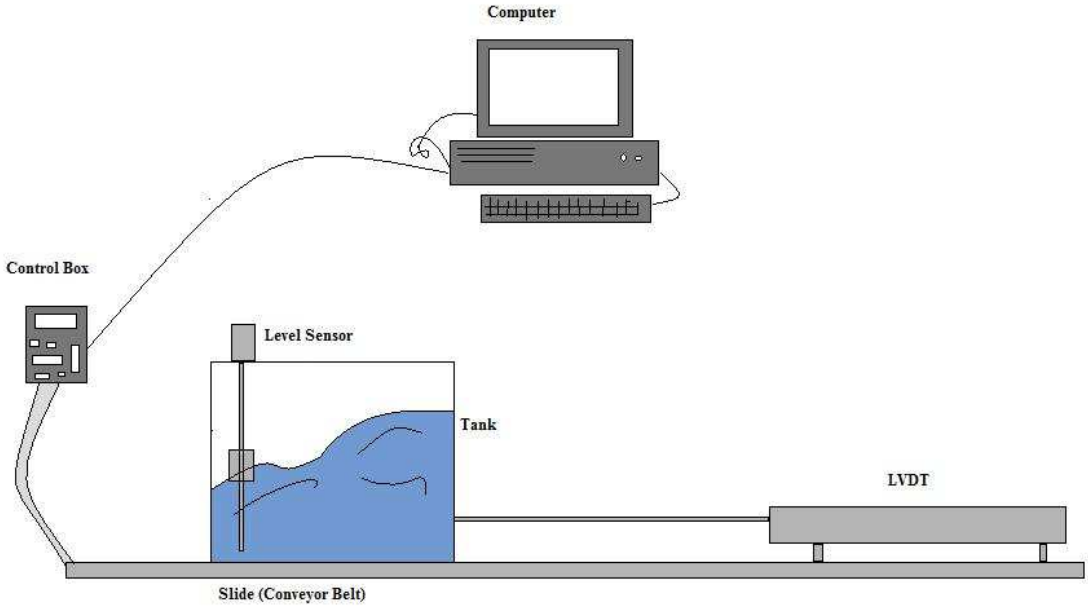


Figure 2.1: Schematic Overview

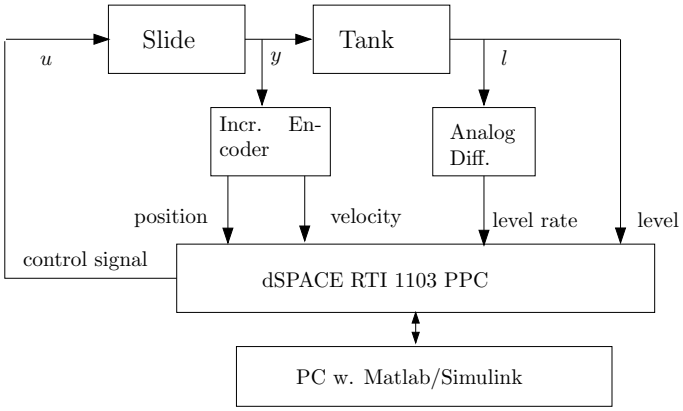


Figure 2.2: Schematic Overview

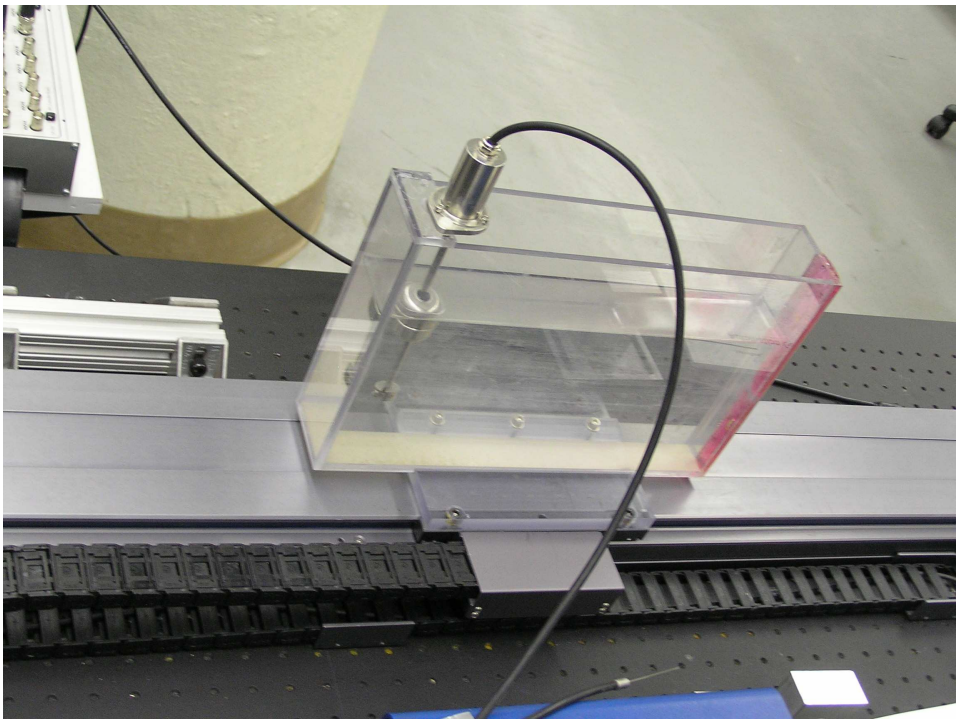


Figure 2.3: Actual Experimental Setup

2.2 Components

2.2.1 Slide and Tank

The tank was mounted on a linear positioning slide (Luge LM150D manufactured by Parker-Bayside). The slide is operated from an analog input voltage in the range of ± 10 V. A steady input voltage, after transients, results in a steady slide velocity. The DC motor driving the slide is fitted with a 2000-line rotary encoder which was used to determine the slide position after the LVDT couldn't give satisfactory results. The encoder gives the slide position and velocity with an accuracy of 1/200 mm.

The inner dimensions of the tank were 305 mm by 50 mm, and the water level at which slosh dynamics were identified was 136 mm.

2.2.2 LVDT

The LVDT used for the experiment was from Honeywell. It was a long stroke LVDT, model *JEC-AG DC-DC*. The data gathered using the LVDT was susceptible to noise and hence needed filtering. A filter was not sufficient to clean the signal and hence the LVDT was replaced by an encoder. In addition, the LVDT has a limited range given by its mechanical stroke. The encoder gave accurate readings of the position of the slide and also had a greater range.

2.2.3 Magnetostrictive Sensor

Magnetostrictive level sensors are similar to float type sensors in that a permanent magnet sealed inside a float travels up and down a specially designed magnetostrictive waveguide. A sonic strain pulse is induced is induced by momentary interaction of the two magnetic fields. One is from the permanent magnet inside the float and another from the current pulse applied along the waveguide. The position of the magnet is determined by calculating the time elapsed between the application of current pulse and arrival of resulting strain pulse. Figure 2.4 (www.mtssensors.com) shows the principle of operation. Ideal for high-accuracy, continuous level measurement of a wide variety of liquids in storage and shipping containers, these sensors require the proper choice of float based on the specific gravity of the liquid. When choosing float and stem materials for magnetostrictive level sensors, the same guidelines described for magnetic and mechanical float level sensors apply.

Because of the degree of accuracy possible with the magnetostrictive technique, it is popular for custody-transfer applications. It can be permitted by an agency of weights and measures for conducting commercial transactions. It is also frequently applied on magnetic sight gages. In this variation, the magnet is installed in a float that travels inside a gage glass or tube. The magnet operates on the sensor which

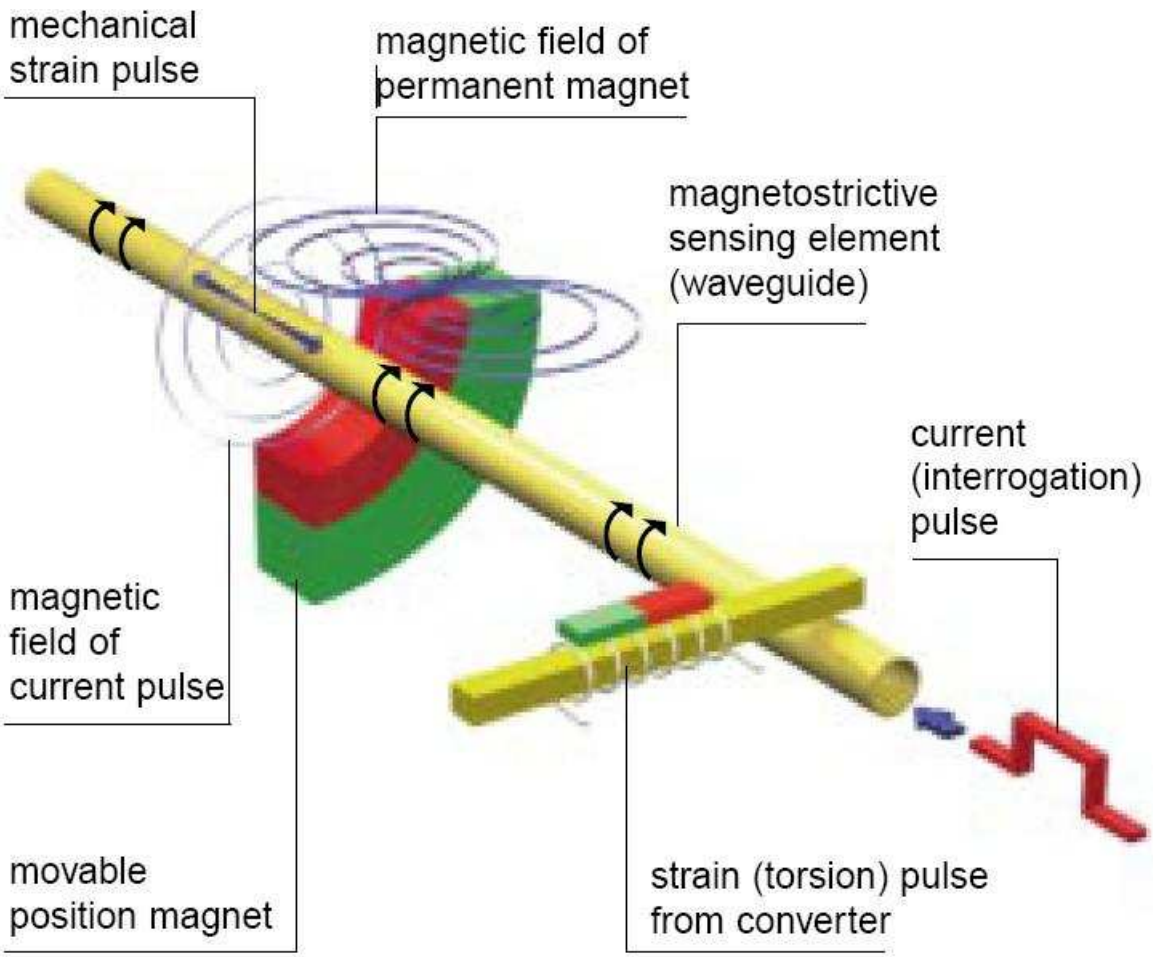


Figure 2.4: Magnetostrictive Level Sensor - Principle of operation

is mounted externally on the gage. Boilers and other high temperature or pressure applications take advantage of this performance quality.

Fig 2.5 shows the curve for sensor output voltage and depth of the liquid. This curve was used to calculate the liquid depth. The sensor provides an analog voltage in the 0-5 V range with a constant sensitivity of -23.43 V/mm. The level rate was obtained by analog differentiation using standard operational amplifier circuits.

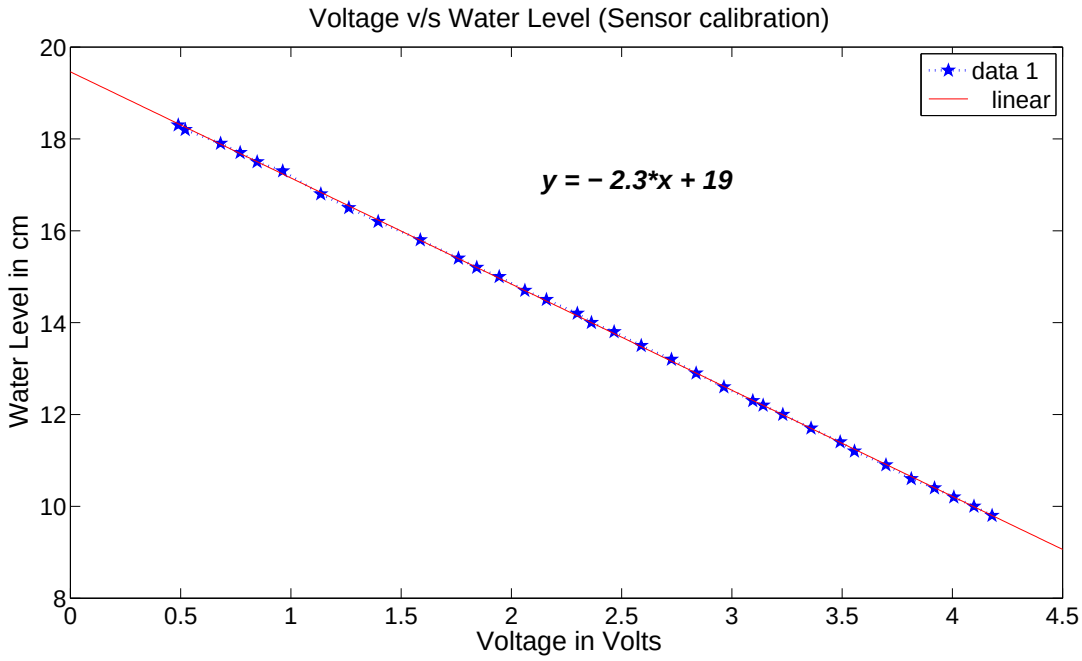


Figure 2.5: Level sensor calibration - Voltage (V) vs Depth (cm)

2.2.4 HP Spectrum Analyser

An HP 3265A spectrum analyzer was used to record the frequency response of the plant to an input swept sine wave from 0.3 Hz to 8 Hz frequency. It helped in determining the resonant frequencies which were later verified mathematically by the formula[1].

2.2.5 Rapid Control Prototyping Hardware

A dSPACE RTI-1103 PPC board and Control desk software were used to create *Simulink* interfaces for system identification and rapid control prototyping. Figure 2.6 shows the application window for dSpace.

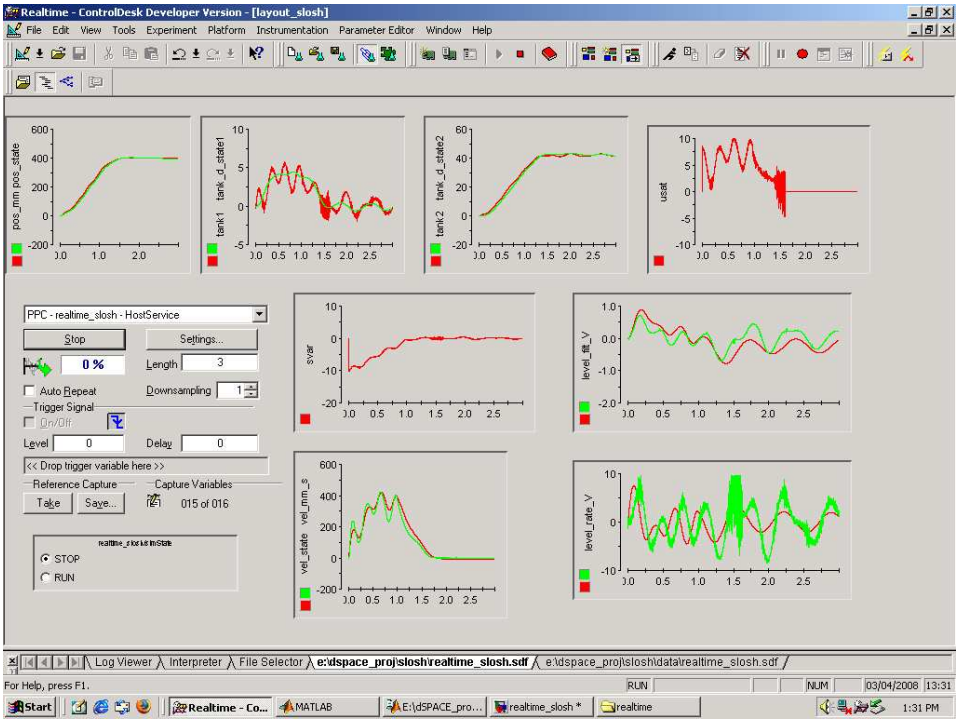


Figure 2.6: dSpace - Application Window

CHAPTER III

SYSTEM IDENTIFICATION

This Chapter describes the use of system identification techniques to obtain a mathematical model of the controlled plant. System identification is a general term to describe mathematical tools and algorithms that build dynamical models from measured data. A dynamical mathematical model in this context is a mathematical description of the dynamic behavior of a system or process.

Examples include:

- physical processes such as the movement of a falling body under the influence of gravity
- economic processes such as stock markets that react to external influences

One could build a so-called white-box model based on first principles, e.g. a model for a physical process from the Newton equations, but in many cases such models will be overly complex and possibly even impossible to obtain in reasonable time due to the complex nature of many systems and processes.

A more common approach is therefore to start from measurements of the behavior of the system and the external influences (inputs to the system) and try to determine a mathematical relation between them without going into the details of what is actually

happening inside the system. This approach is called system identification. Two types of models are common in the field of system identification:

- grey box model: although the peculiarities of what is going on inside the system are not entirely known, a certain model is already available. This model does however still have a number of unknown free parameters which can be estimated using system identification. This is also known as semi-physical modeling.
- black box model: No prior model is available. Most system identification algorithms are of this type.

System identification can be done in either the time or frequency domain [5].

3.1 Physics of Slosh Motion

This section describes the slosh phenomenon and description of different mathematical descriptions of fluid flow.

3.1.1 Related Work

In [8] the fluid is modeled in two dimensions by both the Reynolds Averaged Navier-Stokes Equations (RANSE) for an incompressible flow and the Shallow Water Equations (SWE). Both problems are solved numerically with different finite element methods. RANSE is solved using a method called SIMAC (Semi-Implicit Marker And Cell) which uses a fixed stretched grid. Each cell in the grid can either contain fluid or not. The numerical solutions are compared with experiments that show good correlation with the numerical solutions.

The problem of free surface flows in domains with moving boundaries are described in [4]. The method handles two-fluid flows where the two fluids can both be incompressible or one of the fluids can be compressible. Moving grids are utilized to

accommodate the motion of the domain boundaries. The volume-of-fluid technique is used for tracking the free surface between the two fluids.

In [10] the fluid flow is described as a potential flow using Laplace equation. The dynamic boundary condition on the free surface is the damping-modified Bernoulli equation. The damping in the Bernoulli equation is a simple way to include viscosity in the model. The problem is solved using a boundary element method. The boundary on the free surface is updated to track the motion of the surface.

The flow problem is solved analytically in [9]. The flow is described by the Laplace equation and the boundary conditions on the free surface is described by the Bernoulli equation. Separation of variables is used to find the horizontal and vertical modes. The Bernoulli equation is then used to find the time dependency. In the derivation it is assumed that the surface elevation is small. The derivation shows that a horizontal acceleration only excite the odd numbered oscillation modes.

The above problem gives the following model of the slosh. The states are Position of the slide, velocity of the slide, Level of the liquid and the level rate. The input to the system is the analog signal to drive the slide. The output is the position of the slide.

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned}$$

where,

$$
\begin{aligned}
A &= diag\left(\begin{bmatrix} 0 & 1 \\ -{\omega_1}^2 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ -{\omega_2}^2 & 0 \end{bmatrix}, \ \ldots \ \begin{bmatrix} 0 & 1 \\ -{\omega_n}^2 & 0 \end{bmatrix} \right) \\
B &= \frac{-4\sqrt{2a^3}}{\pi^2} stack\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix} \ \begin{bmatrix} \frac{1}{9} \\ 0 \end{bmatrix} \ \ldots, \ \begin{bmatrix} \frac{1}{n^2} \\ 0 \end{bmatrix} \right) \\
C &= \frac{1}{g}\sqrt{\frac{2}{a}} augment\left(\begin{bmatrix} 0 & \cos\frac{\pi x}{a} \end{bmatrix}, \begin{bmatrix} 0 & \cos\frac{3\pi x}{a} \end{bmatrix}, \ \ldots, \ \begin{bmatrix} 0 & \cos\frac{n\pi x}{a} \end{bmatrix} \right) \\
D &= \frac{1}{g}(\sqrt{\frac{2}{a}} - x) - \frac{4a}{\pi^2 g} \sum_{i=1,iodd}^{n} \frac{1}{i^2} \cos\frac{i\pi x}{a}
\end{aligned}
$$

where a is the width of the container, h is the liquid depth, x is the position where the slosh is measured and ω_i is the oscillation frequency of the ith mode. The function *diag* gives a block diagonal matrix, the function *stack* joins the vectors vertically and *augment* joins the vectors horizontally. The damping is induced in the system from the system identification.

The choice of slosh model depends what the model should be used for. If we want to simulate the slosh, an advanced detailed model based on numerical solution of the Navier-Stokes equation is very useful. However, these models are very hard to use for controller design since they are very large and highly nonlinear. For the controller design we would like to have a simple model that captures the most important features of the slosh. If the model is too complex the optimal control problem is very hard to solve.

The model found using system identification techniques is similar to the model proposed by Venugopal and Bernstein [9] showed in Equation 3.1. Since the higher order modes only have little influence on the surface elevation, only the first oscillation mode has been included in the model. The direct term has been removed and damping is added.

The natural frequencies of the standing waves on an enclosed liquid container can be calculated from Equation 3.1

.

$$\omega_i = \sqrt{\frac{ig\pi \sinh \frac{ih\pi}{a}}{a \cosh \frac{h\pi}{a}}}. \tag{3.1}$$

where a is the width of the container, h is the liquid depth, x is the position where the slosh is measured and ω_i is the oscillation frequency of the ith mode.

When this formula is applied to the tank used for this thesis, the fundamental frequency is obtained as 1.506Hz. This is an excellent agreement with the value obtained from system identification. The damping factor and the gain associated with the fundamental mode, as well as the additional dynamics associated with the liquid sloshing and the positioning slide are best captured by experimental means.

3.2 Frequency Domain Tests

Three different tests were carried out in the frequency domain which have been explained in the following sections.

3.2.1 Frequency domain analysis using HP Spectrum Analyzer

The HP Spectrum analyzer as described in Chapter 2 is a device to record the frequency response of the plant. The analog input, for the movement of slide, was fed with a swept sine wave from 0.3 Hz to 8 Hz frequency and the response was recorded. Figure 3.1 shows the various trials and the repeatability of the response. This frequency plot was then used to determine a 2nd order plant model for the tank.

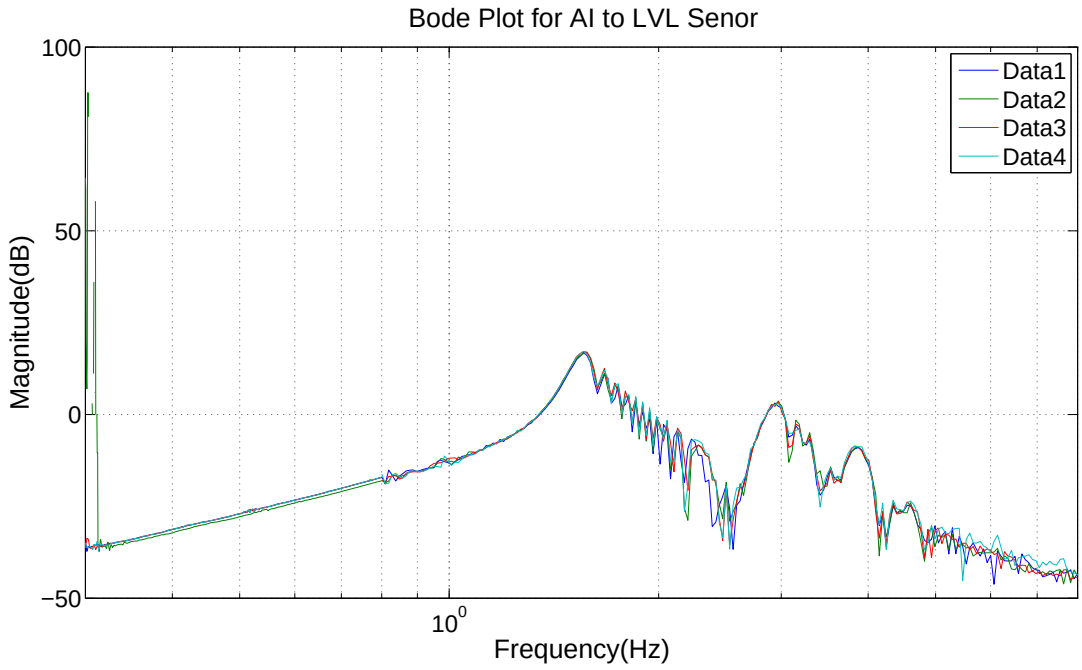


Figure 3.1: Frequency response for the tank - Input is Analog voltage, Output is Level sensor output

3.2.2 Individual frequency tests

These tests were carried out to validate the results from the spectrum analyzer. The slide was injected with an input sine wave signal at various frequencies, including the resonance frequency. The output from the level sensor was recorded using an oscilloscope. With a known input signal and the recorded output signal, the magnitude (in dBs) was calculated. This was compared with the Bode plot from the spectrum analyzer. It was found that both had the same magnitude. This validated the results from the signal analyzer.

3.2.3 Frequency domain test using Chirp signal

The other test to determine the mathematical model of the plant was to record the real time input signal and the output signal. Here, a 1 V chirp signal with frequencies between 0.1 Hz and 10 Hz and lasting 30 seconds was applied to the analog input port

of the slide.The resulting slide motion and the water level oscillations were recorded at a sample rate of 1 kHz. The data was then used for time-domain analysis and parametric estimation using *MATLAB/Simulink System Identification Toolbox*.

3.3 Parametric model for the plant

This section describes the various steps taken to derive a model for the plant using System Identification Toolbox.

The experimental input and output were recorded using real time data acquisition system. The data was imported into the *Matlab's System Identification Toolbox*. Using this, a parametric model was estimated which closely resembles the actual plant. These models were then used to derive the transfer function for the plant. Figure 3.5 shows the frequency response for the plant.

A 2nd order transfer function between the analog input and slide position was identified using the output error method in the *MATLAB System Identification Toolbox*. Conversion to continuous time with zero-order hold method yielded the desired transfer function:

$$\frac{Y(s)}{U(s)} = \frac{421.43}{s^2 + 8.2117s + 0.2016} \tag{3.2}$$

where *U(s)* represents the analog voltage and *Y(s)* represents the displacement from an arbitrary referenced mark, in mm. Figure 3.2 shows the correlation between modeled and experimental data. For consistency with the physics of the system, the s^0 coefficient was forced to zero, without significant loss of prediction accuracy. Without this adjustment, a zero initial velocity together with a non-zero initial position would cause y to change under zero input, due to stiffness term 0.2016. Such behavior is not observed in the physical system.

.

A 4th-order transfer function of the tank subsystem between slide position and liquid level was first obtained. This function was subsequently simplified by reduction to 2nd-order and elimination of a large non-minimum phase zero whose presence did not significantly affect the accuracy of predictions. The tank transfer function in continuous time is

$$\frac{L(s)}{Y(s)} = \frac{0.047165s^2 + 0.0632s - 0.052202}{s^2 + 0.19854s + 88.439} \tag{3.3}$$

where L represents the water level displacement expressed in sensor volts, relative to the baseline water level. Note that the pair of complex poles have a natural frequency of 1.497 Hz, which is very close to the theoretical value of 1.506 Hz obtained based on tank geometry. Figure 3.3 shows the modeled plant for position to level sensor versus the actual data. The two real zeroes close to the origin are forced to zero with out loss of accuracy of prediction. As with the slide, this is done to maintain consistency with the system physics; a steady level of zero should be maintained if the tank moves at steady velocity, hence the double differentiation at the input. Also note that due to the nature of magnetostrictive sensor, the transfer function has a negative DC gain (voltage decreases as the level increases and viceversa.)

3.4 Time Domain Tests

Modeling is the key to developing a control for the system and hence the model needs to be verified for accuracy and closeness to the actual plant. The time domain tests were carried out to compare the step response for the actual and the plant model.

3.4.1 Step response analysis

An overall 4th order model was obtained for the cascaded state space realization of the slide and the tank transfer functions.If $(A_s, B_s, C_2, 0)$ and (A_t, B_t, C_1, D_1) denote

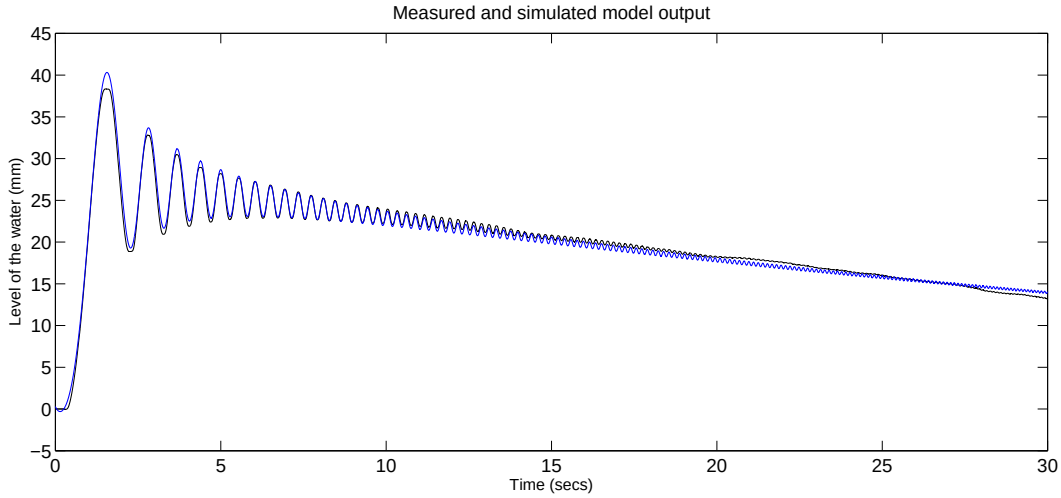


Figure 3.2: Input signal to slide position (mm) - modeled vs recorded output

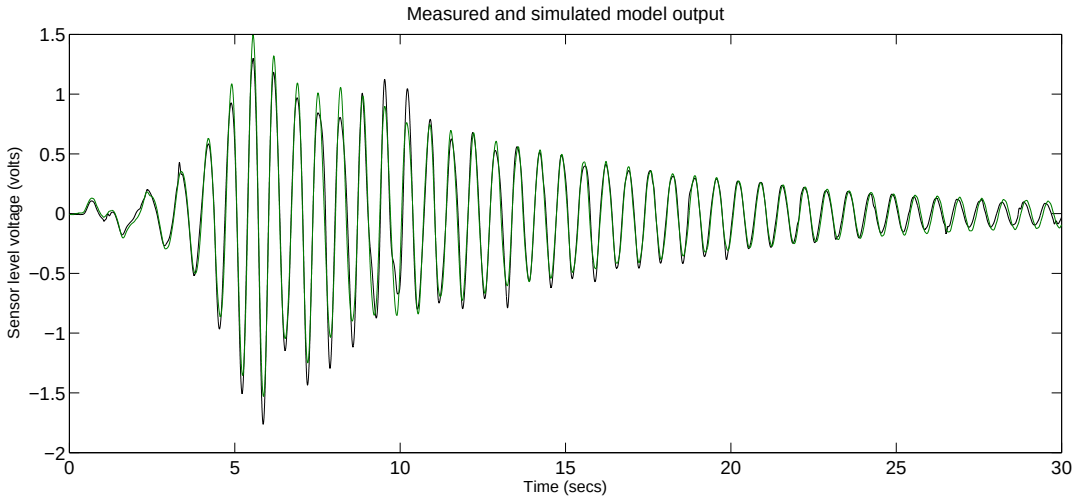


Figure 3.3: Slide position to level sensor (volts) - modeled vs recorded output

the slide and tank systems, respectively, then the cascaded model has realization $(A, B, C, 0)$, with

$$A = \begin{bmatrix} A_s & 0 \\ B_t C_2 & A_t \end{bmatrix} = \begin{bmatrix} 0.0000 & 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & -8.2117 & 0.0000 & 0.0000 \\ 1.0000 & 0.0000 & -0.1985 & -9.4042 \\ 0.0000 & 0.0000 & 9.4042 & 0.0000 \end{bmatrix}$$

$$B = \begin{bmatrix} B_s \\ 0 \end{bmatrix} = \begin{bmatrix} 0.0000 \\ 421.4271 \\ 0.0000 \\ 0.0000 \end{bmatrix}$$

$$C_s = \begin{bmatrix} C_2 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} D_1 C_2 & C_1 \end{bmatrix} = \begin{bmatrix} 0.0522 & 0.0000 & -0.0104 & -0.4908 \end{bmatrix}$$

Figure 3.4 shows the prediction performance of the identified model on the basis of chirp response. As can be seen from Figure 3.5, the low band-width model captures only the fundamental resonance near 1.5 Hz, but still produces good predictions in the time-domain. The overall 4th order model offers reasonable prediction given its simplicity. It is readily verified that the model is marginally stable, controllable, observable and is not numerically ill-conditioned.

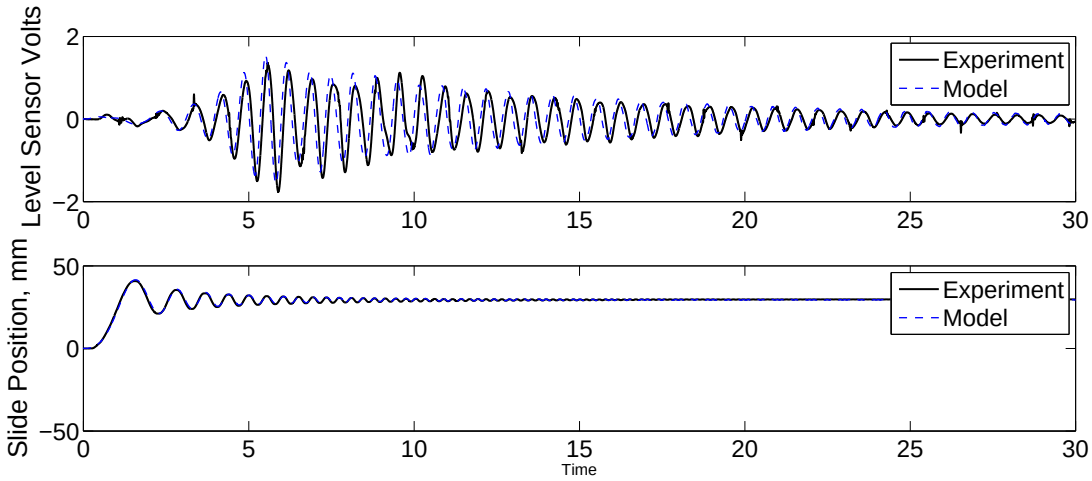


Figure 3.4: Time-domain model prediction, Step response

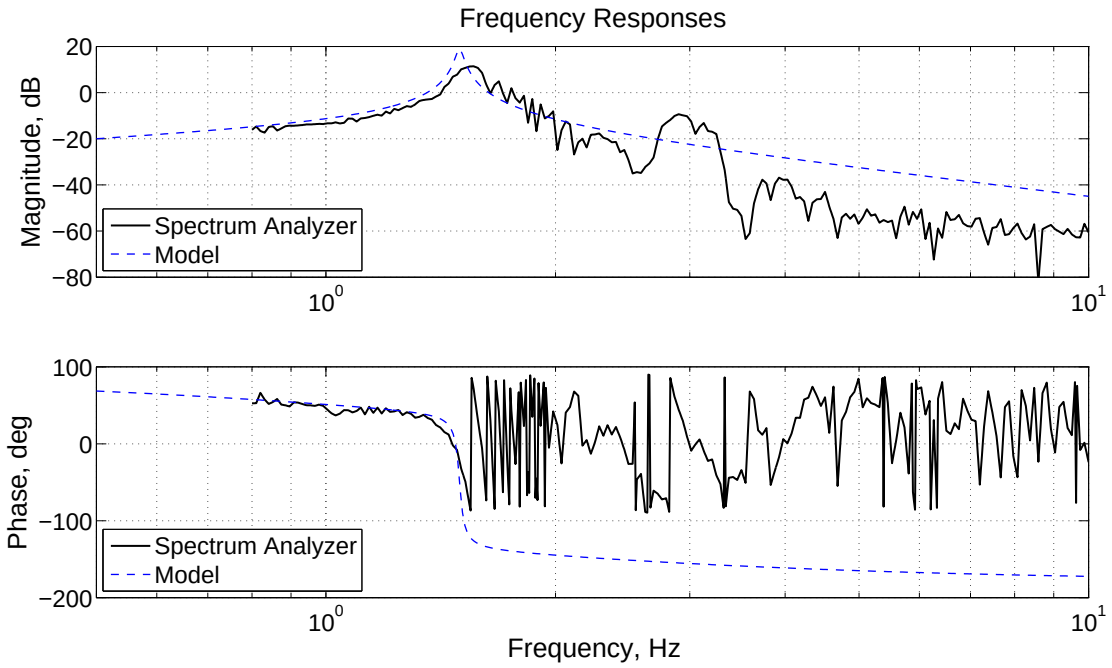


Figure 3.5: Time-domain model prediction; Step response

CHAPTER IV

SLIDING MODE CONTROL

In formulation of control law there will typically be discrepancies between the actual plant and the mathematical model developed for controller design. This mismatch may be due to unmodeled dynamics, variations in parameters or the approximation of complex plant behavior by a straightforward model. This mismatch requires a robust controller. SMC is a particular type of *variable structure control* and is robust in nature.

Variable structure control systems (VSCS) evolved from the pioneering work in Russia of Emel'yanov and Barbshin in early 1960's [2]. In SMC, VSCS are designed to drive and then constrain the system states to lie within a neighborhood of the switching function. There are two main advantages to this approach:

- the dynamic behavior of the system may be tailored by a particular choice switching function.
- the closed loop response becomes totally insensitive to a matched class of uncertainty.

The latter makes it a robust control methodology.

An example as a result of using variable structure law is given below: let,

$$u(t) = \begin{cases} -1 & \text{if } s(y, \dot{y}) < 0 \\ 1 & \text{if } s(y, \dot{y}) > 0 \end{cases} \tag{4.1}$$

where the *switching function* is defined by

$$s(y, \dot{y}) = my + \dot{y} \tag{4.2}$$

Equation 4.1 can be written as

$$u(t) = -sgn(s) \tag{4.3}$$

where sgn is the sign function.

4.1 Sliding Mode Design

The design approach is first to select the coefficients of the sliding hyperplane according to the desired performance in the ideal sliding mode. The coefficients of the sliding hyperplane are chosen following any standard technique like the pole placement or the linear-quadratic (LQ) optimization. The LQ method of Utkin and Young was chosen for its simplicity and efficacy [2]. A feedback control law is then derived to drive the plant's state trajectory on the sliding manifold in the state space and maintain the trajectory on the surface for a subsequent time. The feedback control law is known as the *switching control law*. Ideally, once the sliding manifold has been intercepted, the switching control maintains the plant's state trajectories on the surface and the state trajectories slide along the surface.

Consider a single-input linear plant in the state space form

$$\dot{x} = Ax + B(u(t) + \zeta(t)) \tag{4.4}$$

where $\zeta(t) \epsilon [-\bar{\zeta}, \bar{\zeta}]$ is a matched uncertainty, with $\bar{\zeta} > 0$.

Let $s : \mathbb{R}^n \rightarrow \mathbb{R}^m$ be a linear function represented as

$$s(t) = Sx(t) \tag{4.5}$$

where $S \in \mathbb{R}^{m \times n}$ is of full rank and let S be the switching function defined by $S = x \in \mathbb{R}^n : s(x) = 0$.

For describing the equivalent control, let the matched uncertainty $\zeta(t) = 0$. For ideal sliding motion $Sx(t) = 0$ and $\dot{s}(t) = S\dot{x}(t) = 0$ at any time t, this gives us,

$$S\dot{x}(t) = SAx(t) + SBu(t) = 0 \tag{4.6}$$

We assume that square matrix SB is nonsingular, which gives us the equivalent control

$$u_{eq}(t) = -(SB)^{-1}SAx(t) \tag{4.7}$$

This gives us the ideal dynamics under the sliding motion by substituting Equation 4.7 in Equation 4.4 with $\zeta(t) = 0$.

$$\begin{aligned} \dot{x}(t) &= (I_n - B(SB)^{-1}S)Ax(t) \quad for\ all\ t \geq t_s\ and\ Sx(t_s) = 0 \\ \dot{x}(t) &= A_{eq}x(t) \end{aligned}$$

where, $A_{eq} = (I_n - B(SB)^{-1}S)A$ also, $P_s = (I_n - B(SB)^{-1}S)$ is known as the projection operator [2].

Considering the uncertainty, we can rewrite Equation 4.7 as

$$u_{eq}(t) = -(SB)^{-1}(SAx(t) + SB\zeta(t)) \tag{4.8}$$

From Equation 4.8 the equivalent control is dependent on the uncertainty. But SMC has a key property, it is invariant towards the matched uncertainty. This is shown by substituting Equation 4.8 in Equation 4.4, which gives

$$\dot{x}(t) = P_sAx(t) + P_sB\zeta(t) \tag{4.9}$$

where P_s is a projection operator.

It turns out (by simple evaluation) that $P_sB = 0$ which gives us the dynamics of the sliding motion as

$$\dot{x}(t) = P_sAx(t) = A_{eq}x(t) \tag{4.10}$$

From Equation 4.10, the sliding motion is invariant to the matched uncertainty.

4.1.1 Sliding Mode Control Regulator

To design the SMC, we represent Equation 4.4 in a canonical form given below [2].

$$\begin{aligned} \dot{z_1}(t) &= A_{11}z_1(t) + A_{12}z_2(t) \\ \dot{z_2}(t) &= A_{21}z_1(t) + A_{22}z_2(t) + B_2u(t) \end{aligned}$$

The switching function also is expressed as

$$s(t) = S_1(t)z_1(t) + S_2z_2(t) \tag{4.11}$$

Without losing the generality, we can consider $S_2 = 1$.

$$s(t) = S_1(t)z_1(t) + z_2(t) \tag{4.12}$$

The change of matrix is defined by the orthogonal matrix T_r such that

$$z(t) = T_rx(t) \tag{4.13}$$

This transformation requires controllable pair (A, B), where B is of full rank.

The matrix sub-block, namely $A_{11}, A_{12}, A_{21}, A_{22}, B_2, S_1$ and S_2 can be obtained

as

$$\begin{aligned} T_r A(T_r)^T &= \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \\ T_r B &= \begin{bmatrix} 0 \\ B_2 \end{bmatrix} \\ S(T_r)^T &= \begin{bmatrix} S_1 & 1 \end{bmatrix} \end{aligned}$$

A QR decomposition is carried out on the input distribution matrix to obtain transformation matrix T_r and in turn obtain the sub-blocks. The $MATLAB$ program from decomposition is shown in Appendix 1.4.

Reachability Condition

The mathematical expression given by Equation 4.3 satisfies the reachability condition in the domain $\Omega = (y, \dot{y}) : m \mid \dot{y} \mid < 1$. The condition for reachability can be given by $\dot{s}s < 0$. Essentially this only guarantees the sliding surface is reached asymptotically. A stronger condition, guaranteeing an ideal sliding motion, is the η-reachability condition given by

$$\dot{s}s \leq -\eta \mid s \mid \tag{4.14}$$

where η is a positive constant known as switching gain. The speed of convergence to the sliding surface are directly influenced by the switching gain η.

Now we can define the control law as

$$u = u_{eq} - \eta sgn(s) \tag{4.15}$$

where u_{eq} is given by Equation 4.7.

Design of Switching plane

In this thesis, the *quadratic minimization* approach has been taken to design the switching hyperplane. This method was proposed by Utkin and Young in 1978. This method enables us to place desirable weights to particular elements. Let us consider the problem of minimizing the quadratic performance index

$$J = \frac{1}{2}\int_{t_s}^{\infty} x(t)^T Q x(t) dt \tag{4.16}$$

where Q is both symmetric and positive definite and t_s is the time at which sliding motion commences. The aim is to minimize Equation 4.16 subject to the system equation given by Equation 4.4. It is assumed that $x(t_s)$ is a known initial condition. The matrix Q from Equation 4.16 is transformed compatibly with z:

$$T_r Q T_r^T = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \tag{4.17}$$

where $Q_{21} = Q_{12}^T$. Equation 4.16 is now expressed in terms of the z co-ordinate system as

$$J = \frac{1}{2}\int_{t_s}^{\infty} z_1^T Q_{11} z_1 + 2 z_1^T Q_{12} z_2 + z_2^T Q_{22} z_2 dt \tag{4.18}$$

Re-arranging the Equation 4.18 and Equation 4.4, the problem can be interpreted as a standard linear quadratic optimal state regulator problem [2].

Boundary Layer Control and LQ Tuning

The chattering phenomenon is generally perceived as motion which oscillates about the sliding manifold. The possible mechanisms which produce such a motion is in the absence of switching non-idealities such as delays. The most commonly cited approach to reduce the effects of chattering has been the so called piecewise linear or smooth approximation of the switching element in a boundary layer of the sliding manifold. The signum function was replaced by a linear approximation (saturation

function) with a slope of 100 ($\phi = 0.01$). The chattering is not eliminated but reduced by this linear approximation.

We assume the equal weights for each input and hence we have the value of $R = 1$. We define $Q = diag([1\ 1\ 1e-4\ 1e-4])$ with weights given to the states. At this point in design, $\eta = 7$ was chosen by simulation to meet the a settling time of 1.8 seconds for 400 mm transfer. The *MATLAB* program for Linear Quadratic cost function is shown in Appendix 1.4. This program gives the values for sliding manifold, $S = [0.0381, 0.0024, 0.3042, -0.1197]$.

4.2 Simulation and Results

4.2.1 Model Simulation

The above design of SMC assumes that the target point to reach for the states is the point of origin from any initial condition. But for the experiment considered in this thesis, the target point is different from origin and hence Equation 4.5 for the Sliding manifold becomes

$$s = S(x(t) - \bar{x}(t)) \tag{4.19}$$

where $\bar{x}$ is the target point for the states to reach. Equation 4.7 for equivalent control can be written as

$$u_{eq}(t) = -(SB)^{-1}SA(x(t) - \bar{x}(t)) \tag{4.20}$$

Figure 1.1 and Figure 1.2 (in the appendix) shows the *Simulink model* for digital SMC. Figure 4.1 shows the simulation results. As we can see from the subplot, the target position is 400 mm and not the origin. The system reaches the target position in as little as 1.8 seconds without excessive slosh. The simulation shows that the control signal was not saturated and the slosh is under $25mm$. The limits for the slosh are constrained by the container depth. The model was also simulated for an

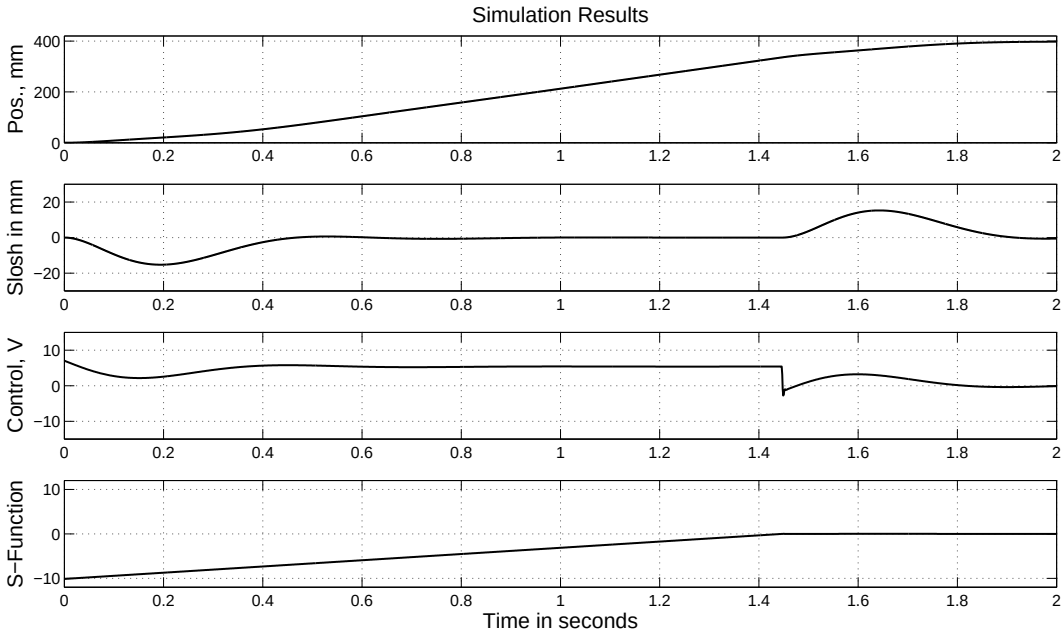


Figure 4.1: Simulation Result, SMC

external disturbance, which is visible at about 1.4 seconds after the system is at steady state. The control action changes and accounts for it.

4.2.2 Real-Time digital SMC

A digitized SMC controller for regulation to a reference state was implemented in a *dSPACE* rapid prototyping system. Level rates were low-pass filtered using a Butterworth op-amp circuit. Since the state of the tank were not level and level rate, a simple algebraic transformation was used to perform the the conversion from level and level rate to tank states. Figure 4.2 shows the actual measurement plots as compared to the simulations. It is visible that the model behaves very closely to the actual plant. This plot is with a sampling rate of $10KHz$ and a linear approximation of saturation function with a slope of 100 ($\phi = 0.01$) . At this rate we see a lot of control action called *chattering*, to keep the system on the sliding manifold. Figure 4.3 shows the final run with an improved control action and also with less sloshing.

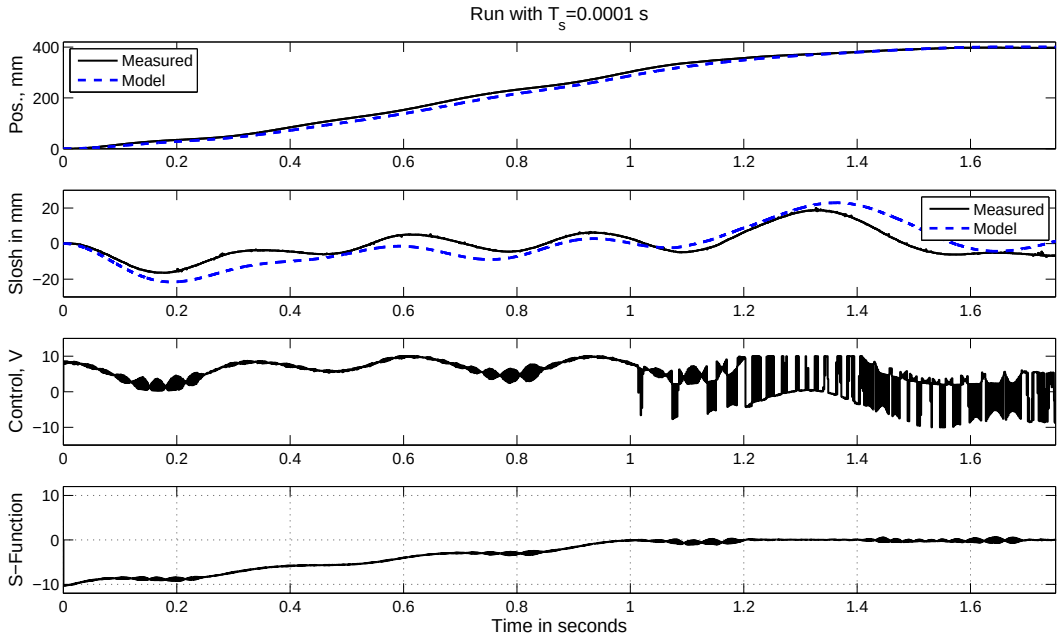


Figure 4.2: Realtime, Trial 1 - Model output from the realtime system and the experimental setup

This run was carried out at $1KHz$ of sampling rate and $\phi = 0.5$. As it can be seen the sliding variable is robustly driven to zero, despite modeling errors from the neglected second mode. There is lot of activity at $3Hz$ as a manifestation of the controller's effort to attenuate the second mode by using whole tank as an actuator. This effect would be impossible to obtain using an open-loop approach based on imperfect plant models.

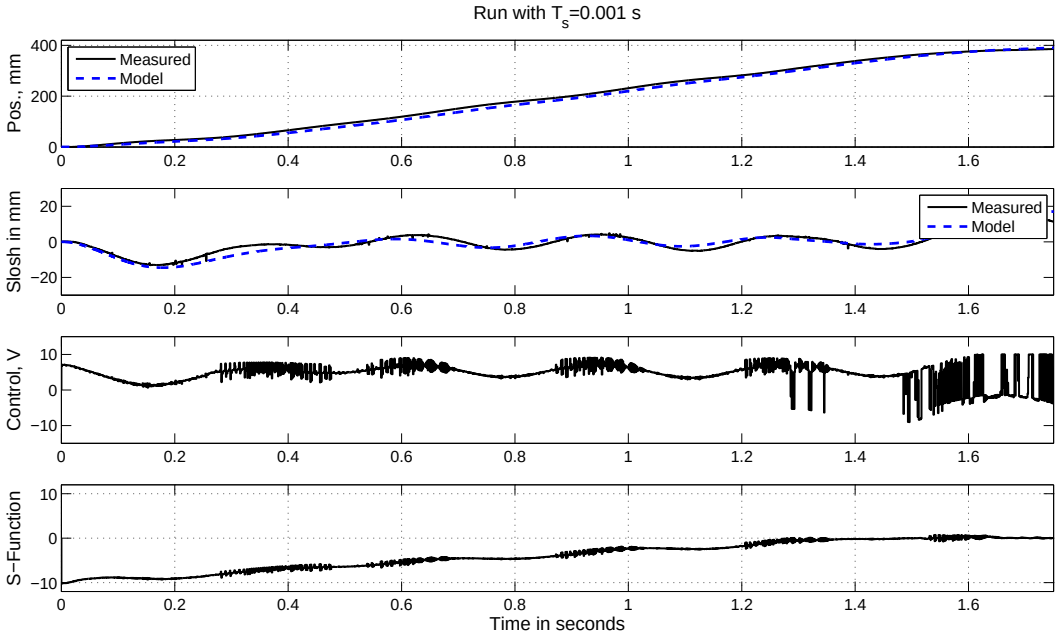


Figure 4.3: Realtime, final - Model output from the realtime system and the experimental setup

CHAPTER V

CONCLUSIONS AND FUTURE WORK

This Chapter describes the inference obtained from the research work of this thesis and also sets the path for future work. The conclusions are based on the simulation results and experiments carried out on the system.

- The simulations show that the 4th order model obtained by system identification technique was close to the actual plant. It can be seen that the model offers reasonable prediction given its simplicity.

- It is also visible that the system identification technique gives a very close prediction of the fundamental frequency when compared with the mathematical equation (3.1).

- The SMC also offers very robust control, considering the model errors induced by neglecting the second mode.

- In this thesis we can also see that the controller uses the whole tank as an actuator to attenuate the second mode near $3Hz$. This effect is impossible to obtain using open loop approach, which has been fostered by Grundelius [3] in his thesis. He relies on the accurate mathematical model of the liquid container and cannot accommodate disturbances or model errors to obtain the

acceleration profile.

- Future work can be carried out using an observer based control which will allow us to predict the states accurately eliminate the uncertainty in measuring the level rate of the fluid in the tank.

BIBLIOGRAPHY

[1] Hakan Akyildz and N. Erdem U nal, *Sloshing in a three-dimensional rectangular tank: Numerical simulation and experimental validation*, Ocean Engineering - 33 (2006).

[2] Christopher Edwards and Sarah K. Spurgeon, *Sliding mode control - theory and applications*, Taylor and Francis.

[3] Mattias Grundelius, *Motion control of open containers with slosh contraints*, Master's thesis, Lund Institute of Technology,Lund,Sweden, 1998.

[4] K.M.Kelkar and S.V.Patankar, *Numerical method for the prediction of free surface flows in domains with moving boundaries*, Numerical Heat Transfer, Part B: Fundamentals **31** (1997), no. 1, 387–399.

[5] Lennart Ljung, *System identification - theory for user*, 2nd ed., PTR Prentice Hall, 1999.

[6] S.K.Bhattacharya N.C.Pal and P.K.Sinha, *Experimental investigation of slosh dynamics of liquid filled containers*, Experimental Mechanics (Vol.41, No.1, Mar 2001).

[7] H. Richter and K.B. Karnik, *High-performance motion control with slosh: A constrained sliding mode approach*, 1st ASME Dynamic Systems and Control Conference, Ann Arbor, Michigan, 2008.

[8] Armenio V and M.La Rocca, *On the analysis of sloshing of water in rectangular containers: Numerical study and experimental validation.*, Ocean Engineering **23** (1996), no. 8, 705–739.

[9] R Venugopal and D.S.Berstein, *State space modeling and active control of slosh*, In Proceedings of the 1996 IEEE International Conference on Control Applications (1996), 1072–1077.

[10] V.J.Romero and M.S.Ingber, *A numerical model for 2-d sloshing of pseudo-viscous liquids in horizontally accelerated rectangular containers*, Boundary Elements XVII (1995), 567583.

APPENDICES

APPENDIX A

MATLAB PROGRAMS AND SIMULATION MODELS

1.1 Plant identification

```
%identify

load ai2lvl3

Ts=ai2lvl3.Capture.SamplingPeriod;
fs=1/Ts;
t=ai2lvl3.X.Data;
input=ai2lvl3.Y(1).Data;
output=ai2lvl3.Y(2).Data;

%Remove mean
input=input-mean(input);
output=output-mean(output);

%Compute and display FFT of input and output (for referential use)
```

```
N=length(input);
df=1/Ts/N;
f=[0:N-1]*df;
values_in=fft(input);
values_out=fft(output);
subplot(2,1,1)
stem(f,abs(values_in));
subplot(2,1,2)
stem(f,abs(values_out));

%Obtain spectral estimate without filtering and superimpose to spectrum
%analyzer estimate (including gain shift due to dSPACE/Simulink settings)
load master_spectr
shift=22.59;
[Txy, F]=tfestimate(input,output,[],[],1024,1/Ts);
figure(2)
subplot(2,1,1)
semilogx(F,20*log10(abs(Txy)));
hold on
semilogx(Freq1,MagdB1+22.59,'r',Freq2,MagdB5+22.59,'r')
grid
subplot(2,1,2)
semilogx(F,angle(Txy)*180/pi);
hold on
semilogx(Freq1,Ph1,'r',Freq2,Ph5,'r')
grid

%Now low-pass filter and see if there's improvement
fc=10;
```

```
Wp=fc/(0.5*fs);
[B,A]=cheby1(4,1,Wp);
input_f=filter(B,A,input);
output_f=filter(B,A,output);
[Txy, F]=tfestimate(input_f,output_f,[],[],1024,1/Ts);
figure(3)
subplot(2,1,1)
semilogx(F,20*log10(abs(Txy)));
hold on
semilogx(Freq1,MagdB1+22.59,'r',Freq2,MagdB5+22.59,'r')
grid
subplot(2,1,2)
semilogx(F,angle(Txy)*180/pi);
hold on
semilogx(Freq1,Ph1,'r',Freq2,Ph5,'r')
grid


%NOTE:  The low frequency slope is close to +40dB/dec, as it should be
%theoretically (steady level proportional to steady acceleration of base)
%It's not exactly +40 due to slide dynamics, float dynamics and meas.
%errors

%The identified model must have an s^2 in the numerator!!!

%What if we pass the AI input through slide dynamics and double diff and
%work from there?

%These are the identified parameters for the slide (see files: ai2lvdt_tf.m
```

```
%and check.mdl
%The units are AI volts to LVDT volts

num=[1108.4 0 0];   %includes double differentiation
den=[1 53.333 877.91];
td=0.038;  %delay time

%Assemble matrix for use with FromWorkspace block
load ident_from_chirp
simin=[t' input'];
%NOTE: The simulation MUST use a fixed time step, so that the resulting
%acceleration data is compatible with the experimental sample rate and time
%vector.


%A fit was obtained using the sys id toolbox
load identified_ss
%remove noise input
[Ad,Bd,Cd,Dd]=ssdata(pss10);
sys_d=ss(Ad,Bd,Cd,Dd,Ts);
%Balance the realization
sys_d=balreal(sys_d);
[Ad,Bd,Cd,Dd]=ssdata(sys_d);
%This model is fortunately stable, controllable
% and observable
%Convert to continuous-time using 'zoh'
%Since it has negative real poles, the order will increase  by one
sys_c=d2c(sys_d,'zoh',Ts);
```

```
rank(ctrb(sys_c))
rank(obsv(sys_c))
%Conversion to continuous time using ZOH appears to introduce massive loss of
%controllability/observability

%Perform balanced model reduction to 6th order

sys_cbalred=balred(sys_c,6);
[A,B,C,D]=ssdata(sys_cbalred);

rank(ctrb(sys_cbalred))
rank(obsv(sys_cbalred))

%System has a very large rhp zero, a large lhp zero and a high-frequency complex pole
%Be careful to preserve the gain
[z,p,k]=ss2zp(A,B,C,D);
%large zeros and fast poles appears at the top
[A,B,C,D]=zp2ss(z(3:6),p,k*abs(z(1)*z(2)));
sys_cbalred=ss(A,B,C,D);
rank(ctrb(sys_cbalred))
rank(obsv(sys_cbalred))

%Throw in double derivative
tank_tf=tf(sys_cbalred);
sys_tank=series(tank_tf,tf([1 0 0],1));
%Join to slide model, including delay
s=tf('s');
sys_slide=exp(-td*s)*1108.4/[s^2+53.333*s+877.91];
sys_final=series(sys_slide,sys_tank);
```

```
%Draw bode plot and superimpose to spectral data
w=logspace(log10(0.3*2*pi),log10(10*2*pi),500);
[mag,ph]=bode(sys_final,w);
figure(2)
subplot(2,1,1)
semilogx(w/2/pi,20*log10(mag(1,:)),'c--')
```

1.2 Mathematical model for tank

```
clear all;

%% Changing the denominator only.

%Starting with the tank transfer function achieved earlier.

%Fine tuning of frequency and damping ratio to match with the experimental
%data.

load sys_tank;

%extracting num and den from the sys.
[Num_tank,Den_tank] = tfdata(sys_tank,'v');

%finding roots for the denominator to adjust the damping ratio and freq.
r = roots(Den_tank);

%seperating roots into two polynomials
r1(1,1) = r(1,1);
r1(2,1) = r(2,1);

r2(1,1) = r(3,1);
r2(2,1) = r(4,1);

r3(1,1) = r(5,1);
r3(2,1) = r(6,1);
```

```
poly1 = poly(r1);

poly2 = poly(r2);

poly3 = poly(r3);


%Above data gives us the freq1,zeta1,freq2,zeta2 as follows

% zeta1 = 0.2024;

% freq1 = 281.6878;

% zeta2 = 0.01371;

% freq2 = 9.1479;

% zeta3 = 0.1447;

% freq3 = 17.795;


%Changing the values to suit the experimental data

zeta1 = 0.2024;

freq1 = 281.6878;

zeta2 = 0.02971;

freq2 = 9.1379;  %9.7479

zeta3 = 0.0080;

freq3 = 17.050;  %18.350


poly1mod = [1 2*zeta1*freq1 freq1^2];

r1mod = roots(poly1mod);

poly2mod = [1 2*zeta2*freq2 freq2^2];

r2mod = roots(poly2mod);

poly3mod = [1 2*zeta3*freq3 freq3^2];

r3mod = roots(poly3mod);


rmod = [r1mod;r2mod;r3mod];
```

```
Den_tank_new = poly(rmod);
gain_num = (freq1^2*freq2^2*freq3^2)/(281.6878^2*9.1479^2*17.795^2);
% sys_tank_new = tf(gain_num*Num_tank,Den_tank_new);
%
% load sys_slide;
% [Num_slide,Den_slide] = tfdata(sys_slide,'v');
% sys_final=series(sys_slide,sys_tank_new);
%
% %Draw bode plot and superimpose to spectral data
% load master_spectr;
%
% figure(1);
% w=logspace(log10(0.3*2*pi),log10(10*2*pi),500);
% [mag,ph] = bode(sys_final,w);
% semilogx(w/2/pi,20*log10(mag(1,:)),'c--')
% hold on
% semilogx(Freq1,MagdB1+22.59,'r',Freq2,MagdB5+22.59,'r');
% hold off;

%% Changing the numerator also

%finding roots for the numerator to adjust the damping ratio and freq.
nr = roots(Num_tank);

%seperating roots into polynomials
nr1(1,1) = nr(1,1);
nr1(2,1) = nr(2,1);

nr2(1,1) = nr(3,1);
```

```
nr2(2,1) = nr(4,1);


nr3(1,1) = nr(5,1);
nr3(2,1) = nr(6,1);



npoly1 = poly(nr1);
npoly2 = poly(nr2);
npoly3 = poly(nr3);

%Above data gives us the freq1,zeta1,freq2,zeta2 as follows

% nzeta2 = 0.1282;
% nfreq2 = 14.5589;
% nzeta3 = -0.1966;
% nfreq3 = 30.0088;

% Changing the values to suit the experimental data
nzeta2 = 0.0010;
nfreq2 = 16.10; %16.10
nzeta3 = -0.0966;
nfreq3 = 25.0088;

npoly2mod = [1 2*nzeta2*nfreq2 nfreq2^2];
nr2mod = roots(npoly2mod);
npoly3mod = [1 2*nzeta3*nfreq3 nfreq3^2];
nr3mod = roots(npoly3mod);

nrmod = [nr1;nr3mod;nr2mod];
```

```
gain_den = (nfreq2^2*nfreq3^2)/(14.5589^2*30.0088^2);
Num_tank_new = 450*gain_num*poly(nrmod);%multiply 450
Den_tank_new = gain_den*poly(rmod);

sys_tank_new = tf(Num_tank_new,Den_tank_new);

load sys_slide;
[Num_slide,Den_slide] = tfdata(sys_slide,'v');
sys_final=series(sys_slide,sys_tank_new);

td=0.038;  %delay time (identified from the data see identify.m)
%Draw bode plot and superimpose to spectral data
load master_spectr;

figure(1);
w=logspace(log10(0.3*2*pi),log10(10*2*pi),500);
[mag,ph] = bode(sys_final,w);
semilogx(w/2/pi,20*log10(mag(1,:)),'c--')
hold on
semilogx(Freq1,MagdB1+22.59,'r',Freq2,MagdB5+22.59,'r');
hold off;

%% To compare the data.

load ai2lvl3

TsLvl=ai2lvl3.Capture.SamplingPeriod;
fsLvl=1/TsLvl;
```

```
tLvl=ai2lvl3.X.Data;
inputLvl=ai2lvl3.Y(1).Data;
outputLvl=ai2lvl3.Y(2).Data;

%% Assemble matrix for use with FromWorkspace block
load ident_from_chirp
simin=[tLvl' inputLvl'];

%% Plotting to compare experimental and simulated data.

sim('systemmodel');
figure(2);
plot(tLvl,outputLvl,Time,Waterlevelvolts+3.42,'r');
```

1.3 Reduced order model

```
%load_test_noaccel_reduced
%Script to reduce the model and validate against experimental
%data. 02/12/08

%Load slide model in state-space, with position-velocity states

load slidematrices_w_tank %Aslide_w_tank Bslide_w_tank Cslide_w_tank (Dslide_w_tank=0
%Need to tweak Aslide_w_tank(2,1) to zero, for compatibility with system
%physics (slide should not move under zero input, zero initial velocity and
%arbitrary initial position).
Aslide_w_tank(2,1)=0;

%Load identified tank model from position to relative level (taken at
%sensor output=2.66V

load identif_pos2level  %oe441‘‘

Ts=0.001;

[Ad,Bd,Cd,Dd]=ssdata(oe441);
sys_d=ss(Ad,Bd,Cd,Dd,Ts);
sys_c=d2c(sys_d,’zoh’,Ts);
%Balance the realization
sys_cbal=balreal(sys_c);
[Atank,Btank,Ctank_level,Dtank]=ssdata(sys_cbal);

%Try eliminating the large non-minimumphase zero in the tank to simplify
```

```
%future design. Also eliminate second resonant mode.
[z,p,k]=ss2zp(Atank,Btank,Ctank_level,Dtank);
%large zero appears at the top of z
[Atank,Btank,Ctank_level,Dtank]=zp2ss(z(2:end),p(3:4),-k*z(1)/9405);

%New modification: Moved the two zeroes to the origin for consistency with
%physics of the system.  3/18/08. Some deterioration in prediction quality
%is seen.

[Atank,Btank,Ctank_level,Dtank]=zp2ss([0 0],p(3:4),k*z(1)*z(2)*z(3)/9405);

tanksys_no_z=ss(Atank,Btank,Ctank_level,Dtank);

sys_slide=ss(Aslide_w_tank,Bslide_w_tank,Cslide_w_tank,0);
sys_final=series(sys_slide,tanksys_no_z);

%Display frequency fit

w=logspace(log10(0.3*2*pi),log10(10*2*pi),500);
[mag,ph]=bode(sys_final,w);

load aivolt2levelrun
fsize=12;                                          %font size for all labels
lw1=1.5;                                            %line width
lw2=1;                                               %line width
subplot(2,1,1)
semilogx(f,20*log10(abs(sqrt(real_part.^2+imag_part.^2))),'k','Linewidth',lw1);
hold on
semilogx(w/2/pi,20*log10(mag(1,:)),'k--','Linewidth',lw2)
```

```
ylabel('Magnitude, dB','Fontsize',fsize)
title('Frequency Responses','Fontsize',fsize)
legend('Spectrum Analyzer','Model','Location','SW')
grid
axis([0.5 10 -80 20])
subplot(2,1,2)
semilogx(f,atand(imag_part./real_part),'k','Linewidth',lw1)
hold on
semilogx(w/2/pi,ph(1,:),'k--','Linewidth',lw2);
grid
xlabel('Frequency, Hz','Fontsize',fsize)
ylabel('Phase, deg','Fontsize',fsize)
legend('Spectrum Analyzer','Model','Location','SW')
axis([0.5 10 -200 100])

%% To compare the data.
%Check for model prediction capabilities in the time domain:
load runflat
%simin in this file contains smoothed slide position information

%Using the position information consistent with Aslide(2,1)=0 modification
load run_w_filters
slide_pos=run_w_filters.Y(5).Data';

simin2=[t in]; %the 1 Vpp chirp from 0.1 to 10 Hz, 30 seconds long
simin3=[t level];  %level data as captured (no filtering)
simin4=[t slide_pos];   %position data as captured (no filtering)
sim('test_model')
figure(2);
```

```
subplot(2,1,1)
plot(t,level_filt,'k','Linewidth',lw1)
hold on
plot(tsim,level_volts,'k--','Linewidth',lw2)
ylabel('Level Sensor Volts','Fontsize',fsize);
legend('Experiment','Model');

subplot(2,1,2)
plot(t,slide_pos,'k','Linewidth',lw1)
hold on
plot(t,pos_mm,'k--','Linewidth',lw2)
ylabel('Slide Position, mm','Fontsize',fsize);
legend('Experiment','Model');

xlabel('Time','Fontsize',fsize);

%Now obtain the cascade realization

%Slide:
As=Aslide_w_tank;
Bs=Bslide_w_tank;
C2=Cslide_w_tank;
D2=0;

%Tank:

At=Atank;
Bt=Btank;
C1=Ctank_level;
```

```
D1=Dtank;

nt=size(At,1);

ns=size(As,1);


A=[As zeros(ns,nt);Bt*C2 At];

B=[Bs;zeros(nt,1)];

Clevel=[D1*C2 C1];

Cslide=[C2 zeros(1,nt)];


%Discretized stuff


sys_slide_d=c2d(sys_slide,Ts,'zoh');

[Aslide_d,Bslide_d,Cslide_d,Dslide_d]=ssdata(sys_slide_d);

sys_tank_d=c2d(tanksys_no_z,Ts,'zoh');

[Atank_d,Btank_d,Ctank_d,Dtank_d]=ssdata(sys_tank_d);


save fin_model A B Clevel Cslide As Bs C2 D2 At Bt C1 D1  nt ns


save fin_modelR12 A B Clevel Cslide As Bs C2 D2 At Bt C1 D1 nt ns  -V6
```

1.4 Sliding Mode Control design

```
 %Updated on 02/21/08: Use finalized models

%Run after load_test_noaccel_reduced

C=Cslide;

 %Design conventional SMC regulator

 %Establishing the size of input distribution matrix
[nn,mm] = size(B);

%Perform QR decomposition on the input distribution matrix
[Tr temp] = qr(B);
Tr = Tr';
Tr = [Tr(mm+1:nn,:);Tr(1:mm,:)];
clear temp;
%Special measure to have B2=1
%Tr(4,2)=1/B(2);
%To obtain Areg,Breg
Areg = Tr*A*Tr';
Breg = Tr*B;

%Obtain matrix sub-blocks for sliding mode controller design
A11 = Areg(1:nn-mm,1:nn-mm);
A12 = Areg(1:nn-mm,nn-mm+1:nn);
A21 = Areg(nn-mm+1:nn,1:nn-mm);
A22 = Areg(nn-mm+1:nn,nn-mm+1:nn);
```

```
B2 = Breg(nn-mm+1:nn,1:mm);


%Sliding gain selection via LQ method (Utkin & Young, 1978)

Q=diag([1 1 1e4 1e4]);  %weighs states in original coordinates

%Transform:
Qt=Tr*Q*Tr';
Q11=Qt(1:nn-mm,1:nn-mm);
Q12=Qt(1:nn-mm,nn-mm+1:nn);
Q21=Qt(nn-mm+1:nn,1:nn-mm);
Q22=Qt(nn-mm+1:nn,nn-mm+1:nn);

Qhat=Q11-Q12*inv(Q22)*Q21;
Ahat=A11-A12*inv(Q22)*Q21;

[K,P1,E]=lqr(Ahat,A12,Qhat,Q22);

M=inv(Q22)*(A12'*P1+Q21);
S2 = -1/B(2);%/eye(mm);
S = S2*[M eye(mm)]*Tr;

%Equivalent Control Gain
G = -(inv(S*B))*S*A;

eta = 7;

%Design observer
```

```
%Q=eye(8);
Q=eye(4);
R=1;
L=lqr(A',C',Q,R);

D=0;

phi=0.01;

%Auxiliary matrices to obtain states from measurements
invmatrix=inv([Ctank_level;Ctank_level*Atank]);
gain_aux=Ctank_level*Btank;
initial_pos=400;
stack=[-Dtank*initial_pos;-gain_aux*initial_pos];
TP=[initial_pos;0;invmatrix*stack];
X0=0;

%save stuff for realtime simulation
save realtime_pars Aslide_d Atank_d Bslide_d Btank_d Cslide_d Ctank_d Ctank_level
     Dslide_d Dtank Dtank_d G Ts eta gain_aux invmatrix phi -V6

sim('slosh_digital_ctrl')
level_sens=-23.43;                              %in mm/volt
lw1=1.5;
lw2=1;
fsize=12;
subplot(4,1,1)
plot(T,Y,'k','Linewidth',lw1);
title('Simulation Results','Fontsize',fsize)
```

```
axis([0 2 0 420])
grid
ylabel('Pos., mm','Fontsize',fsize)
subplot(4,1,2)
plot(T,level*level_sens,'k','Linewidth',lw1)
axis([0 2 -30 30])
ylabel('Slosh in mm','Fontsize',fsize)
subplot(4,1,3)
plot(DT,U,'k','Linewidth',lw1)
axis([0 2 -15 15])
ylabel('Control, V','Fontsize',fsize)
subplot(4,1,4)
plot(DT,Sliding,'k','Linewidth',lw1)
axis([0 2 -12 12])
ylabel('S-Function','Fontsize',fsize)
xlabel('Time in seconds','Fontsize',fsize)
grid
```

1.5 Real Time plots

```
%realtime_plots


load prod_run %this run uses phi=0.01, Ts=1e-3;


t=prod_run.X.Data';


level_sensor_volts=prod_run.Y(1).Data';
level_model_volts=prod_run.Y(2).Data';
level_rate_sensor_volts=prod_run.Y(3).Data';
slide_pos_encoder=prod_run.Y(4).Data';
slide_pos_model=prod_run.Y(5).Data';
level_rate_model=prod_run.Y(6).Data';
s=prod_run.Y(7).Data';
tank_state1_meas=prod_run.Y(8).Data';
tank_state2_meas=prod_run.Y(9).Data';
tank_state1_model=prod_run.Y(10).Data';
tank_state2_model=prod_run.Y(11).Data';
control_signal=prod_run.Y(12).Data';
slide_vel_encoder=prod_run.Y(13).Data';
slide_vel_model=prod_run.Y(14).Data';


level_sens=-23.43;                                %in mm/volt
lw1=1.5;
lw2=1;
fsize=12;
subplot(4,1,1)
plot(t,slide_pos_encoder,'k','Linewidth',lw1);
```

```
title('Run with T_s=0.001 s','Fontsize',fsize)
hold on
plot(t,slide_pos_model,'k--','Linewidth',lw2)
axis([0 1.75 0 420])
grid
ylabel('Pos., mm','Fontsize',fsize)
legend('Measured','Model','Location','NW')
subplot(4,1,2)
plot(t,level_sensor_volts*level_sens,'k','Linewidth',lw1)
hold on
plot(t,level_model_volts*level_sens,'k--','Linewidth',lw2)
axis([0 1.75 -30 30])
ylabel('Slosh in mm','Fontsize',fsize)
legend('Measured','Model')
subplot(4,1,3)
plot(t,control_signal,'k','Linewidth',lw1)
axis([0 1.75 -15 15])
ylabel('Control, V','Fontsize',fsize)
subplot(4,1,4)
plot(t,s,'k','Linewidth',lw1)
axis([0 1.75 -12 12])
ylabel('S-Function','Fontsize',fsize)
xlabel('Time in seconds','Fontsize',fsize)
grid

load prod_run2 %this run uses phi=0.01, Ts=1e-4;

t=prod_run2.X.Data';
```

```
level_sensor_volts=prod_run2.Y(1).Data';
level_model_volts=prod_run2.Y(2).Data';
level_rate_sensor_volts=prod_run2.Y(3).Data';
slide_pos_encoder=prod_run2.Y(4).Data';
slide_pos_model=prod_run2.Y(5).Data';
level_rate_model=prod_run2.Y(6).Data';
s=prod_run2.Y(7).Data';
tank_state1_meas=prod_run2.Y(8).Data';
tank_state2_meas=prod_run2.Y(9).Data';
tank_state1_model=prod_run2.Y(10).Data';
tank_state2_model=prod_run2.Y(11).Data';
control_signal=prod_run2.Y(12).Data';
slide_vel_encoder=prod_run2.Y(14).Data';
slide_vel_model=prod_run2.Y(15).Data';


figure(2)
subplot(4,1,1)
plot(t,slide_pos_encoder,'k','Linewidth',lw1);
title('Run with T_s=0.0001 s','Fontsize',fsize)
hold on
plot(t,slide_pos_model,'k--','Linewidth',lw2)
axis([0 1.75 0 420])
grid
ylabel('Pos., mm','Fontsize',fsize)
legend('Measured','Model','Location','NW')
subplot(4,1,2)
plot(t,level_sensor_volts*level_sens,'k','Linewidth',lw1)
hold on
```

```
plot(t,level_model_volts*level_sens,'k--','Linewidth',lw2)
axis([0 1.75 -30 30])
ylabel('Slosh in mm','Fontsize',fsize)
legend('Measured','Model')
subplot(4,1,3)
plot(t,control_signal,'k','Linewidth',lw1)
axis([0 1.75 -15 15])
ylabel('Control, V','Fontsize',fsize)
subplot(4,1,4)
plot(t,s,'k','Linewidth',lw1)
axis([0 1.75 -12 12])
ylabel('S-Function','Fontsize',fsize)
xlabel('Time in seconds','Fontsize',fsize)
grid


load prod_run_final %this run uses phi=0.01, Ts=1e-3;
%latest corrections

t=prod_run_final.X.Data';

level_sensor_volts=prod_run_final.Y(1).Data';
level_model_volts=prod_run_final.Y(2).Data';
level_rate_sensor_volts=prod_run_final.Y(3).Data';
slide_pos_encoder=prod_run_final.Y(4).Data';
slide_pos_model=prod_run_final.Y(5).Data';
level_rate_model=prod_run_final.Y(6).Data';
s=prod_run_final.Y(7).Data';
tank_state1_meas=prod_run_final.Y(8).Data';
```

```
tank_state2_meas=prod_run_final.Y(9).Data';
tank_state1_model=prod_run_final.Y(10).Data';
tank_state2_model=prod_run_final.Y(11).Data';
control_signal=prod_run_final.Y(12).Data';
slide_vel_encoder=prod_run_final.Y(13).Data';
slide_vel_model=prod_run_final.Y(14).Data';



figure(3)
subplot(4,1,1)
plot(t,slide_pos_encoder,'k','Linewidth',lw1);
title('Run with T_s=0.001 s','Fontsize',fsize)
hold on
plot(t,slide_pos_model,'k--','Linewidth',lw2)
axis([0 1.75 0 420])
grid
ylabel('Pos., mm','Fontsize',fsize)
legend('Measured','Model','Location','NW')
subplot(4,1,2)
plot(t,level_sensor_volts*level_sens,'k','Linewidth',lw1)
hold on
plot(t,level_model_volts*level_sens,'k--','Linewidth',lw2)
axis([0 1.75 -30 30])
ylabel('Slosh in mm','Fontsize',fsize)
legend('Measured','Model')
subplot(4,1,3)
plot(t,control_signal,'k','Linewidth',lw1)
axis([0 1.75 -15 15])
ylabel('Control, V','Fontsize',fsize)
```

```
subplot(4,1,4)
plot(t,s,'k','Linewidth',lw1)
axis([0 1.75 -12 12])
ylabel('S-Function','Fontsize',fsize)
xlabel('Time in seconds','Fontsize',fsize)
grid
```

1.6 Simulink model for Digital SMC

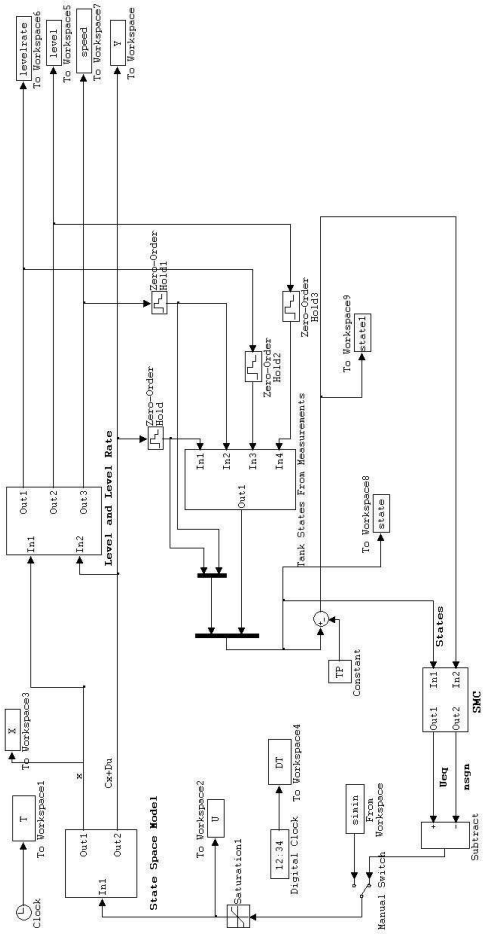
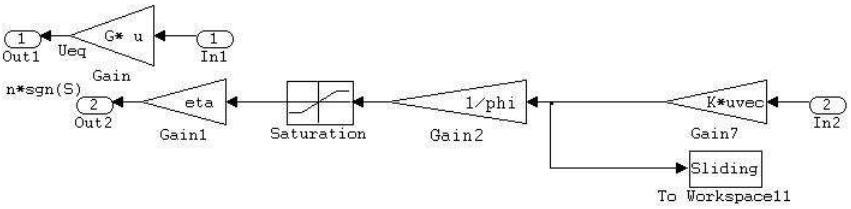
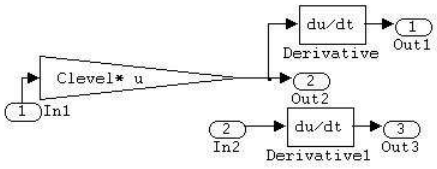


Figure 1.1: Simulink model for digital SMC

1.7 Simulink model for masked subsystems



SMC Subsystem Block



Level and Level rate Subsystem Block

Figure 1.2: Simulink model for masked subsystem