Sriniwas Chowdhary Maddukuri

Modelling and Control of a Tank Mixing System

Masterarbeit

zur Erlangung des akademischen Grades
Diplom–Ingenieur

Studium Information Technology

Alpen-Adria-Universität Klagenfurt
Fakultät für Technische Wissenschaften

Begutachter: Univ.–Prof. Dr. techn. Dipl.-Ing. Martin Horn
Betreuer: Dipl.-Ing. Dr. Felix Staudegger

Institut für Intelligente Systemtechnologien
Messe und Regelungstechnik

Klagenfurt, im Mai 2012
Eidesstattliche Erklärung

Ich erkläre ehrenwörtlich, dass ich die vorliegende wissenschaftliche Arbeit selbstständig angefertigt und die mit ihr unmittelbar verbundenen Tätigkeiten selbst erbracht habe. Ich erkläre weiters, dass ich keine anderen als die angegebenen Hilfsmittel benutzt habe. Alle aus gedruckten, ungedruckten oder dem Internet im Wortlaut oder im wesentlichen Inhalt übernommenen Formulierungen und Konzepte sind gemäß den Regeln für wissenschaftliche Arbeiten zitiert und durch Fußnoten bzw. durch andere genaue Quellenangaben gekennzeichnet.

Die während des Arbeitsvorganges gewährte Unterstützung einschließlich signifikanter Betreuungshinweise ist vollständig angegeben.


Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

(Unterschrift)  (Ort, Datum)
Abstract

Mathematical modelling and control of multi-input multi-output (MIMO) systems presents quite a few challenges. The main challenge behind the modelling of MIMO systems is to identify the cross interaction between its outputs. Due to this cross interaction, control of MIMO systems becomes a significant challenge to overcome. The basic control design strategy realized for MIMO systems is proportional-integral (PI) control. PI control of MIMO systems is mainly performed by decoupling the actual MIMO system into a set of single-input single-output (SISO) subsystems. A fluid mixing system exhibiting MIMO system characteristics is dealt with in this thesis study.

Fluid mixing systems often have dead zone behavior. A dead zone is a common nonlinearity in which the system responds to the input only after the input attains a certain level. This particular nonlinearity in the fluid mixing system is identified with real-time experiments. The realization of fluid mixing system as a nonlinear MIMO system is presented. A model based controller design approach is used for the fluid mixing system to have the desired flow control.

Mathematical modelling of the fluid mixing system is described on the basis of an empirical modelling technique. A sequential way of identifying the model parameters is demonstrated through least squares and optimization methods. Pole-placement method is adopted for tuning the PI control parameters. The robustness of the computed PI parameters is illustrated through simulation and real-time experiments on fluid mixing system.
Acknowledgments

I am grateful to Alpen-Adria University of Klagenfurt under the Institute of Smart System Technologies who entitled me to pursue the master study program in Information Technology. I am thankful to the Department of Control and Measurement Systems for providing a peaceful study environment.

I express my gratitude to Univ.-Prof. Dr. techn. Dipl.-Ing. Martin Horn for his consistent, invaluable guidance and supervision regarding this master thesis work. I am grateful to you for assigning fundamental research topics and its critical examination of the technical work which in turn gave me the confidence to do the master thesis work. Through your organization, I was fortunate enough to attend the sliding mode control workshop which was actually the origin to the master thesis work.

I am extremely thankful to have you Dipl.-Ing. Dr. Felix Staudegger, I appreciate your advice and supervision during the master thesis progress, your endless support and patience make this to be achieved. Special regards to Test Engineer of Component Test Team in LAM Research AG Mr. Helmut Loy and his colleagues for their constant technical support. I am thankful to the staff of Spin Clean Division, LAM Research AG for sharing their industrial experience and the technical challenges which they had to overcome in the end.

I heartily thank Univ.-Prof. Dr. techn. Dipl.-Ing. Martin Horn and Dipl.-Ing. Dr. Felix Staudegger for providing the financial support which gave me additional motivation to concentrate on master thesis work.

I must express my appreciation to Dipl.-Ing. Dr. Markus Reichhartinger and Dipl.-Ing. Dr. Jakob Rehrl who were always ready to give their invaluable suggestions and providing the constructive feedback throughout the masterthesis study. I should thank Prof. Dr. Vadim Utkin for being generous enough to share his instinctive ideas with me during the early stages of thesis study.

I kindly appreciate my parents and relatives for being the driving force in my academic career and allowing me the freedom to put all my energy on the preferred educational track.

I am pleased to thank my colleagues for their love and encouragement during the development and progressing of this master thesis work. I thank anyone who in one way or another shared a moment with me during this study, I hope the best for all of you to achieve their respective long term goals in time.
# List of Figures

2.1 Leviflow TM flowmeter ......................................................... 3  
2.2 BPS 200 Pump System .......................................................... 4  
2.3 Pneumatic manual shutoff valve SAV series ............................. 5  
2.4 MML schematic diagram ....................................................... 6  
3.1 Single mass spring damper system .......................................... 8  
3.2 Measured step response ....................................................... 9  
3.3 Measured step response and first order model ......................... 10  
3.4 Measured step response with time delay .................................. 11  
3.5 Measured step response and first order model with time delay .... 11  
3.6 Measured frequency response .............................................. 12  
3.7 Measured frequency response and first order model frequency response 13  
3.8 MS flow with respect to experiment I .................................... 14  
3.9 PS flow with respect to experiment I .................................... 15  
3.10 Experiment II ................................................................. 16  
4.1 Continuous time system with applied discrete input and measured discrete output .................................................. 20  
4.2 Continuous time system with least squares algorithm ............... 21  
4.3 Least squares principle ..................................................... 21  
4.4 Linear fit of PS data points(Identification of $\alpha_1$ and $\beta_1$) ........ 28  
4.5 MS flow with respect to constant pressure differential $P$ ............ 29  
4.6 Measured v/s simulation of MS and PS fluid flows with respect to scenario 1 .................................................. 33  
4.7 Measured v/s simulation of MS and PS fluid flows with respect to scenario 2 .................................................. 34  
4.8 Measured v/s simulation of MS and PS fluid flows with respect to scenario 3 .................................................. 35  
5.1 Plant with feed forward control ............................................ 37  
5.2 Plant with feedback control .................................................. 37  
5.3 Unity feedback loop ......................................................... 38  
5.4 Graphical solution I-Damping factor $\zeta$ v/s overshoot $M_p$ .......... 41  
5.5 Graphical solution II-Damping factor($\zeta$) v/s product of natural frequency($\omega_n, t_r$) ........................................... 41  
5.6 Blending test with 2:1 mixing ratio in scenario 1 ....................... 46  
5.7 Blending test with 10:1 mixing ratio in scenario 1 ...................... 47  
5.8 Blending test with 20:1 mixing ratio in scenario 1 ...................... 48  
5.9 Blending test with 20:1 mixing ratio in scenario 2 ...................... 50  
5.10 Blending test with 80:1 mixing ratio in scenario 2 ...................... 51
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Estimates of coefficients $\alpha_i$ and $\beta_i$ with respect to three scenarios</td>
<td>28</td>
</tr>
<tr>
<td>4.2</td>
<td>Estimates of parameters $k_0$ and $k_1$</td>
<td>29</td>
</tr>
<tr>
<td>4.3</td>
<td>Estimates of MS and PS parameters with respect to scenario 1</td>
<td>31</td>
</tr>
<tr>
<td>4.4</td>
<td>Estimates of MS and PS parameters with respect to scenario 2</td>
<td>32</td>
</tr>
<tr>
<td>4.5</td>
<td>Estimates of MS and PS parameters with respect to scenario 3</td>
<td>32</td>
</tr>
<tr>
<td>5.1</td>
<td>PI parameters of MML system under first scenario</td>
<td>44</td>
</tr>
<tr>
<td>5.2</td>
<td>PI parameters of MML system under second scenario</td>
<td>44</td>
</tr>
<tr>
<td>5.3</td>
<td>PI parameters of MML system under third scenario</td>
<td>45</td>
</tr>
<tr>
<td>5.4</td>
<td>PI parameter settings of MML system</td>
<td>45</td>
</tr>
<tr>
<td>5.5</td>
<td>Mixing ratios of MML system in different scenarios</td>
<td>45</td>
</tr>
<tr>
<td>5.6</td>
<td>PI parameters of the decoupled MML system under first scenario</td>
<td>61</td>
</tr>
<tr>
<td>5.7</td>
<td>PI parameters of decoupled MML system under second scenario</td>
<td>61</td>
</tr>
<tr>
<td>5.8</td>
<td>PI parameters of decoupled MML system under third scenario</td>
<td>61</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Motivation

Fluid mixing systems are often found in chemical and process engineering. It incorporates the mixing of two identical or distinct fluids to make it available for etching. Etching is one of the processes involved in integrated circuit design which deals with the chemical removal of layers from the surface of a silicon wafer. Fluid mixing system provides the required fluid mixtures of precise concentration level to carry out the etching process. To have the fluid mixtures of desired concentration level, blending of identical or distinct fluids has to be performed in an appropriate manner. Therefore, blending of the fluid mediums can be achieved through flow control. Hence, a control design strategy was required for the fluid mixing system to ensure consistent blending of fluid mediums with respect to user-defined specifications.

1.2 State of the Art

Fluid mixing system can be regarded as a typical multi-input multi-output (MIMO) system. MIMO systems often have substantial coupling between each of its process outputs and sometimes between the process inputs and process outputs as well. Real world plants with such kind of relations between its inputs and outputs are mentioned in [24, 18, 11, 2, 16, 21]. In the case of fluid mixing system, multi-variable controllers are designed to meet fluid delivery requirements. The control design strategy preferred for the realization of multi-variable controllers is the proportional integral (PI) control. In earlier works [2, 21], PI control of MIMO systems was carried out by decoupling the strong cross interaction between the process outputs. Through decoupling, the actual MIMO system is transformed into set of independent single-input single-output (SISO) subprocesses. Even if there is some kind of abrupt change in one process output, this would not have a negative impact on the other process output when the decoupled MIMO system is operated with PI controllers. In some cases, the PI parameters of decoupled MIMO system were computed using pole-placement design and dominant pole design methods [3]. In other cases, the PI parameters of decoupled MIMO system were tuned on the basis of bode plot approach from the closed loop time-domain specifications [18].
1.3 Goals

Tuning of PI control parameters can be time consuming without realizing a mathematical model for the actual system. The mathematical model has to be identified in such a way that it is capable enough of simulating the dynamics exhibited by the real world plant. The goal was to come up with a systematic approach for computing the PI control parameters of fluid mixing system. The systematic approach is developed under the following routines:

- Mathematical modelling
- Parameter estimation
- Model based controller design

PI control design of the fluid mixing system has to be realized on the basis of a theoretical approach which gives more freedom to tune its parameters. The resultant PI parameters has to be robust enough to guarantee the desired operation of fluid mixing system.

1.4 Outline of the Thesis

Chapter 2 (Mix Module Levitronix system) gives a detailed description about Mix Module Levitronix system based on its major hardware components. Later, the basic operation of the Mix Module Levitronix system is covered extensively with the help of its schematic diagram.

Chapter 3 (Mathematical modelling) presents the concept of mathematical modelling. It specifies two distinct methods for computing the mathematical model of a dynamical system. Mathematical modelling of the fluid mixing system will be explained in detail based on one of the modelling methods.

Chapter 4 (Parameter estimation) deals with the identification of the parameters involved with the formulated mathematical model of the fluid mixing system under different scenarios. It shows how least squares method is applied to compute the initial estimates for the unidentified parameters. Later, it depicts how the initial estimates are practiced with an optimization method to yield optimum values for the unidentified parameters.

Chapter 5 (Model based controller design) deals with the PI control design of the fluid mixing system. It describes the computation of PI control parameters based on the pole placement for unity feedback loop technique. It also demonstrates about the PI control design with decoupling of fluid mixing system in an elaborated manner.

Chapter 6 (Conclusion) concludes the thesis work and specifies interesting details about the fluid mixing system which can be extended in future.
Chapter 2

Mix Module Levitronix System

Mix Module Levitronix (MML) system is defined as the fluid mixing system dedicated for obtaining homogeneous or heterogeneous fluid mixtures of relatively high or low concentration. Fluid mixing systems are applied in the process of etching. The main hardware components of MML system consist of flowmeters, centrifugal pumps and pneumatic valves.

2.1 Flowmeters

Flowmeter is defined as the sensor device used to measure the flow of a fluid. Leviflow TM flowmeters are employed for measuring the fluid flows of MML system. Leviflow TM flowmeter is shown in Figure 2.1.

![Figure 2.1: Leviflow TM flowmeter](http://www.levitronix.com/Products_Brochures_and_Manuals/Brochure_LEVIFLOW_english_Rev04.html)

Figure 2.1 consists of two piezoelectric transducers attached at both ends of the measuring fluid path. These two transducers generate and receive an ultrasonic wave. The wave going in the direction of flow (with the stream wave) is accelerated and the wave going against (against the stream wave) the flow direction is slowed down. The two waves are processed by a signal converter. The difference of the transit-time of both the waves is proportional to the velocity of the fluid. The calibration and required operation of leviflow TM flowmeters is carried out with the leviflow TM flowmeters configuration software.
2.2 Centrifugal Pumps

Centrifugal pump is a variable-displacement pump in which the amount of fluid displaced per one revolution of the shaft can be varied while the pump is operating. Centrifugal pumps are commonly used to move fluids through a piping system. In the case of MML system, BPS-200 pump systems are installed for the discharge of fluid volume at each fluid path. BPS-200 pump system is shown in Figure 2.2.

Figure 2.2: BPS 200 Pump System

Figure 2.2 consists of a maglev centrifugal pump motor with pump head and impeller. Impeller is a rotating component which takes the incoming fluid and induces the fluid outwards from the center of rotation. Head is a fluid mechanics term used to measure the height of the fluid column formed from the fluid’s kinetic energy created by the pump. The fluid enters the impeller along or near to the rotating axis and is accelerated by the impeller, flowing outward into a diffuser or volute chamber (casing).

2.3 Pneumatic Valves

Pneumatic valve is a mechanical device that directs the flow of a fluid by the use of compressed air. In the case of MML system, SAV series pneumatic manual/shut-off valves are used for directing the fluid to its desired path. SAV series pneumatic manual/shut off valve is shown in Figure 2.3.

From the Figure 2.3 compressed air is supplied through operational air ports. Initially, with no supply air, the spring forces the diaphragm upward against the casing and holds the valve fully open. As supply air pressure is increased from zero, the force on the top of the diaphragm overcomes the opposing force of the spring. This causes the diaphragm to move downward and the valve to close. With increasing supply air pressure, the diaphragm will continue to move downward and expand the spring until the valve is closed. Conversely, if supply air pressure is decreased, the spring will begin to force the diaphragm upward.
and open the valve. An air compressor in combination with a solenoid valve delivers the required compressed air to the pneumatic valves in order to perform its normal operation.

2.4 Operation of MML System

The real world plant is realized with all the major components in Figure 2.4. MML system is provided with two fluid streams namely mainstream and partstream. Mainstream (MS) is defined as the fluid path in which the fluid assigned to it is used as a solvent. Partstream (PS) is defined as the fluid path in which the fluid assigned to it is used as a solute. Distilled water is used as the fluid medium for both MS and PS. The fluid mixtures which are obtained from the MML system are of homogenous nature.

The flowmeters and centrifugal pumps (rotational actuators) are placed on their respective fluid paths. Also, there is a main storage tank located at the bottom left corner of Figure 2.4 and a PS tank located just before the PS pump which are attributed for the storage of fluids with respect to MS and PS. Three contactless sensors are fixed on one side of the PS tank. These sensors would detect the fluid level inside the PS tank.

An orifice of variable size is located on the PS. Its purpose is to restrict the volume of PS fluid which is about to mix with the MS fluid. In order to relieve overpressure inside the MML system, a pressure relief valve denoted by ‘FCRM’ is located on the MS. A pressure measurement sensor is placed just before the MS flowmeter for setting the prescribed pressure limit at the MS injection point. In order to provide constant supply of
fluid to MS, an additional pump I is located inside the main storage tank which is operated at a constant pressure differential of 1.3 bar. Additional pump II is located at a certain height from the main storage tank which is operated at a constant pressure differential of 5 bar. Additional pump II operates whenever there is a certain reduction of fluid level inside the PS tank.

Depending upon the speed of both the pumps and the switching of pneumatic valves 'VMSC' and 'VPS1', the MS and PS fluids are delivered. During the manual operation and control of MML system, the fluid inside the PS tank gets emptied. Replenishing of PS tank is carried out by the additional pump II and pneumatic valve 'VPS1'. If the fluid level inside the PS tank falls, one of the contactless sensors would detect the decrease of fluid level and switching of VPS1 takes place. In turn the additional pump II would displace the fluid to PS tank. There are three scenarios in which the PS fluid flow would vary drastically with respect to the orifice sizes 3mm, 1mm and 0.45mm. To create and assign various mixing ratios with respect to the change in orifice size on PS and also to carry out the blending tests on the real world plant, a graphical user interface is created for the MML system using MML system control software. The control of MML system in real-time is exercised with the WAGO 750-842 ethernet programmable fieldbus controller which is interfaced with a personal computer (PC). The programmable fieldbus controller for ethernet combines the WAGO 750-337 fieldbus coupler with the functionality of a programmable logic controller.
Chapter 3

Mathematical Modelling

3.1 Introduction

Mathematical modelling plays an important role in today’s science and engineering. Modelling is an essential way to carry out many tasks such as simulation, control design and signal processing. Mathematical modelling is defined as a theoretical method to represent the behavior of a dynamical system in the form of e.g., ordinary differential equations. The objective behind the mathematical modelling was to have a clear and precise understanding about the system dynamics. There are two ways to perform the mathematical modelling of a dynamical system:

- Physical modelling
- Empirical modelling

3.2 Physical Modelling

Physical modelling is one approach where the behavior of a dynamical system is known by its physics and mathematics. Based on various physical laws, the system dynamics can be modelled. A classic example to describe the physical modelling is the spring mass damper system [17]. A typical spring mass damper system is shown in Figure 3.1.

Figure 3.1 [17, 10] consists of a body of mass $m$. The mass $m$ is attached to an immovable object with the help of a spring and a damper. Spring has the characteristic of opposing the force exerted by the motion of mass particle. Damper is placed for reducing the oscillations produced by the mass particle. Mathematical modelling of the spring mass damper system is performed on the following assumptions:

- Assuming mass particle is placed on a frictionless surface
- Spring and damper never loses their contact with the body during its motion
- Assuming an external force is applied to the mass particle

According to Newton’s second law of motion [22], the sum of forces acting on a mass particle equals the mass times its acceleration. The forces which act on the mass particle
Figure 3.1: Single mass spring damper system

are the spring force, force exerted by the damper and external force applied to the mass particle. Equation of motion which govern the mass particle is specified from the following equation,

\[ m \ddot{x} = F + F_k + F_d \] (3.1)

- \( x \) is the position of the mass particle
- \( F \) is the applied external force
- \( F_k \) is the spring force
- \( F_d \) is the damper force

According to Hooke’s law, spring force \( F_k \) is given by the following equation:

\[ F_k = -k.x \] (3.2)

- \( k \) is spring constant

Damping force is modelled as being proportional to the velocity of the mass particle. Therefore, damping force \( F_d \) is given by the Equation 3.3

\[ F_d = -d.\dot{x} \] (3.3)

- \( d \) is damping coefficient
Substituting Equation 3.3 and Equation 3.2 into Equation 3.1, we get,

\[ m.\ddot{x} = -k.x - d.\dot{x} + F \]  

(3.4)

Mathematical model of a spring mass damper system using first principles modelling can be reviewed from Equation 3.4.

### 3.3 Empirical Modelling

Empirical modelling is another approach where the behavior of a dynamical system is identified from its measured data. In other words, the mathematical model is constructed without any prior knowledge about the system. Empirical modelling is accomplished by making the best use of the measured data of a dynamical system. There are several methods to perform this particular modelling, two of the most widely adopted methods are specified as follows:

- Step response based modelling
- Frequency response based modelling

#### Step Response Based Modelling

Step response is defined as the behavior of output whenever the input state changes from zero to one in a very short time interval. In practice, system dynamics with respect to sudden changes in the input are investigated by acquiring its step response \[ \text{[11, 6, 7]} \]. Step response data is easier to obtain and they give apparent knowledge about the system dynamics. While acquiring the step response, low level of disturbance is acceptable. A typical step response is shown in Figure 3.2. Measured step response is acquired with high frequent noise.

![Measured Step Response](image-url)  

Figure 3.2: Measured step response
CHAPTER 3. MATHEMATICAL MODELLING

It seems from Figure 3.2 that a first order model might fit the data well. A first order model is represented by the following transfer function,

\[ G_1(s) = \frac{Y(s)}{U(s)} = \frac{K}{1 + s.\tau} \]  (3.5)

- \( G_1(s) \) is transfer function, \( s \) is the Laplacian operator
- \( K \) is steady state gain
- \( \tau \) is time constant of the system response

In Equation 3.5, \( Y(s) \) and \( U(s) \) are the Laplace transforms of output and input signals \( y(t) \) and \( u(t) \) respectively. Now the input is applied with step magnitude of \( a \). The output response with respect to the applied step magnitude \( a \) is given by Equation 3.6,

\[ U(s) = \frac{a}{s} \Rightarrow Y(s) = \frac{K.a}{s.(1 + s.\tau)} \]  (3.6)

\( K \) and \( \tau \) can be determined graphically using the following two equations,

\[ t \to \infty : K = \frac{y(\infty)}{a} \]  (3.7)

\[ t = \tau : y(\tau) = K.a.(1 - e^{-1}) = 0.63.K.a \]  (3.8)

Steady state gain \( K \) is now determined as the steady state value of the model output \( y(\infty) \) divided by the step magnitude \( a \). The time constant \( \tau \) is the amount of time it takes to achieve 63\% of the steady state value of the model output \( y(\infty) \) [6, 7]. Therefore, a smooth exponential curve can be drawn through the measured step response. Figure 3.3 shows the match between the first order model with the measured step response.

Figure 3.3: Measured step response and first order model
CHAPTER 3. MATHEMATICAL MODELLING

In reality, transport delays occur in many processes. System with one large time constant and a number of small time constants could be approximated by the large time constant plus a time delay equal to the sum of the small time constants. Typical measured step response with a time delay is shown in Figure 3.4. From the Figure 3.4 it seems that a first order model with time delay can be a proper choice to fit the data well. First order model with time delay $T$ is given by the following transfer function $[9, 12]$, 

$$G_2(s) = \frac{K}{1 + s\tau}e^{sT}$$

(3.9)

For the above model the three parameters, time delay $T$, time constant $\tau$ and steady state gain $K$ can be determined graphically using Figure 3.5.

Figure 3.4: Measured step response with time delay

Figure 3.5: Measured step response and first order model with time delay
Frequency Response Based Modelling

Frequency response is defined as the behavior of output built over a wide range of frequencies. In practice, system dynamics with respect to an applied test signal (e.g., sinusoidal wave) of different frequencies are studied by acquiring the frequency response. Frequency response data is time consuming to acquire than step response data. Particularly, this is true if the determined time constants of the process are high. But the model computed is generally more accurate because of better noise filtering ability of the frequency response method. A typical measured frequency response data is shown in Figure 3.6.

![Figure 3.6: Measured frequency response](image)

From the Figure 3.6 it appears that there seems to be a time delay occurrence in the process. A first order model with time delay would be the best to fit the measured frequency response. Modelling from frequency response data starts from drawing asymptotes on the magnitude plot with slopes 0 and -20dB/decade [6, 7]. Once the asymptotes are sketched properly on the magnitude plot, time constant $\tau$ can be determined by the inverse of the frequency at which the distance between the asymptote with slope 0dB/decade and magnitude curve is more. Steady state gain $K$ is determined from the amplitude at which the distance between the asymptote with slope 0dB/decade and magnitude curve is less. According to the Figure 3.6, $\omega_1$ is the frequency at which distance between the asymptote with slope 0dB/decade and magnitude curve is high and $A$ is the amplitude at which distance between the asymptote with slope 0dB/decade and magnitude curve is less. Therefore, steady state gain $K$ and time constant $\tau$ are computed from Equation 3.10 and Equation 3.11.

$$\tau = \frac{1}{\omega_1} \quad (3.10)$$

$$20 \log_{10} K = A \quad (3.11)$$
After determining $K$ and $\tau$, phase response of the first order model without the computed delay is plotted. Now the phase plots occurred from the measured frequency response and the first order model are simultaneously analyzed using Figure 3.7.

Figure 3.7: Measured frequency response and first order model frequency response

From the Figure 3.7 phase characteristics of the first order model are studied [6, 7]. The phase characteristics of the first order model with respect to certain range of frequencies are specified in the following way:

\[
\omega \leq \frac{\omega_1}{10} \Rightarrow \phi \cong 0^\circ \tag{3.12}
\]

\[
\frac{\omega_1}{10} < \omega < 10.\omega_1 \Rightarrow \phi = -45^\circ \cdot (1 + \log_{10}(\frac{\omega}{\omega_1})) \tag{3.13}
\]

\[
\omega \geq 10.\omega_1 \Rightarrow \phi \cong -90^\circ \tag{3.14}
\]

The extra phase $\Delta\phi$, due to the time delay $T$ at frequency $\omega_2$ is then determined as the phase deviation between the phase of the first order model and phase of the measured frequency response at $\omega_2$. The corresponding delay $T$ is calculated in the following way,

\[
-\Delta\phi = \angle e^{-j\omega_2 T} = -\frac{180^\circ}{\pi} \omega_2 T \tag{3.15}
\]

Time delay $T$ is given by the following equation,

\[
T = \frac{\pi}{180^\circ} \cdot \frac{\Delta\phi}{T} \tag{3.16}
\]
CHAPTER 3. MATHEMATICAL MODELLING

3.4 MML System Modelling

Mathematical modelling of MML system deals precisely with fluid dynamics. Fluid dynamics is modelled based on the law of conservation of energy and law of conservation of momentum. Using these two physical laws, the dynamics of fluid flow are represented in the form of partial differential equations. Partial differential equations describe the fluid flow phenomenon in a multi-dimensional manner [1]. In general, partial differential equations are much more difficult to solve analytically than are ordinary differential equations. In terms of modelling the MML system, fluid dynamics with respect to a single dimensional quantity was preferred. Therefore, the mathematical modelling of MML system was carried out based on fluid flow characteristics with respect to time. Hence, empirical modelling approach was an appropriate way to identify the fluid dynamics based on the measurement data acquired from the MML system. At first, the inputs and outputs have to be known within the perspective of MML system. In the case of MML system, inputs are the two pump speeds and outputs are the two fluid flows measured from the flowmeters located on each of the fluid paths.

3.4.1 Experimental Setup

Two kinds of test experiments were performed on MML system. The motivation behind to carry out the test experiments on the MML system was to look at the behavior of the outputs whenever a single input or multiple inputs are applied to it. In the first experiment, a constant speed is applied to one of the centrifugal pumps and its corresponding fluid flow is measured by deactivating the other fluid path. For instance, a constant MS pump speed is applied and its respective MS fluid flow is measured by making the PS fluid path inactive and vice-versa. In the case of first experiment, the MS and PS fluid paths are regarded as two separate subsystems of MML system. The MS fluid flow resulted from the first experiment is shown in Figure 3.8. From the Figure 3.8, the MS fluid flow reaches a steady state value with respect to its applied constant pump speed.

![Figure 3.8: MS flow with respect to experiment I](image)
The curve shown in Figure 3.8 looks similar to the step response of a first order system without time-delay. A first order system without time-delay is given by the following transfer function:

$$G(s) = \frac{X(s)}{U(s)} = \frac{K}{1 + s\tau} \quad (3.17)$$

According to the Figure 3.8, MS pump speed and the obtained MS fluid flow are taken into consideration as input and output respectively. Therefore, the MS fluid flow dynamics represented by Figure 3.8 was modelled on the basis of the following first order differential equation.

$$\dot{Q}_{ms} = -k_1.Q_{ms} + k_2.W_{ms} \quad (3.18)$$

- $W_{ms}$ is MS pump speed measured in $r/min$
- $Q_{ms}$ is MS fluid flow measured in $cm^3/min$

![Partstream Flow](image1)

![Partstream Pump Speed](image2)

Figure 3.9: PS flow with respect to experiment I

Now the MS fluid path is made static and the same experiment is repeated for the PS fluid path i.e., a constant PS pump speed is applied and the corresponding PS fluid flow is measured. The PS fluid flow resulted from the first experiment is shown in Figure 3.9.

From the Figure 3.9, the PS fluid flow reaches a steady state value with respect to its applied constant pump speed. Here, the applied PS pump speed and the measured PS fluid flow are taken into consideration as input and output respectively. Similarly, the PS fluid flow dynamics shown in Figure 3.9 was modelled on the basis of Equation 3.18.

$$\dot{Q}_{ps} = -k_4.Q_{ps} + k_5.W_{ps} \quad (3.19)$$

- $W_{ps}$ is PS pump speed measured in $r/min$
- $Q_{ps}$ is PS fluid flow measured in $cm^3/min$
The second experiment was to obtain the complete dynamics of the fluid mixing system by applying one of the centrifugal pumps with constant speed and the other one with varying speed in magnitude. The MS and PS fluid flows resulted from the second experiment are shown in Figure 3.10.

Figure 3.10: Experiment II
Depending upon the behavior of MS and PS fluid flows which resulted in Figure 3.10, the following data driven observations were deduced:

- Observation I: MS fluid flow appears even before the MS pump speed is operated
- Observation II: MS fluid flow starts to decrease as soon as PS fluid flow starts to increase
- Observation III: At a certain applied PS pump speed, the PS fluid flow exists and thereupon it starts to increase

**Observation I**

The reason behind the existence of MS fluid flow at zero MS pump speed is due to the switching action of the pneumatic valve located on the fluid path of MS. According to the MML schematic diagram shown in Figure 2.4, the MS fluid flow primarily exists due to the constant pressure differential (CPD) maintained across the additional pump I. This particular dependency over MS fluid flow is introduced by extending the first order differential Equation 3.18 in the following manner:

\[
\dot{Q}_{ms} = -k_1 Q_{ms} + k_2 W_{ms} + k_0 P
\]

- \(k_0\) is the CPD parameter

**Observation II**

The decrease of MS fluid flow with the increase of PS fluid flow is due to the blending of both the fluids at the mixing point. PS fluid can be regarded as a kind of disturbing element to MS fluid. The higher the PS fluid flow, greater the decrease of MS fluid flow. Therefore, the MS fluid flow is dependent upon PS fluid flow as well. Equation 3.20 is elongated with this peculiar dependency in the following way,

\[
\dot{Q}_{ms} = -k_1 Q_{ms} + k_2 W_{ms} + k_0 P - k_3 Q_{ps}
\]

- \(k_1, k_2\) and \(k_3\) are MS fluid flow parameters

**Observation III**

According to second experiment, existence of PS fluid flow is due to the suppression of high MS pressure at the mixing point. With respect to fluid dynamics theory, a fluid medium moves from high pressure to low pressure. To lower the MS pressure at the mixing point, the PS pump needs to be operated at a considerable high speed. If the PS pump happens to be operated at high speed, the velocity of the PS fluid particles will increase further. Due to which the PS fluid begins to have an apparent motion. During experiment II, MS pressure at the mixing point was immeasurable. Therefore, acquired MS fluid flow is taken into consideration in order to complement this MS pressure effect. Hence, the behavior of
PS fluid flow was modelled on the basis of a nonlinear function comprising of PS pump speed and MS fluid flow which is evident from the following equation,

\[ \dot{Q}_{ps} = -k_4 Q_{ps} + f_{ps}(W_{ps}, Q_{ms}) \]  
\[ (3.22) \]

- \( k_4 \) is PS fluid flow parameter
- \( f_{ps}(W_{ps}, Q_{ms}) \) is a nonlinear function composed of \( W_{ps} \) and \( Q_{ms} \)

The nonlinear function depicted in Equation 3.22 was approximated by certain conditional expressions given in following way:

\[ f_{ps}(W_{ps}, Q_{ms}) = \begin{cases} k_5 W_{ps} - k_6 Q_{ms} & \text{if } W_{ps} + \alpha Q_{ms} \geq \beta \\ 0 & \text{Otherwise} \end{cases} \]  
\[ (3.23) \]

- \( k_5 \) and \( k_6 \) are additional PS fluid flow parameters
- \( \alpha \) and \( \beta \) are numerical constants

Using Equation 3.21, Equation 3.22 and Equation 3.23, a mathematical model was formulated which comprises the complete dynamics of MML system based on the measured data obtained from the second experiment and its data driven observations.

\[ \dot{Q}_{ms_i} = -k_1 Q_{ms_i} + k_2 W_{ms} + k_0 P - k_{3i} Q_{ps_i} \]  
\[ (3.24) \]

\[ \dot{Q}_{ps_i} = -k_{4i} Q_{ps_i} + f_{ps_i}(W_{ps}, Q_{ms_i}) \]  
\[ (3.25) \]

- \( f_{ps_i}(W_{ps}, Q_{ms_i}) = \begin{cases} k_{5i} W_{ps} - k_{6i} Q_{ms_i} & \text{if } W_{ps} + \alpha_i Q_{ms_i} \geq \beta_i \\ 0 & \text{Otherwise} \end{cases} \)

- \( i = 1, 2, 3 \) represents the MS and PS fluid flow scenarios with respect to the change of PS orifice size

The mathematical model of MML system resulted from Equation 3.24 and Equation 3.25 are an extension to the first order differential Equation 3.18 and Equation 3.19 which were earlier derived on the basis of first experiment. The derived mathematical model of MML system consists of MS and PS flow dynamics represented from both the real-time experiments. The mathematical model of MML system is a nonlinear multi-input multi-output (MIMO) system. The nonlinear nature of MML system is precisely involved with the PS flow characteristics. Although the MML system is nonlinear, it turns into a linear one when the PS pump speed enters into the operating range.
Chapter 4

Parameter Estimation

4.1 Introduction

Parameter estimation is defined as the technique to identify the unknown parameters involved with the mathematical model of a dynamical system in a numerical manner. Parameter estimation is a kind of requisite which needs to be achieved whenever an unidentified mathematical model is realized for a particular dynamical system. The objective behind the parameter estimation was to have a close match between the real world plant dynamics and the model carried out in simulation (i.e., mathematical model). There are several methods to perform parameter estimation for an undetermined mathematical model. Estimation of unknown parameters of the mathematical model of MML system was carried out using the following:

- Least squares method
- Simulink design optimization

4.2 Least Squares Method

Least squares method depicts a standard approach to identify the unknown parameters in a system of equations. The objective consists of setting the parameters of a model function to best fit a data set. For instance, there are N pairs of observations \((X_i, Y_i)\) which consist of an independent variable \(X\) and a dependent variable \(Y\). A straight line needs to be constructed using the variables \(X\) and \(Y\). Thus, a least squares problem is formulated with one variable and a linear function.

\[
\hat{Y} = a + b.X
\] (4.1)

Equation (4.1) involves two free parameters which specify the intercept \(a\) and slope \(b\) of the straight line. The least squares method defines the estimate of these parameters as the values which minimize the sum of squares between the measurements and the model (i.e., the predicted values) [5]. This amounts to minimizing the expression:

\[
e = \Sigma_{i=1}^{N} (Y_i - \hat{Y}_i)^2 = \Sigma_{i=1}^{N} [Y_i - (a + b.X_i)]^2
\] (4.2)
The error minimization can be achieved using standard techniques from calculus, namely the property that a polynomial expression reaches its minimum value when its derivatives vanish. Taking the partial derivative of $e$ with respect to $a$ and $b$ and setting them to zero gives the following set of equations:

$$\frac{\partial e}{\partial a} = 2a + 2b \sum_{i=1}^{N} X_i - 2 \sum_{i=1}^{N} Y_i = 0 \quad (4.3)$$

$$\frac{\partial e}{\partial b} = 2b \sum_{i=1}^{N} X_i^2 + 2a \sum_{i=1}^{N} X_i - 2 \sum_{i=1}^{N} X_i Y_i = 0 \quad (4.4)$$

Solving Equation 4.3 and Equation 4.4 gives the least square estimates of $a$ and $b$ as:

$$a = N \bar{Y} - b \cdot N \bar{X} \quad (4.5)$$

$$b = \frac{\sum_{i=1}^{N} (Y_i - \bar{Y})(X_i - \bar{X})}{\sum_{i=1}^{N} (X_i - \bar{X})^2} \quad (4.6)$$

In general, least squares can be formulated to identify the estimates of unknown parameters of a mathematical model. For instance, the dynamical system is represented in the form of continuous-time system. Assuming discrete input applied to the continuous-time system is known and its discrete output is measurable. Such a continuous-time system with applied input and measured output in a discrete manner can be viewed from the Figure 4.1.

![Figure 4.1: Continuous time system with applied discrete input and measured discrete output](image)

From the Figure 4.1, the complete dynamical system from discrete input $u_k$ to discrete output $y_k$ is represented by a discrete-time transfer function $H(z)$. The goal was to use the least squares method to find an estimate for the discrete-time transfer function $H(z)$. The applied discrete input and measured discrete output are fed as necessary constituents to the least squares method. Upon which the method will yield an estimate for $H(z)$. The continuous-time system with the least squares method is shown in Figure 4.2. In order
to perform least squares for an unidentified mathematical model, few pre-requisites are needed. The pre-requisites are as follows:

- Measured data of the real world plant
- Discrete-time transfer function, \( H(z) \)

![Figure 4.2: Continuous time system with least squares algorithm](image1)

Assuming measured data of the real world plant is available to the practitioner and the real world process is disturbed. The least squares principle for the estimation of parameters of a dynamical system is illustrated in Figure 4.3.

![Figure 4.3: Least squares principle](image2)

Estimate for \( H(z) \) is given by the following discrete-time transfer function \([23]\),

\[
\hat{H}(z) = \frac{\hat{Y}(z)}{\hat{U}(z)} = \frac{\hat{b}_0 + \hat{b}_1z^{-1} + \ldots + \hat{b}_nz^{-n}}{1 + \hat{a}_1z^{-1} + \ldots + \hat{a}_nz^{-n}} \tag{4.7}
\]

- \( n \) is the model order
From Equation 4.7, an identification model is derived in the following way,

\[ \hat{Y}(z)[1 + \alpha_1 z^{-1} + \ldots + \alpha_n z^{-n}] = U(z)[\hat{b}_0 + \hat{b}_1 z^{-1} + \ldots + \hat{b}_n z^{-n}] \quad (4.8) \]

Applying inverse Z-transforms to Equation 4.8 leads to,

\[ \hat{y}_k + \hat{\alpha}_1 \hat{y}_{k-1} + \ldots + \hat{\alpha}_n \hat{y}_{k-n} = \hat{b}_0 u_k + \hat{b}_1 u_{k-1} + \ldots + \hat{b}_n u_{k-n} \quad (4.9) \]

- Where \( k = 0, 1, \ldots, n - 1 \)

The estimated discrete output \( \hat{y}_k \) is replaced by the measured discrete output \( y_k \) and deviation \( \delta_k \) using the following Equation,

\[ \delta_k = y_k - \hat{y}_k \quad (4.10) \]

Using the Equation 4.9 and Equation 4.10 we get,

\[ \hat{\delta}_k + \hat{\alpha}_1 \hat{\delta}_{k-1} + \ldots + \hat{\alpha}_n \hat{\delta}_{k-n} = y_k + \hat{\alpha}_1 y_{k-1} + \ldots + \hat{\alpha}_n y_{k-n} - \hat{b}_0 u_k - \hat{b}_1 u_{k-1} - \ldots - \hat{b}_n u_{k-n} \quad (4.11) \]

Difference equation of deviation \( \hat{\delta}_k \) is equated to error \( e_k \). Therefore, Equation 4.11 is simplified by introducing \( e_k \) in the following way:

\[ e_k = y_k + \hat{\alpha}_1 y_{k-1} + \ldots + \hat{\alpha}_n y_{k-n} - \hat{b}_0 u_k - \hat{b}_1 u_{k-1} - \ldots - \hat{b}_n u_{k-n} \quad (4.12) \]

Equation 4.12 is re-written as,

\[ e_n = y_n + \hat{\alpha}_1 y_{n-1} + \ldots + \hat{\alpha}_n y_0 - \hat{b}_0 u_n - \hat{b}_1 u_{n-1} - \ldots - \hat{b}_n u_0 \quad (4.13) \]

\[ e_{n+1} = y_{n+1} + \hat{\alpha}_1 y_{n} + \ldots + \hat{\alpha}_n y_{1} - \hat{b}_0 u_{n+1} - \hat{b}_1 u_{n+2} - \ldots - \hat{b}_n u_{1} \quad (4.14) \]

\[ e_N = y_N + \hat{\alpha}_1 y_{N-1} + \ldots + \hat{\alpha}_n y_{N-n} - \hat{b}_0 u_N - \hat{b}_1 u_{N-1} - \ldots - \hat{b}_n u_{N-n} \quad (4.15) \]

- \( N \) is the number of measurements, where \( N \gg n \)

The Equation 4.13, Equation 4.14 and Equation 4.15 are represented in the following matrix notation [20],

\[
\begin{bmatrix}
  e_n \\
  e_{n+1} \\
  \vdots \\
  e_{N-1} \\
  e_N
\end{bmatrix} = 
\begin{bmatrix}
  y_n \\
  y_{n+1} \\
  \vdots \\
  y_{N-1} \\
  y_N
\end{bmatrix} - 
\begin{bmatrix}
  -y_{n-1} & \cdots & -y_0 & u_n & \cdots & u_0 \\
  -y_n & \cdots & -y_1 & u_{n+1} & \cdots & u_1 \\
  \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\
  -y_{N-2} & \cdots & -y_{N-n-1} & u_{N-1} & \cdots & u_{N-n-1} \\
  -y_{N-1} & \cdots & -y_{N-n} & u_N & \cdots & u_{N-n}
\end{bmatrix} \cdot 
\begin{bmatrix}
  \hat{\alpha}_1 \\
  \vdots \\
  \hat{\alpha}_n \\
  \hat{b}_0 \\
  \hat{b}_1 \\
  \vdots \\
  \hat{b}_n
\end{bmatrix}
\]

\[ \hat{\lambda} \quad (4.16) \]
Equation 4.16 can be further simplified into,

\[ e = y - M\hat{\lambda} \]  

(4.17)

• where \( e \) is error vector, \( e \in \mathbb{R}^{(N-n+1) \times 1} \)

• \( y \) is output vector, \( y \in \mathbb{R}^{(N-n+1) \times 1} \)

• \( M \) is the measurement matrix, \( M \in \mathbb{R}^{(2n+1) \times (N-n+1)} \)

• \( \hat{\lambda} \) is parametric vector, \( \hat{\lambda} \in \mathbb{R}^{(2n+1) \times 1} \)

Therefore, error minimization problem can be introduced using the Equation 4.17 in the following manner:

\[ e^T e = e_1^2 + e_2^2 + \ldots + e_N^2 \]  

(4.18)

Substituting Equation 4.17 into Equation 4.18 leads to,

\[ e^T e = y^T y - 2y^T M\hat{\lambda} + \hat{\lambda}^T M^T M\hat{\lambda} \]  

(4.19)

Taking the partial derivative of Equation 4.19 with respect to \( \hat{\lambda} \) and setting it to zero gives the following equation,

\[ \frac{\partial e^T e}{\partial \hat{\lambda}} = 0 - 2M^T y - 2M^T M\hat{\lambda} = 0 \]  

(4.20)

Therefore, the parametric vector \( \hat{\lambda} \) could be calculated in the following way [14]:

\[ \hat{\lambda} = (M^T M)^{-1} M^T y \]  

(4.21)

Hence, the estimate for \( H(z) \) or in other words the unknown parameters of the mathematical model could be calculated from the parametric vector \( \hat{\lambda} \).

### 4.3 Simulink Design Optimization

Simulink design optimization is an interactive user-friendly tool which lets the professional to run parameter estimation tasks of an arbitrary dynamical system realized as a simulink model. Using optimization techniques, simulink design optimization estimates the parameter and (optionally) initial conditions of states to minimize a user-selected cost function. The cost function typically calculates a least-square error between the empirical and model data signals. The following requisites need to be fulfilled to even prepare an estimation task for a realized simulink model:

• Transient data

• Variables

• Optimization method
The measured data of the MML system is defined in the transient data part. The unidentified parameters of the simulink model are added in the variables part. In the variables part, minimum and maximum range for each unidentified parameter needs to be specified. The minimum and maximum range is specified on the basis of the estimates found out from least squares method. These estimates are used as the initial values for each and every undetermined parameter. An optimization method has to be assigned which in turn yields closer estimates for the unidentified parameters. Simulink design optimization adopts the optimization method to minimize the cost function over several iterations resulting into a close match between the measured data and simulated data. Since nonlinear effect of PS flow is evident with the mathematical model of MML system, the optimization method was selected on the basis of the following:

- Nonlinear least squares method
- Levenberg-Marquardt algorithm

### 4.3.1 Nonlinear Least Squares Method

Nonlinear least squares method is an extension to least squares method to determine the unidentified parameters of a nonlinear model. Consider a set of m data observations with a model that is nonlinear in n unknown parameters with $m \geq n$. The nonlinear model is assumed in the following form [19],

$$y_i = f(x_i, \lambda)$$  

- $f(x_i, \lambda)$ is a nonlinear function
- $y$ is dependent variable, $i = 1, 2, \ldots, m$
- $x$ is independent variable
- $\lambda$ is the parametric vector

The parameter vector $\lambda$ can be found by formulating the sum of the squares of the error between the data observations $y_i$ and the nonlinear model $f(x_i, \lambda)$ in a least squares sense. The sum of the squares is given by,

$$S = \Sigma_{i=1}^{m} e_i^2 = \Sigma_{i=1}^{m} (y_i - f(x_i, \lambda))^2$$  

- $e$ is the error

Taking the partial derivative of $S$ and equating it to zero gives the minimum value of $S$. Since the nonlinear model contains $n$ parameters there are $n$ gradient equations:

$$\frac{\partial S}{\partial \lambda_j} = 2\sum_{i=1}^{m} e_i \frac{\partial e_i}{\partial \lambda_j} = 0$$  

- $j = 1, 2, \ldots, n$
In a nonlinear system, the partial derivatives are functions of both independent variables and the parameters as well. Therefore, the gradient equations do not have a closed solution. Instead, initial values must be chosen for the parameters. Then, the identification of the parameters is improved iteratively, i.e., the values are obtained by successive approximation,

\[ \lambda_j \approx \lambda_j^{k+1} = \lambda_j^k + \Delta \lambda_j \]  

- \( k \) is the iteration number
- \( \Delta \lambda_j \) is the parametric shift vector

At each iteration, the model is linearized by approximating it to a first order Taylor series expansion about \( \lambda^k \).

\[
f(x_i, \lambda) \approx f(x_i, \lambda^k) + \Sigma_{j=1}^n \frac{\partial f(x_i, \lambda^k)}{\partial \lambda_j} (\lambda_j - \lambda_j^k) \]  

\[
f(x_i, \lambda^k) + \Sigma_{j=1}^n \frac{\partial f(x_i, \lambda^k)}{\partial \lambda_j} (\lambda_j - \lambda_j^k) \approx f(x_i, \lambda^k) + \Sigma_{j=1}^n J_{ij} \Delta \lambda_j \]  

- \( J_{ij} \) consist of row and vector elements of the Jacobian \( J \)

Jacobian \( J \) is the resultant matrix from the partial derivatives of the nonlinear function with respect to the parametric vector \( \lambda \) and it changes from one iteration to the next. Therefore, the errors and the linearized model are given by the following equations,

\[ e_i = \Delta y_i - \Sigma_{l=1}^n J_{il} \Delta \lambda_l \]  

\[ \Delta y_i = y_i - f(x_i, \lambda^k) \]

\[ \frac{\partial e_i}{\partial \Delta \lambda_j} = -J_{ij} \]  

Substituting the above expressions into the gradient equations, we get

\[-2 \Sigma_{i=1}^m J_{ij}(\Delta y_i - \Sigma_{l=1}^n J_{il} \Delta \lambda_l) = 0 \]

On rearrangement, the above equation results into \( n \) simultaneous linear equations. The normal equations are as follows:

\[ \Sigma_{i=1}^m \Sigma_{l=1}^n J_{ij} J_{il} \Delta \lambda_l = \Sigma_{i=1}^m J_{ij} \Delta y_i \]  

The normal equations can be represented in the following matrix notation,

\[ (J^T J) \Delta \lambda = J^T \Delta y \]  

Hence, the parametric shift vector \( \Delta \lambda \) is given by the following equation,

\[ \Delta \lambda = (J^T J)^{-1} J^T \Delta y \]
4.3.2 Levenberg-Marquardt Algorithm

The Levenberg-Marquardt method is a standard technique used to solve nonlinear least squares problems. Least squares problems arise when fitting a parameterized function to a set of measured data points by minimizing the sum of the squares of the errors between the data points and the function. Nonlinear least squares problems arise when the function is not linear in the parameters. Nonlinear least squares methods involve an iterative improvement to the parameter estimates in order to reduce the sum of the squares of the errors between the function and the measured data points. Given a set of \( m \) data pairs of independent variable \( x_i \) and dependent variable \( y_i \). The objective is to optimize the parameters \( \lambda \) of the nonlinear model by minimizing the nonlinear least squares problem.

The nonlinear squared problem is formulated in the following form [15],

\[
S(\lambda) = \sum_{i=1}^{m} (y_i - f(x_i, \lambda))^2 \tag{4.34}
\]

- \( f(x_i, \lambda) \) is a nonlinear model, \( i = 1, 2, \ldots, m \)

The Levenberg-Marquardt algorithm requires an initial guess for the parametric vector \( \lambda \) which needs to be estimated. At each iteration step, the parametric vector \( \lambda \) is replaced by a new estimate which can be seen from the following equation,

\[
f(x_i, \lambda) \approx f(x_i, \lambda + \varrho) \tag{4.35}
\]

- \( \lambda + \varrho \) is the new estimate
- \( \varrho \) is the parametric shift vector

To determine \( \varrho \), the function \( f(x, \lambda + \varrho) \) is approximated through first order Taylor series expansion.

\[
f(x_i, \lambda + \varrho) = f(x_i, \lambda) + J \varrho \tag{4.36}
\]

- \( J \) is the Jacobian matrix, \( J = \frac{\partial f(x_i, \lambda)}{\partial \lambda} \)

Now the nonlinear squares problem with respect to \( \lambda + \varrho \) is computed in the following manner,

\[
S(\lambda + \varrho) \approx \sum_{i=1}^{m} (y_i - f(x_i, \lambda) - J \varrho)^2 \tag{4.37}
\]

Taking the partial derivative of Equation 4.37 with respect to \( \varrho \) and setting it to zero gives the minimum value of \( S \).

\[
\frac{\partial S(\lambda + \varrho)}{\partial \varrho} = -2(y_i - f(x_i, \lambda) - J \varrho) J^T = 0 \tag{4.38}
\]

Upon simplification the above equation can be written in the following matrix notation,

\[
(J^T J) \varrho = J^T [y - f(\lambda)] \tag{4.39}
\]
Levenberg replaced the above equation by introducing an algorithmic parameter $\Gamma$ which adaptively varies the parameter updates between the Gradient Descent (see Section 7.3.1) and Gauss-Newton (see Section 7.3.2) update,

$$(J^T J + \Gamma I) \varrho = J^T [y - f(\lambda)] \quad (4.40)$$

Where small values of the algorithmic parameter $\Gamma$ result in a Gauss-Newton update and large values of $\Gamma$ result in a gradient descent update. At a large distance from the function minimum, the Gradient Descent method is utilized to provide steady and convergent progress toward the solution. As the solution approaches minimum, $\Gamma$ is adaptively decreased, the Levenberg-Marquardt method approaches the Gauss-Newton method, and the solution typically converges rapidly to the local minimum. Therefore, Marquardt replaced the identity matrix $I$ with a diagonal matrix consisting of the diagonal elements of $J^T J$, resulting in the Levenberg-Marquardt algorithm:

$$[J^T J + \Gamma \text{diag}(J^T J)] \varrho = J^T [y - f(\lambda)] \quad (4.41)$$

The estimates of parametric shift vector $\varrho$ are determined by solving the above system of linear equations. Hence, the estimates of $\varrho$ are added to the initial estimates of $\lambda$ to yield optimum values for $\lambda$.

### 4.4 Parameter Estimation of MML System

Identification of the unknown parameters of the mathematical model of MML system had to come across few challenges. To even carry out the least squares method appropriately, mathematical model of MML system was not simple enough. The first challenge was to identify the inequality expression introduced by the nonlinear function $f_{ps}(W_{ps}, Q_{ms})$ which describes the existence of PS fluid flow. The second challenge was that there happens to be a constant MS fluid flow with respect to the constant pressure differential $P$. The parameter involved with this particular dependency over MS fluid flow needs to be estimated by making certain changes to the mathematical model of MML system. The third challenge was to carry out the linear approximation of the nonlinear function $f_{ps}(W_{ps}, Q_{ms})$ associated with the mathematical model of MML system in an appropriate manner. Parameter estimation of MML system was carried out in the following three steps:

- Identification of numerical constants
- Identification of constant pressure differential parameter
- Identification of MS and PS parameters

**Identification of Numerical Constants**

According to the nonlinear function $f_{ps}(W_{ps}, Q_{ms})$, the numerical constants $\alpha_i$ and $\beta_i$ of the inequality expression which is dependent upon PS pump speed $W_{ps}$ and MS fluid flow $Q_{ms}$, are identified by an iterative experiment performed on MML system. The iterative experiment was practiced with several MS fluid flows being acquired throughout
its preferred operating range (i.e., $Q_{ms,op} = (8000 - 10000) cm^3/min$) and simultaneously the PS pump speed was applied in such a way that the PS fluid flow is about to exist. In the first scenario, list of data points were collected representing the same PS fluid flow at various MS fluid flows and PS pump speeds. A linear fit of these data points was carried out using least squares approach \[5\] resulting into a straight line which can be seen from the Figure 4.4.

Estimates of the numerical constants $\alpha_i$ and $\beta_i$ are determined from the equation of the straight line. Figure 4.4 shows the PS fluid flow data points under the first scenario (i.e., PS has a maximum flow of 4500 $cm^3/min$ with respect to its orifice size of $3mm$). Similarly, $\alpha_i$ and $\beta_i$ were estimated for the remaining two scenarios as well. Estimated values of the numerical constants $\alpha_i$ and $\beta_i$ under the three scenarios is shown in Table 4.1.

<table>
<thead>
<tr>
<th>i(scenario)</th>
<th>PS Orifice Size(mm)</th>
<th>PS Full Scale Flow(cm$^3$/min)</th>
<th>$\alpha_i$</th>
<th>$\beta_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>4500</td>
<td>0.49</td>
<td>9.6e+03</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>750</td>
<td>0.42</td>
<td>8.9e+03</td>
</tr>
<tr>
<td>3</td>
<td>0.45</td>
<td>150</td>
<td>0.47</td>
<td>9.5e+03</td>
</tr>
</tbody>
</table>

Table 4.1: Estimates of coefficients $\alpha_i$ and $\beta_i$ with respect to three scenarios

Identification of Constant Pressure Differential Parameter

The constant pressure differential parameter $k_0$ is responsible for the constant MS flow at zero MS pump speed. Identification of parameter $k_0$ doesn’t require the complete mathematical model of MML system. Instead, the MS pump speed and PS fluid flow dependencies over MS fluid flow are neglected. To find out an estimate for the unknown parameter $k_0$, step response data of MS fluid flow is acquired with respect to the constant pressure differential $P$ by deactivating the PS fluid path at zero MS pump speed. The step response is shown in Figure 4.5.
From the Figure 4.5, MS fluid flow dynamics are described by a first order model in the following way [6, 7].

\[
\frac{Q_{ms}(s)}{P(s)} = \frac{k_0}{s + k_1} \tag{4.42}
\]

Using the forward rectangular rule (see Section 7.4.1), a discrete-time transfer function is computed by replacing the Laplacian operator \(s\) with \((z - 1)/T_s\) in Equation 4.42.

\[
\frac{Q_{ms}(z)}{P(z)} = H(z) = \frac{k_0.T_s}{z - 1 + k_1.T_s} \Rightarrow \hat{H}(z) = \frac{\hat{b}_0}{z + \hat{a}_0} \tag{4.43}
\]

- Sampling time \(T_s = 0.05s\)
- \(\hat{H}(z)\) is an estimate of \(H(z)\)
- Where \(\hat{b}_0 = k_0.T_s\), \(\hat{a}_0 = k_1.T_s - 1\)

\(\hat{H}(z)\) and the measured data of Figure 4.5 are provided as the required elements to the least squares method. Therefore, the estimates of the parameters \(k_0\) and \(k_1\) are computed from \(\hat{b}_0\) and \(\hat{a}_0\) by using the least squares method [14]. The identified parameters \(k_0\) and \(k_1\) are shown in Table 4.2.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Estimates</th>
</tr>
</thead>
<tbody>
<tr>
<td>(k_0)</td>
<td>0.082</td>
</tr>
<tr>
<td>(k_1)</td>
<td>1.3544</td>
</tr>
</tbody>
</table>

Table 4.2: Estimates of parameters \(k_0\) and \(k_1\)
CHAPTER 4. PARAMETER ESTIMATION

Identification of MS and PS Parameters

The MS and PS fluid flow dependent parameters were identified on the basis of simulink design optimization and least squares method. Initially, an assumption was made where the inequality expression of the nonlinear function $f_{ps}(W_{ps}, Q_{ms})$ holds true with respect to the PS fluid flow operating range. The mathematical model of MML system was simplified under the following assumptions:

- Assuming the PS fluid flow exists with respect to its operational pump speed
- Linear approximation of nonlinear function i.e., $f_{ps}(W_{ps}, Q_{ms}) = k_{5i}W_{ps} - k_{6i}Q_{ms}$
- Constant pressure differential was regarded as a virtual input to MML system

Mathematical model of MML system is represented in the following state space model notation,

$$
\begin{bmatrix}
\dot{Q}_{msi} \\
\dot{Q}_{psi}
\end{bmatrix} = \begin{bmatrix}
-k_1 & -k_{3i} \\
-k_{6i} & -k_{4i}
\end{bmatrix} \begin{bmatrix}
Q_{msi} \\
Q_{psi}
\end{bmatrix} + \begin{bmatrix}
k_2 & 0 & k_0 \\
0 & k_5 & 0
\end{bmatrix} \begin{bmatrix}
W_{ms} \\
W_{ps} \\
P
\end{bmatrix} \tag{4.44}
$$

- Where $i = 1$ represents the MS and PS fluid flows in the first scenario
- $P$ is virtual input

A multi-input multi-output continuous-time transfer matrix is constructed by considering the MS and PS fluid flows $Q_{msi}, Q_{psi}$ as outputs and MS and PS pump speeds $W_{ms}, W_{ps}$ as inputs. The continuous-time transfer matrix is given by the following,

$$
G_{mn}(s) = \begin{bmatrix}
G_{11}(s) & G_{12}(s) \\
G_{21}(s) & G_{22}(s)
\end{bmatrix} = \begin{bmatrix}
\frac{\gamma_1.s + \gamma_0}{s^2 + a_1.s + a_0} & \frac{\mu_0}{s^2 + a_1.s + a_0} \\
\frac{s^2 + a_1.s + a_0}{s^2 + a_1.s + a_0} & \frac{s^2 + a_1.s + a_0}{s^2 + a_1.s + a_0}
\end{bmatrix} \tag{4.45}
$$

- $m$ represents the number of outputs and $n$ represents the number of inputs
- $\gamma_1 = k_2, \gamma_0 = k_{4i}.k_2, a_1 = k_1 + k_{4i}, a_0 = k_1.k_{4i} - k_{3i}.k_{6i}$
- $\mu_0 = -k_{3i}.k_{5i}, \nu_0 = -k_{6i}.k_2, \eta_1 = k_{5i}, \eta_0 = k_1.k_{5i}$

Using the forward rectangular rule (see Section 7.4.1), a discrete-time transfer matrix is computed by replacing the Laplacian operator $s$ with $(z - 1)/T_s$ in Equation 4.45

$$
H_{mn}(z) = \begin{bmatrix}
H_{11}(z) & H_{12}(z) \\
H_{21}(z) & H_{22}(z)
\end{bmatrix} = \begin{bmatrix}
\frac{\vartheta_1.z + \vartheta_0}{z^2 + \sigma_1.z + \sigma_0} & \frac{\chi_0}{z^2 + \sigma_1.z + \sigma_0} \\
\frac{z^2 + \sigma_1.z + \sigma_0}{z^2 + \sigma_1.z + \sigma_0} & \frac{\nu_1.z + \nu_0}{z^2 + \sigma_1.z + \sigma_0}
\end{bmatrix} \tag{4.46}
$$

- Sampling time $T_s = 0.05s$
- $\vartheta_1 = \gamma_1.T_s, \vartheta_0 = \gamma_0.T_s^2 - \gamma_1.T_s, \sigma_1 = a_1.T_s - 2, \sigma_0 = a_0.T_s^2 - a_1.T_s + 1$
- $\chi_0 = \mu_0.T_s^2, \xi_0 = \nu_0.T_s^2, \nu_1 = \eta_1.T_s, \nu_0 = \eta_0.T_s^2 - \eta_1.T_s$
An estimate of the discrete-time transfer matrix $H(z)$ is specified in the following way,

$$
\hat{H}_{mn}(z) = \begin{bmatrix}
\hat{H}_{11}(z) & \hat{H}_{12}(z) \\
\hat{H}_{21}(z) & \hat{H}_{22}(z)
\end{bmatrix} = \begin{bmatrix}
\hat{\vartheta}_1 z + \hat{\sigma}_0 \\
\frac{\hat{\chi}_0}{z^2 + \sigma_1 z + \sigma_0}
\end{bmatrix}
\begin{bmatrix}
\hat{\chi}_0 \\
\frac{\hat{\vartheta}_1 z + \hat{\sigma}_0}{z^2 + \sigma_1 z + \sigma_0}
\end{bmatrix}
$$

(4.47)

The measured data of experiment II (i.e., Figure 3.10) mentioned in Section 3.4.1 and the discrete transfer matrix $\hat{H}_{mn}(z)$ are the required constituents for least squares method. Thereupon, the initial estimates of the unidentified parameters of the mathematical model of MML system were computed using the least squares method.

**Scenario 1**

In the first scenario, PS has a maximum flow of $4500\text{cm}^3/\text{min}$ with respect to its orifice size of $3\text{mm}$. The initial estimates of MS and PS parameters found out from the least squares method are then provided as a pre-requisite to simulink design optimization. Using the optimization method, simulink design optimization performs several iterations to minimize the defined cost function in order to give more accurate estimates of MS and PS parameters in this scenario. The estimates of MS and PS parameters in the first scenario are shown in Table 4.3.

<table>
<thead>
<tr>
<th>Parameters (scenario 1)</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_1$</td>
<td>1.3663</td>
</tr>
<tr>
<td>$k_2$</td>
<td>0.6</td>
</tr>
<tr>
<td>$k_{31}$</td>
<td>0.8</td>
</tr>
<tr>
<td>$k_{41}$</td>
<td>0.6498</td>
</tr>
<tr>
<td>$k_{51}$</td>
<td>0.405</td>
</tr>
<tr>
<td>$k_{61}$</td>
<td>0.18</td>
</tr>
</tbody>
</table>

Table 4.3: Estimates of MS and PS parameters with respect to scenario 1

**Scenario 2**

In the case of second scenario, PS has a maximum flow of $750\text{cm}^3/\text{min}$ with respect to its orifice size of $1\text{mm}$. A scaling factor was introduced in the realized simulink model to depict the change of orifice size from previous scenario to scenario 2. The scaling factor was computed based on the maximum flows of PS in the following manner.

$$
c_{f_2} = \frac{Q_{p_{s\text{max},1}}}{Q_{p_{s\text{max},2}}}
$$

(4.48)

- $c_{f_2}$ is the scaling factor introduced in scenario 2
- $Q_{p_{s\text{max},1}} = 4500\text{sccm}$ is the maximum PS flow in scenario 1
- $Q_{p_{s\text{max},2}} = 750\text{sccm}$ is the maximum PS flow in scenario 2
Using the simulink design optimization, measured data of the MML system obtained in scenario 2 is defined in the transient part. The minimum and maximum ranges and also the initial values for each and every unidentified parameter are the same which were earlier defined in scenario 1. The parameter estimation task in this scenario was carried out using the specified transient data and computed scaling factor $c_{f_2}$. In turn, the estimates of MS and PS parameters with respect to this scenario were found. The estimates of MS and PS parameters in the second scenario are shown in Table 4.4.

<table>
<thead>
<tr>
<th>Parameters(scenario 2)</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_{32}$</td>
<td>0.12</td>
</tr>
<tr>
<td>$k_{42}$</td>
<td>0.7</td>
</tr>
<tr>
<td>$k_{52}$</td>
<td>0.44</td>
</tr>
<tr>
<td>$k_{62}$</td>
<td>0.16</td>
</tr>
</tbody>
</table>

Table 4.4: Estimates of MS and PS parameters with respect to scenario 2

**Scenario 3**

In the third scenario, PS has a maximum flow of $150 cm^3/min$ with respect to its orifice size of 0.45 mm. Here also a scaling factor was introduced in the realized simulink model to depict the change in orifice size from scenario 1 to scenario 3. Similarly, the scaling factor was computed based on the maximum flows of PS in the following manner.

$$c_{f_3} = \frac{Q_{ps_{max,1}}}{Q_{ps_{max,3}}}$$

- $c_{f_3}$ is the scaling factor introduced in scenario 3
- $Q_{ps_{max,3}} = 150 sccm$ is the maximum PS flow in scenario 3

Using the simulink design optimization, measured data of the MML system obtained in scenario 3 is defined in the transient part. To find respective estimates of MS and PS parameters in this scenario, same steps were repeated which were earlier performed in scenario 2. The estimates of MS and PS parameters in the first scenario are shown in Table 4.5.

<table>
<thead>
<tr>
<th>Parameters(scenario 3)</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_{33}$</td>
<td>0.08</td>
</tr>
<tr>
<td>$k_{43}$</td>
<td>0.75</td>
</tr>
<tr>
<td>$k_{53}$</td>
<td>0.48</td>
</tr>
<tr>
<td>$k_{63}$</td>
<td>0.18</td>
</tr>
</tbody>
</table>

Table 4.5: Estimates of MS and PS parameters with respect to scenario 3
4.5 Identification Results

A comparison between the behavior exhibited by the real world plant and its mathematical model was performed. Figure 4.6, Figure 4.7 and Figure 4.8 shows various MS and PS fluid flows with respect to the applied MS and PS pump speeds under the three scenarios.

Figure 4.6: Measured v/s simulation of MS and PS fluid flows with respect to scenario 1

Figure 4.6 shows the measured and simulated MS and PS flows in the first scenario. Based on the error computed between measured and simulated MS and PS flows, the model seems to be generating the same kind of dynamics but relatively slower than the real world plant. This is due to the uncertain nature of MS and PS flow parameters $k_{31}$ and $k_{61}$ which were introduced as time independent parameters. In the case of first scenario, the model is regarded as a compromise of the reality.
Figure 4.7 shows the measured and simulated MS and PS flows in the second scenario. The MS and PS flow errors computed in the second scenario are comparatively less than that of the first scenario. This huge decrease in the MS and PS flow errors is due to the change in the PS orifice size. In the case of second scenario, the model seems to be generating the same kind of dynamics but relatively slower than the real world plant. This is due to the uncertain nature of MS and PS flow parameters $k_{32}$ and $k_{62}$ which were introduced as time independent parameters. In the case of second scenario, the model is regarded as a compromise of the reality.

Figure 4.7: Measured v/s simulation of MS and PS fluid flows with respect to scenario 2
Figure 4.8 shows the measured and simulated MS and PS flows in the third scenario. The MS and PS flow errors computed in the third scenario are comparatively less than that of the second scenario. This huge decrease in the MS and PS flow errors is due to the change in the PS orifice size. In the case of third scenario, the model seems to be generating the same kind of dynamics but relatively slower than the real world plant. This is due to the uncertain nature of MS and PS flow parameters $k_{33}$ and $k_{63}$ which were introduced as time independent parameters. In the case of third scenario, the model is regarded as a compromise of the reality.

Figure 4.8: Measured v/s simulation of MS and PS fluid flows with respect to scenario 3
Chapter 5

Model Based Controller Design

5.1 Introduction

To have the control of a physical quantity over a wide operating range, a controller must be designed. Control design is defined as the method which provides the input signal of a dynamical system capable of manipulating the physical quantity over time to perform its desired action. After completing the parameter estimation procedure, control design is the next step to be carried out on a dynamical system. There are no strict guidelines about which kind of controller needs to be designed for a particular dynamical system. The control design is dependent upon the mathematical representation of dynamical system and its desired functionality which needs to be performed over a specific application. Typically, control design is practiced in the following two ways:

- Feed forward control design
- Feedback control design

Feed Forward Control Design

Feed forward control design is a open loop control technique which computes the plant input based on its pre-determined mathematical model (i.e., plant model which depict the complete system dynamics) and measured disturbances. Feed forward control design is applicable to those kinds of systems where the input to the plant is known in advance which is required to drive the output to reach the assigned reference within a finite time. A typical structure of the plant with feed forward control can be seen in Figure 5.1. With the feed forward control, the disturbances are measured and accounted before they have time to affect the system. Here the output which needs to be controlled is not error based. Instead, it is based on the knowledge about the system in the form of a mathematical model and knowledge about the process disturbances. In practice, there will be some disturbances acting on the real world plant which are relatively unknown and immeasurable as well. In this case, feed forward control design would lead to system failures. Moreover, the feed forward control design is unaware of how well the plant is behaving.
Feedback Control Design

Feedback control design is a closed loop control technique which computes the plant input based on the error calculated from the difference between the actual output and assigned reference. With feedback control, there is no need to measure the disturbance. The actual output itself would be containing process disturbance. Feedback control design automatically follows change in the reference. A typical structure of the plant with the feedback control design can be seen in Figure 5.2. In general, feedback control design doesn’t require the complete knowledge of the system. In practice, feedback control design is the most preferred way of designing a controller for a particular dynamical system.
5.2 PI Control

Proportional-integral (PI) control is the chosen control design strategy for the MML system. PI control consists of a proportional part and an integral part. Typical mathematical representation of a PI controller is given by Equation 5.1

\[ u(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau \] (5.1)

- \( u(t) \) is the control output
- \( e(t) \) is the error computed between the desired output (setpoint) and actual output
- \( k_p \) is the proportional gain
- \( k_i \) is the integral gain

The proportional gain \( k_p \) computes the control output which is directly proportional to the current value of the error signal. The integral gain \( k_i \) calculates the control output based upon the integral of the error signal computed between the previous and current time instants. Proper tuning of PI control parameters (i.e., \( k_p \) and \( k_i \)) leads to better performance of the overall system (i.e., plant+controller in closed loop form). There are several methods to determine the PI control parameters. To compute a robust PI parameter setting for MML system with respect to the change of PS orifice under the three scenarios, a control algorithm was devised on the basis of the following adopted method:

- Pole placement for unity feedback loop

![Unity feedback loop diagram](image)

Figure 5.3: Unity feedback loop

5.3 Pole Placement for Unity Feedback Loop

A typical structure of a unity feedback loop can be seen from the Figure. In Figure, real world plant and the controller are realized by their individual transfer functions \( P(s) \) and \( C(s) \) respectively. Assigned set point (desired output) and actual output is
denoted by \( r \) and \( y \) respectively. Error computed between the desired output and actual output is represented by \( e \). Transfer functions of the real world plant \( P(s) \) and controller \( C(s) \) are given by the following,

\[
P(s) = \frac{\mu(s)}{\nu(s)} = \frac{\mu_{n-1}s^{n-1} + \ldots + \mu_1s + \mu_0}{\nu_n s^n + \nu_{n-1}s^{n-1} + \ldots + \nu_1s + \nu_0} \tag{5.2}
\]

- Where \( \text{deg} \mu(s) < \text{deg} \nu(s) \)
- \( \text{deg} \nu(s) = n \), \( n \) is plant order

\[
C(s) = \frac{b(s)}{a(s)} = \frac{b_n s^n + b_{n-1}s^{n-1} + \ldots + b_1s + b_0}{a_n s^n + a_{n-1}s^{n-1} + \ldots + a_1s + a_0} \tag{5.3}
\]

- Where \( \text{deg} b(s) = \text{deg} a(s) = \delta \), \( \delta \) is controller order

The relation between the set point \( r \) and the actual output \( y \) is given by the overall transfer function \( T(s) \). \( T(s) \) can be computed in the following manner:

\[
T(s) = \frac{C(s)P(s)}{1 + C(s)P(s)} = \frac{\mu_T(s)}{\nu_T(s)} \tag{5.4}
\]

Substituting Equation 5.2 and Equation 5.3 into Equation 5.4 we get,

\[
T(s) = b(s)\mu(s) \quad \frac{\mu_T(s)}{\nu_T(s)} \tag{5.5}
\]

\[
T(s) = \frac{b(s)\mu(s)}{a(s)\nu(s) + b(s)\mu(s)} \quad \frac{\mu_T(s)}{\nu_T(s)} \tag{5.6}
\]

Two design relations were deduced from Equation 5.6,

\[
b(s)\mu(s) = \mu_T(s), \tag{5.7}
\]
\[
a(s)\nu(s) + b(s)\mu(s) = \nu_T(s) \tag{5.8}
\]

Based on the latter design relation, denominator polynomial \( \nu_T(s) \) of the overall transfer function \( T(s) \) is represented in the following way:

\[
\nu_T(s) = f_{n+\delta}s^{n+\delta} + \ldots + f_1s + f_0 \tag{5.9}
\]

According to the latter design relation and Equation 5.9, there happens to be \( 2\delta+2 \) unknowns and \( n+\delta+1 \) equations. A unique relation can be obtained between the plant order \( n \) and controller order \( \delta \), i.e.,

\[
\delta = n - 1, \tag{5.10}
\]

From the above relation it was evident that if plant order is \( n \) then controller order has to be chosen \( n - 1 \). Also, the order of the denominator polynomial \( \nu_T(s) \) of \( T(s) \)
needs to be the sum of plant order and controller order. A kind of matrix notation can be formulated depending upon the latter design relation and Equation 5.9 [4].

\[
\begin{bmatrix}
\nu_0 & 0 & \cdots & 0 & \mu_0 & 0 & \cdots & 0 \\
\nu_1 & \nu_0 & \cdots & 0 & 0 & \mu_0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots \\
\nu_n & \cdots & \nu_0 & 0 & \mu_{n-1} & \cdots & \mu_0 & 0 \\
0 & \nu_n & \cdots & \nu_1 & 0 & 0 & \cdots & \vdots \\
0 & 0 & \cdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots \\
0 & 0 & \cdots & \nu_n & 0 & 0 & \cdots & 0 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
a_0 \\
a_1 \\
\vdots \\
a_\delta \\
b_0 \\
b_1 \\
\vdots \\
b_\delta \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
f_0 \\
f_1 \\
\vdots \\
f_{n+\delta-1} \\
f_{n+\delta} \\
\end{bmatrix}
\]

- \( R \) is coefficient matrix, \( R \in (n + \delta + 1) \times (2\delta + 2) \)
- \( c_p \) is parametric vector of controller \( C(s) \), \( c_p \in (2\delta + 2) \times 1 \)
- \( f \) is coefficient vector of \( \nu_T(s) \), \( f \in (n + \delta + 1) \times 1 \)

The computation of controller parameters from Equation 5.11 is based on the following pre-requisites [4]:

- Inverse of coefficient matrix \( R \) should exist, i.e., \( R \) should be strictly a square matrix
- Appropriate choice of the overall transfer function \( T(s) \)

Choice of Overall Transfer Function

Overall transfer function \( T(s) \) plays a key role in determining the parameters of a PI controller. The overall transfer function \( T(s) \) describes the actual system(i.e., plant) dynamics with the controller in closed loop form. The main idea behind the choice of \( T(s) \) is originated from the classical example of a second order model. A typical second order model can be represented by the following transfer function,

\[
T(s) = \frac{\mu_T(s)}{\nu_T(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}
\]

- Where \( \zeta \) is damping factor, \( 0 < \zeta < 1 \)
- \( \omega_n \) is natural frequency, \( \omega_n > 0 \)

Here the order of the denominator polynomial \( \nu_T(s) \) of the overall transfer function \( T(s) \) is 2. Based on the realization of overall transfer function \( T(s) \) in Equation 5.12 this would be the ideal choice for a plant order(\( n \)) of 1 and controller order(\( \delta \)) of 1. Numerical determination of damping factor \( \zeta \) and natural frequency \( \omega_n \) depends upon the selection of certain closed loop time-domain specifications such as overshoot(\( M_p \)) and rise time(\( t_r \)).
An arbitrary choice of the closed loop time-domain specifications is not possible. A numerical simulation was performed to find out a relation between the closed loop time domain specifications \((M_p \text{ and } t_r)\) and the parameters \((\zeta \text{ and } \omega_n)\) of the overall transfer function \(T(s)\). The numerical simulation was carried out based on the following assumptions,

- \(0.1 \leq \zeta \leq 0.9\)
- \(\omega_n = 4 \text{rad/s}\)

The overshoots and rise times were computed with respect to the above natural frequency and each damping factor chosen within its parametric interval. Based on the iteration process, two graphical solutions were obtained which can be seen from Figure 5.4 and Figure 5.5.

Figure 5.4: Graphical solution I-Damping factor \(\zeta \text{ v/s overshoot } M_p\)

![Damping factor v/s Overshoot](image)

\[
\omega_n * t_r = 1.3\zeta^2 + 0.65\zeta + 1.2
\]

Figure 5.5: Graphical solution II-Damping factor \((\zeta) \text{ v/s product of natural frequency } (\omega_n, t_r) \text{ and its quadratic fit}\)

![Damping factor v/s Product of Natural frequency and Rise time](image)

Quadratic fit
From the Figure 5.4 it is clear that selecting a high damping factor would lead to small amount of overshoot. From the Figure 5.5, the curve resulted from the iteration process is approximated by performing a quadratic fit. Upon which the product of natural frequency and rise time is defined by the following quadratic expression,

\[ \omega_n t_r = 1.3 \zeta^2 + 0.64 \zeta + 1.2 \]  

implies,

\[ \omega_n = \frac{1.3 \zeta^2 + 0.64 \zeta + 1.2}{t_r} \]  

- Where \( t_r < t_{RC} \), \( t_{RC} \) is time constant of actual system response (i.e., \( P(s) \))

Rise time \( t_r \) was chosen to be less than the time constant \( t_{RC} \) of the actual system response \( P(s) \) so that the overall system (i.e., actual system + controller) dynamics within the closed loop can be improved. In order to guarantee this particular direct choice of rise time \( t_r \), an additional factor \( \Lambda \) was introduced in the following manner.

\[ t_r = \Lambda t_{RC} \]  

- Where \( 0 < \Lambda < 1 \)

Therefore, the natural frequency can be computed from Equation 5.14 and Equation 5.15 with respect to the proper choice of rise time \( t_r \). Based on the determination of \( \zeta \) and \( \omega_n \), overall transfer function \( T(s) \) can be numerically formulated from the Equation 5.12.

### 5.4 PI Control Design of MML system

#### 5.4.1 Challenges

To determine the PI control parameters of MML system, mathematical model needs to be altered. At first, the existence of PS fluid flow with respect to the operating range of PS pump speed and MS fluid flow was assumed. Constant pressure differential \( P \) which was considered as a virtual input is inactive in the design of MS fluid flow. Simplification of the mathematical model of MML system is carried out under the following terms:

- Linear approximation of nonlinear function i.e., \( f_{ps_i}(W_{ps}, Q_{ms_i}) = k_5i.W_{ps} - k_6i.Q_{ms_i} \)
- Coupling associated with both the fluid flows is neglected

A more simple and diminished version of the mathematical model of MML system is given by the following,

\[ \dot{Q}_{ms_i} = -k_1.Q_{ms_i} + k_2.W_{ms} \]  

\[ \dot{Q}_{ps_i} = -k_4i.Q_{ps_i} + k_5i.W_{ps} \]
From Equation 5.16 and Equation 5.17, mathematical model of MML system can be classified into two single-input single-output (SISO) subsystems independent of each other. Applying Laplace transforms to Equation 5.16 and Equation 5.17, transfer functions of both the SISO subsystems are obtained.

\[ P_{ms_i}(s) = \frac{Q_{ms_i}(s)}{W_{ms}(s)} = \frac{k_2}{s + k_1} \]  

(5.18)

\[ P_{ps_i}(s) = \frac{Q_{ps_i}(s)}{W_{ps}(s)} = \frac{k_{5i}}{s + k_{4i}} \]  

(5.19)

Here the plant order of both the deduced SISO subsystems is 1. According to the relation between the plant order \( n \) and controller order \( \delta \) formulated in pole placement for unity feedback loop technique, proportional controller would be realized for the SISO subsystems (i.e., \( \delta = n - 1 \)). In order to realize PI control for both the SISO subsystems, controller order \( \delta \) should be increased by 1. Transfer functions of PI controllers dedicated for both the SISO subsystems are specified in the following way,

\[ C_{ms_i}(s) = \frac{k_{pr_{ms_i}} \cdot s + k_{in_{ms_i}}}{s} \]  

(5.20)

- \( k_{pr_{ms_i}} \) is the proportional gain of \( C_{ms_i}(s) \)
- \( k_{in_{ms_i}} \) is the integral gain of \( C_{ms_i}(s) \)

\[ C_{ps_i}(s) = \frac{k_{pr_{ps_i}} \cdot s + k_{in_{ps_i}}}{s} \]  

(5.21)

- \( k_{pr_{ps_i}} \) is the proportional gain of \( C_{ps_i}(s) \)
- \( k_{in_{ps_i}} \) is the integral gain of \( C_{ps_i}(s) \)

Depending upon the order of both the SISO subsystems and the orders of the respective opted PI controllers, overall transfer functions of the SISO subsystems were chosen in the following manner [25].

\[ T_{ms_i}(s) = \frac{\omega_{n_1}^2}{s^2 + 2\zeta_1 \omega_{n_1} s + \omega_{n_1}^2} \]  

(5.22)

- \( \zeta_1 \) is the damping factor of \( T_{ms_i}(s) \), \( 0 < \zeta_1 < 1 \)
- \( \omega_{n_1} \) is the natural frequency of \( T_{ms_i}(s) \), \( \omega_{n_1} > 0 \)

\[ T_{ps_i}(s) = \frac{\omega_{n_{2i}}^2}{s^2 + 2\zeta_2 \omega_{n_{2i}} s + \omega_{n_{2i}}^2} \]  

(5.23)

- \( \zeta_2 \) is the damping factor of \( T_{ps_i}(s) \), \( 0 < \zeta_2 < 1 \)
- \( \omega_{n_{2i}} \) is the natural frequency of \( T_{ps_i}(s) \), \( \omega_{n_{2i}} > 0 \)
Numerical determination of the overall transfer functions $T_{ms_i}(s)$ and $T_{ps_i}(s)$ is based on the selection of the following rise times and damping factors.

\[ t_{r1i} = \Lambda_{1i} t_{RC1i} \]  \hspace{1cm} (5.24)

- $t_{RC1i}$ is the time constant of MS SISO subsystem, $t_{RC1i} = \frac{1}{k_{1i}}$
- $\Lambda_{1i} = 0.6$

\[ t_{r2i} = \Lambda_{2i} t_{RC2i} \]  \hspace{1cm} (5.25)

- $t_{RC2i}$ is the time constant of PS SISO subsystem, $t_{RC2i} = \frac{1}{k_{4i}}$
- $\Lambda_{2i} = 0.55$

\[ \zeta_1 = \zeta_2 = 0.995 \]  \hspace{1cm} (5.26)

$t_{r1i}$ is the rise time of the overall MS fluid subsystem i.e., actual system $P_{ms_i}(s)$ with its designated PI controller $C_{ms_i}(s)$ in closed loop form. $t_{r2i}$ is the rise time of the overall PS fluid subsystem i.e., actual system $P_{ps_i}(s)$ with its designated PI controller $C_{ps_i}(s)$ in closed loop form. Based on the SISO transfer functions $P_{ms_i}(s)$ and $P_{ps_i}(s)$, controller transfer functions $C_{ms_i}(s)$ and $C_{ps_i}(s)$ and overall transfer functions determined from Equation 5.24, Equation 5.25 and Equation 5.26, the PI control parameters of MML system were found out by applying the pole placement for unity feedback loop technique under the three scenarios [4]. PI control parameters of MML system are shown in Table 5.1, Table 5.2 and Table 5.3.

<table>
<thead>
<tr>
<th>PI Parameters(scenario 1)</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_{pr_{ms_1}}$</td>
<td>20.16</td>
</tr>
<tr>
<td>$k_{in_{ms_1}}$</td>
<td>80.00</td>
</tr>
<tr>
<td>$k_{pr_{ps_1}}$</td>
<td>17.90</td>
</tr>
<tr>
<td>$k_{in_{ps_1}}$</td>
<td>39.71</td>
</tr>
</tbody>
</table>

Table 5.1: PI parameters of MML system under first scenario

<table>
<thead>
<tr>
<th>PI Parameters(scenario 2)</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_{pr_{ms_2}}$</td>
<td>20.16</td>
</tr>
<tr>
<td>$k_{in_{ms_2}}$</td>
<td>80.00</td>
</tr>
<tr>
<td>$k_{pr_{ps_2}}$</td>
<td>18.18</td>
</tr>
<tr>
<td>$k_{in_{ps_2}}$</td>
<td>43.46</td>
</tr>
</tbody>
</table>

Table 5.2: PI parameters of MML system under second scenario

To have a robust PI parameter setting, mean value of the PI parameters which were found under the three scenarios needs to be calculated. Also, a standard PI parameter
CHAPTER 5. MODEL BASED CONTROLLER DESIGN

Table 5.3: PI parameters of MML system under third scenario

<table>
<thead>
<tr>
<th>PI Parameters(Scenario 3)</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_{pr_{m3}}$</td>
<td>20.16</td>
</tr>
<tr>
<td>$k_{in_{m3}}$</td>
<td>80.00</td>
</tr>
<tr>
<td>$k_{pr_{p3}}$</td>
<td>17.86</td>
</tr>
<tr>
<td>$k_{in_{p3}}$</td>
<td>45.74</td>
</tr>
</tbody>
</table>

The setting was founded earlier through trial and error method by performing several blending experiments on MML system with the PI controllers. This particular PI parameter setting happened to work quite good with respect to the three scenarios. The standard PI parameter setting and the computed robust PI parameter setting is shown in Table 5.4.

<table>
<thead>
<tr>
<th>PI Parameters</th>
<th>Standard Setting</th>
<th>Computed Robust Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_{pr_{m}}$</td>
<td>40.00</td>
<td>20.16</td>
</tr>
<tr>
<td>$k_{in_{m}}$</td>
<td>50.00</td>
<td>80.00</td>
</tr>
<tr>
<td>$k_{pr_{p}}$</td>
<td>50.00</td>
<td>17.92</td>
</tr>
<tr>
<td>$k_{in_{p}}$</td>
<td>20.00</td>
<td>42.00</td>
</tr>
</tbody>
</table>

Table 5.4: PI parameter settings of MML system

5.4.2 Experimental and Simulation Results

MS and PS fluids have to be mixed at a certain pre-defined mixing ratio. According to the three scenarios, different mixing ratios were defined by which the reference MS and PS fluid flows were assigned. The PI parameter settings specified in Table 5.4 are used for blending of MS and PS fluids inside the MML system by assigning different mixing ratios to MML system. In each and every scenario, comparison was made between the practical and simulation results which were obtained from the real world plant and its realized mathematical model in MATLAB/Simulink respectively. Also, the mixing ratios which were used for the blending of MS and PS fluids of MML system is shown in Table 5.5.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Mixing Ratio(assigned)</th>
<th>PS Orifice Size(mm)</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2:1, 10:1, 20:1</td>
<td>3</td>
<td>2:1 - 10:1</td>
</tr>
<tr>
<td>2</td>
<td>20:1, 80:1, 120:1</td>
<td>1</td>
<td>20:1 - 120:1</td>
</tr>
<tr>
<td>3</td>
<td>120:1, 600:1, 1000:1</td>
<td>0.45</td>
<td>120:1 - 1000:1</td>
</tr>
</tbody>
</table>

Table 5.5: Mixing ratios of MML system in different scenarios
The flow control of MS and PS fluids with respect to first scenario can be seen in Figure 5.6, Figure 5.7 and Figure 5.8.

Figure 5.6: Blending test with 2:1 mixing ratio in scenario 1
Mainstream Flow Tracking

Partstream Flow Tracking

Mainstream Pump Speed

Partstream Pump Speed

Figure 5.7: Blending test with 10:1 mixing ratio in scenario 1
Figure 5.8: Blending test with 20:1 mixing ratio in scenario 1
From the Figure 5.6, Figure 5.7 and Figure 5.8 the curves denoted by red and blue represent the measured MS and PS fluid flows with respect to its measured MS and PS pump speeds when the real world plant is operated with the standard PI parameter setting and computed PI parameter setting respectively. And the curves denoted by green and magenta represent the simulated MS and PS fluid flows with respect to its simulated MS and PS pump speeds when the realized simulink model is operated with the standard PI parameter setting and computed PI parameter setting respectively.

The MS and PS fluid flows track their respective assigned reference flows quite good. With respect to computed PI parameter setting, there is considerable improvement in the settling time and rise time of PS fluid flows than it was with the standard PI parameter setting. PS pump tries to run at an optimum speed which is evident from the simulation and real world plant. Also, the settling time of MS fluid flows obtained with computed PI parameter setting is comparatively less than the settling time of MS fluid flows obtained with standard PI parameter setting. The MS pump tries to run at an optimum speed after a certain time delay.

The reason behind the occurrence of time delay during the operation of MS pump is not to have air bubbles in the MS fluid path. In general, MS fluid has to travel quite an amount of distance from the pneumatic valve 'VMSC' to the point where MS pump is located. If the MS pump is running unnecessarily, then there would be a possible occurrence of air bubbles which are undesired for MML system. The mathematical model of MML system which is simulated with PI controllers doesn’t come across this particular time delay element. Due to the absence of this external time delay in the realized simulink model, the MS pump in simulation tries to start at a pre-determined amount of speed. During the blending test of MS and PS fluids, occurrence of time delay in the measured MS pump speed is acceptable since the MS fluid has to rise uphill in order to reach the inlet of the MS pump.
The flow control of MS and PS fluids with respect to second scenario can be seen in Figure 5.9 Figure 5.10 and Figure 5.11.

Figure 5.9: Blending test with 20:1 mixing ratio in scenario 2

From the Figure 5.9 Figure 5.10 and Figure 5.11 the MS and PS fluid flows track the reference fluid flows which were assigned during the blending test of MML system with respect to the mixing ratios used in scenario 2.
Figure 5.10: Blending test with 80:1 mixing ratio in scenario 2
The settling time and rise time of PS fluid flows obtained from the computed PI parameter setting is an improved version compared to that of the PS fluid flows which were obtained from the standard PI parameter setting. Also, the settling time of MS fluid flows incurred from the computed PI parameter setting is comparatively less than that of the settling time of MS fluid flows obtained with the standard PI parameter setting. Since there is this uncertain nature of PS parameter $k_6$ which was assumed to be time independent, the MS and PS pump speeds in simulation never seem to be a close match with the reality. The small spikes present in the measured PS fluid flows is due to the high frequent noise caused by the operation of the MS and PS pumps.

Figure 5.11: Blending test with 120:1 mixing ratio in scenario 2
The flow control of MS and PS fluids with respect to third scenario can be seen in Figure 5.12, Figure 5.13 and Figure 5.14.

From the Figure 5.12, Figure 5.13 and Figure 5.14, the MS and PS fluid flows seem to track their assigned reference flows with respect to the mixing ratios used in this scenario.
Figure 5.13: Blending test with 600:1 mixing ratio in scenario 3
Even in scenario 3, there is substantial improvement of the settling time and rise time of PS fluid flows obtained from computed PI parameter setting than that of the PS fluid flows obtained with standard PI parameter setting. The settling time of MS fluid flow obtained from the computed PI parameter setting is relatively less than the settling time of MS fluid flow obtained with the standard PI parameter setting. The small spikes which were earlier present in scenario 2 with the PS fluid flows are evident in this scenario as well. High frequent noise is associated with the PS fluid flows which results into these spikes.

Figure 5.14: Blending test with 1000:1 mixing ratio in scenario 3
5.5 MIMO Decoupling

In Section 5.4 PI control of a typical MIMO system was carried out by eliminating the cross-coupling between its outputs. In principle, omission of the substantial cross-coupling between the system outputs is not guaranteed. In those cases, decoupled PI control of MIMO systems needs to be performed. Considering a 2×2 order MIMO linear system which is included with two inputs and two outputs. With the help of Figure 5.15, preferred MIMO system can be represented in the form of transfer function notation.

\[
\begin{bmatrix}
X_1(s) \\
X_2(s)
\end{bmatrix} = \begin{bmatrix}
G_{11}(s) & G_{12}(s) \\
G_{21}(s) & G_{22}(s)
\end{bmatrix} \cdot \begin{bmatrix}
u_1(s) \\
u_2(s)
\end{bmatrix}
\]  

(5.27)

The decoupling of the above MIMO system is performed by introducing two new input variables \(v_1\) and \(v_2\) in the following way [21],

\[
u_1(s) = \frac{C_1(s)}{G_{11}(s)} \cdot v_1(s) - d_1(s) \cdot u_2(s)
\]

(5.28)

\[
u_2(s) = \frac{C_2(s)}{G_{22}(s)} \cdot v_2(s) - d_2(s) \cdot u_1(s)
\]

(5.29)

- Where \(C_1(s)\) and \(C_2(s)\) are single-input single-output (SISO) transfer functions
- \(d_1(s) = \frac{G_{12}(s)}{G_{11}(s)}\) is decoupler 1 and \(d_2(s) = \frac{G_{21}(s)}{G_{22}(s)}\) is decoupler 2
If Equation 5.28 and Equation 5.29 hold, then the mutual interaction between $X_1$ and $X_2$ is not exhibited. MIMO system with the decouplers is shown in Figure 5.16.

![Decoupling of MIMO system](image)

Figure 5.16: Decoupling of MIMO system

MIMO system in combination with the introduced decouplers can be replaced by the SISO transfer functions $C_1(s)$ and $C_2(s)$. This particular situation can be seen from Figure 5.17 where the decoupled MIMO system is specified with its respective PI controllers. Assuming the decoupled MIMO system is not operated with its respective PI controllers, $C_1(s)$ and $C_2(s)$ can be found with the decouplers $d_1$ and $d_2$ where the newly introduced input variables $v_1$ and $v_2$ becomes inactive [18].

\[
C_1(s) = G_{11}(s) - G_{12}(s) \frac{G_{21}(s)}{G_{22}(s)} \tag{5.30}
\]

\[
C_2(s) = G_{22}(s) - G_{21}(s) \frac{G_{12}(s)}{G_{11}(s)} \tag{5.31}
\]

The resultant $C_1(s)$ and $C_2(s)$ guarantees the BIBO stability of the inner loops of the MIMO system in combination with the decouplers. The dynamics of the decoupled MIMO system are simply replaced by the SISO transfer functions $C_1(s)$ and $C_2(s)$. Therefore, the system outputs $X_1$ and $X_2$ can be represented with the SISO transfer functions $C_1(s)$ and $C_2(s)$ and the newly introduced input variables $v_1$ and $v_2$.

\[
X_1(s) = C_1(s) \cdot v_1(s) \tag{5.32}
\]

\[
X_2(s) = C_2(s) \cdot v_2(s) \tag{5.33}
\]
Now the SISO transfer functions $C_1(s)$ and $C_2(s)$ represents two independent sub-systems. The PI controllers will be designed for these two subsystems. Pole placement for unity feedback loop technique discussed in Section 5.3 is adopted for determining the corresponding PI control parameters.

### 5.5.1 PI Control of Decoupled MML System

The mathematical model of MML system described by the Equation 5.34 and Equation 5.35 is revisited again from Chapter 3:

$$
\dot{Q}_{msi} = -k_1 Q_{msi} + k_2 W_{ms} + k_0 P - k_3 i Q_{psi}
$$

$$
\dot{Q}_{psi} = -k_{4i} Q_{psi} + f_{psi}(W_{ps}, Q_{msi})
$$

- $f_{psi}(W_{ps}, Q_{msi}) = \begin{cases} 
  k_{5i} W_{ps} - k_{6i} Q_{msi} & \text{if } W_{ps} + \alpha_i Q_{msi} \geq \beta_i \\
  0 & \text{Otherwise}
\end{cases}$

- $i = 1, 2, 3$ represents the MS and PS fluid flow scenarios with respect to the change of PS orifice size
To find the decoupled PI control parameters of MML system, mathematical model of MML system has to undergo certain changes. MS pump speed $W_{ms}$ and constant pressure differential $P$ was replaced by a new input variable $u_1$. The existence of PS fluid flow with respect to the PS pump speed and MS fluid flow was assumed. Furthermore, the mathematical model of MML system was simplified under the following conditions:

- Linear approximation of the nonlinear function $f_{ps_i}(W_{ps}, Q_{ms_i})$ was carried out
- Coupling associated with MS and PS fluid flows is not neglected

Mathematical model of MML system was strictly transformed into a multi-input multi-output (MIMO) system which is given from the following matrix notation,

$$
\begin{bmatrix}
\dot{Q}_{ms_i} \\
\dot{Q}_{ps_i}
\end{bmatrix} = 
\begin{bmatrix}
-k_1 & -k_{3i} & 1 & 0 \\
-k_{6i} & -k_{4i} & 0 & k_{5i}
\end{bmatrix} \cdot 
\begin{bmatrix}
Q_{ms_i} \\
Q_{ps_i}
\end{bmatrix} + 
\begin{bmatrix}
u_1 \\
u_2
\end{bmatrix}
$$

(5.36)

- Where $u_1 = k_2.W_{ms} + k_0.P$, $u_2 = W_{ps}$

Equation (5.36) represents a $2 \times 2$ order MIMO linear system. For simplicity, the outputs $Q_{ms_i}$ and $Q_{ps_i}$ can be specified in terms of its inputs $W_{ms}$ and $W_{ps}$ and the transfer functions realized from each input to each output irrespective of the scenarios in which MML system is operated.

$$
\begin{bmatrix}
X_1(s) \\
X_2(s)
\end{bmatrix} = 
\begin{bmatrix}
G_{11}(s) & G_{12}(s) \\
G_{21}(s) & G_{22}(s)
\end{bmatrix} \cdot 
\begin{bmatrix}
u_1(s) \\
u_2(s)
\end{bmatrix}
$$

(5.37)

- $X_1 = Q_{ms}$, $X_2 = Q_{ps}$

The decoupling of the MIMO system is performed in the exact same way as it was discussed in Section 5.5

$$
u_1(s) = \frac{P_1(s)}{G_{11}(s)}.v_1(s) - d_1(s).u_2(s)
$$

(5.38)

$$
u_2(s) = \frac{P_2(s)}{G_{22}(s)}.v_2(s) - d_2(s).u_1(s)
$$

(5.39)

- $d_1$ and $d_2$ are decouplers, $d_1(s) = \frac{G_{12}(s)}{G_{11}(s)}$, $d_2(s) = \frac{G_{21}(s)}{G_{22}(s)}$
- $v_1$ and $v_2$ are newly introduced input variables
- $P_1(s)$ and $P_2(s)$ are SISO transfer functions

The SISO transfer functions $P_1(s)$ and $P_2(s)$ are computed with the help of introduced decouplers $d_1$ and $d_2$ in the following manner:

$$
P_1(s) = G_{11}(s) - G_{12}(s).d_1(s)
$$

(5.40)

$$
P_2(s) = G_{22}(s) - G_{21}(s).d_2(s)
$$

(5.41)
Therefore, the outputs of the MIMO system $X_1$ and $X_2$ can be represented in terms of the new input variables $v_1$ and $v_2$ and the SISO transfer functions $P_1(s)$ and $P_2(s)$.

\[ X_1(s) = P_1(s).v_1(s) \]  
\[ X_2(s) = P_2(s).v_2(s) \]

The SISO transfer functions $P_1(s)$ and $P_2(s)$ are regarded for the PI control design. Pole placement technique for unity feedback loop discussed in Section 5.3 is adopted for the computation of parameters of its respective decoupled PI controllers. The pole placement for unity feedback loop technique primarily depends on the realization of the overall transfer functions. The overall transfer functions are given by the following Equation 5.44 and Equation 5.45,

\[ T_1(s) = \frac{\omega_{n_1}^2}{s^2 + 2\zeta_1 \omega_{n_1} s + \omega_{n_1}^2} \]  
\[ T_2(s) = \frac{\omega_{n_2}^2}{s^2 + 2\zeta_2 \omega_{n_2} s + \omega_{n_2}^2} \]

- $\zeta_1$ is the damping factor of $T_1(s)$, $0 < \zeta_1 < 1$
- $\omega_{n_1}$ is the natural frequency of $T_1(s)$, $\omega_{n_1} > 0$

Using Equation 5.14 and the graphical solutions obtained from Figure 5.4 and Figure 5.5, natural frequencies $\omega_{n_1}$ and $\omega_{n_2}$ were computed. Numerical determination of the overall transfer functions $T_1(s)$ and $T_2(s)$ is based on the selection of its rise times and damping factors which are specified in the following way,

\[ t_{r_1} = \Lambda_1.t_{RC_1} \]  
\[ t_{r_2} = \Lambda_2.t_{RC_2} \]

- $\Lambda_1 = 0.4$
- $t_{RC_1}$ is the time constant of $P_1(s)$, $t_{RC_1} = \frac{1}{k_1}$

\[ t_{r_2} = \Lambda_2.t_{RC_2} \]

- $\Lambda_2 = 0.8$
- $t_{RC_2}$ is the time constant of $P_2(s)$, $t_{RC_2} = \frac{1}{k_4}$
\( \zeta_1 = \zeta_2 = 0.995 \)  

(5.48)

Based on the SISO transfer functions \( P_1(s) \) and \( P_2(s) \) and the realized overall transfer functions \( T_1(s) \) and \( T_2(s) \), the PI parameters were computed using the pole placement for unity feedback loop technique. The PI parameters resulted from the decoupling of the realized simulink model of MML system and pole placement for unity feedback loop technique are shown in Table 5.6, Table 5.7 and Table 5.8.

<table>
<thead>
<tr>
<th>PI Parameters(scenario 1)</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k_{pr_{ms1}} )</td>
<td>16.0274</td>
</tr>
<tr>
<td>( k_{in_{ms1}} )</td>
<td>93.3766</td>
</tr>
<tr>
<td>( k_{pr_{ps1}} )</td>
<td>10.8629</td>
</tr>
<tr>
<td>( k_{in_{ps1}} )</td>
<td>15.8964</td>
</tr>
</tbody>
</table>

Table 5.6: PI parameters of the decoupled MML system under first scenario

<table>
<thead>
<tr>
<th>PI Parameters(Scenario 2)</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k_{pr_{ms2}} )</td>
<td>16.0274</td>
</tr>
<tr>
<td>( k_{in_{ms2}} )</td>
<td>93.3766</td>
</tr>
<tr>
<td>( k_{pr_{ps2}} )</td>
<td>10.7713</td>
</tr>
<tr>
<td>( k_{in_{ps2}} )</td>
<td>16.9800</td>
</tr>
</tbody>
</table>

Table 5.7: PI parameters of decoupled MML system under second scenario

<table>
<thead>
<tr>
<th>PI Parameters(Scenario 3)</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k_{pr_{ms3}} )</td>
<td>16.0274</td>
</tr>
<tr>
<td>( k_{in_{ms3}} )</td>
<td>93.3766</td>
</tr>
<tr>
<td>( k_{pr_{ps3}} )</td>
<td>10.5790</td>
</tr>
<tr>
<td>( k_{in_{ps3}} )</td>
<td>17.8680</td>
</tr>
</tbody>
</table>

Table 5.8: PI parameters of decoupled MML system under third scenario
5.5.1.1 Simulation Results

The PI parameters specified in Table 5.6, Table 5.7 and Table 5.8 are applied on the realized simulink model which includes the decoupling of MML system. The blending of MS and PS fluids is examined with respect to the three scenarios. The blending of MS and PS fluids is shown in Figure 5.18, Figure 5.19 and Figure 5.20.

Figure 5.18: Blending of MS and PS with respect to PI control of decoupled MML system in scenario 1
Figure 5.19: Blending of MS and PS with respect to PI control of decoupled MML system in scenario 2
From the Figure 5.18, Figure 5.19 and Figure 5.20 the MS and PS fluid flows obtained in simulation track their respective reference flows quite good. Even though there is slight amount of overshoot in PS fluid flow, there is no considerable effect on the MS fluid flow. The overshoot resulted in PS fluid flow would be more if the PS fluid flow needs to have much faster response. Therefore, a kind of trade off between the overshoot and choice of the rise time was performed. Based on the earlier results yielded from the PI control of MML system (without decoupling) the settling time and rise time for both MS and PS fluid flows is acceptable.

Figure 5.20: Blending of MS and PS with respect to PI control of decoupled MML system in scenario 3
Chapter 6

Conclusion

The dynamics of MML system were studied in practice and later realized it as a MIMO nonlinear system. Based on the measured data of MML system, its mathematical model was derived under different scenarios (i.e., change in PS orifice size) where distilled water \((H_2O)\) is used as a fluid medium for both fluid streams. Using least squares and optimization methods, mathematical model was identified with all its parameters. The behavioral match between the measured data of MML system and its identified mathematical model was satisfactory.

Pole placement technique was adopted and in turn a robust PI parameter setting for MML system was found. The response of MS and PS fluids were examined with respect to the robust and standard PI parameter settings. According to these PI parameter settings, the MS and PS fluids blends appropriately with respect to its assigned mixing ratios in the three scenarios in which MML system is operated. PI control of decoupled MML system was performed in simulation. A set of PI parameter settings were computed based on the PI control design of decoupled MML system. With these particular PI parameter settings, proper blending of MS and PS fluids was achieved. During the PI control of decoupled MML system, there was no cross coupling effect between the MS and PS fluid flows.

In practice, PI control of decoupled MML system can be a good application. This particular control design where decoupling of a MIMO system is included would lead to further study of the overall dynamics (i.e., Decoupled MML system+PI controllers in closed loop form). The modelling of MML system can be further improved by introducing the MS and PS flow parameters \(k_{3i}\) and \(k_{6i}\) as time varying parameters instead of fixed constants. Since distilled water was used as the fluid mediums for both MS and PS, it would create a little curiosity to examine whether the robust PI parameter setting computed from pole placement technique would work substantially for obtaining heterogeneous fluid mixtures as well.
Chapter 7

Appendix

7.1 Appendix A

7.1.1 Leviflow Flowmeter

The standard configuration of the Leviflow TM flowmeter shown in Figure 7.1 consists of a flow sensor and a converter with a digital signal processor (DSP) for processing the sensor signals. Various signals (analog, digital input and digital output) are provided and configured with PC software. A two wire RS485 bus allows arrays of multiple flowmeters. In addition, the sensor value is shown on a 4-digit display.

![Figure 7.1: Standard single channel flowmeter configuration](http://www.levitronix.com/Product_Brochures_and_Manuals/Brochure_LEVIFLOW_english_Rev04.html)

7.1.2 Centrifugal Pump

The basic configuration used for the centrifugal pumps is the extended version. The extended version shown in Figure 7.2 consists of a controller with integrated PLC interface.
CHAPTER 7. APPENDIX

Figure 7.2: Extended operation (flow control) with extended controller

This allows setting the speed by an external signal and enables precise flow control in connection with their flow sensor. The calibration and appropriate functionality of the centrifugal pumps are carried out with the help of levitronix service software.

7.1.3 Air Compressor

Air compressor is defined as a device which takes air at the atmospheric pressure and delivers it at a higher pressure. In order to provide the compressed air to the pneumatic valves, reciprocating piston type air compressor is used to serve this purpose. Reciprocating air compressors are positive displacement machines, which increase the pressure of the air by reducing its volume. This means they are taking in successive volumes of air which is confined within a closed space and elevating this air to a higher pressure.

The reciprocating air compressor accomplishes this by a piston within a cylinder as the compressing and displacing element. A reciprocating piston type air compressor consists of a crankshaft, a connecting rod and piston, a cylinder and a valve head. The crankshaft is driven by an electric motor. A typical reciprocating piston type air compressor is shown in Figure 7.3.

From the Figure 7.3 as the piston moves down, a vacuum is created above it. This allows outside air at atmospheric pressure to push open the inlet valve and fill the area above the piston. As the piston moves up, the air above it compresses, holds the inlet valve shut and pushes the discharge valve open. The air moves from the discharge port to the tank. With each stroke, more air enters the tank and the pressure rises. Compressors use a pressure switch to stop the motor when tank pressure reaches a preset limit about 8.6 bar. Most of the time, there won’t be that much pressure needed. Therefore, the air line will include a regulator that provides to set the pressure requirements of the tool which are being used. A gauge before the regulator monitors tank pressure and a gauge after the regulator monitors air-line pressure. In addition, the tank has a safety valve that
Figure 7.3: Air compressor

Source: http://www.popularmechanics.com/home/improvement/energy-efficient/1275131
Figure 7.4: Schematic diagram of four port solenoid valve

opens if the pressure switch malfunctions. The pressure switch may also incorporate an unloaded valve that reduces tank pressure when the compressor is turned off.

7.1.4 Solenoid Valve

Solenoid valve is defined as the electro-mechanical device which also regulates, directs or controls the flow of fluid (here compressed air). It is used to distribute the compressed air in order to close the pneumatic valves. SJ 3000 series four port solenoid valve with manifolds is used in conjunction with the pneumatic valves and air compressor. Schematic diagram of a typical four port solenoid valve is shown in Figure 7.4.

Figure 7.4 consists of four ports out of which one port goes to the inlet port of pneumatic valve, another port is assigned to the exhaust port of pneumatic valve, one coming from the air compressor and the remaining one going to exhaust. The solenoid valve consists of two parts; one is the solenoid and the other one is the spool valve. Solenoids are coils of wire with iron core. When one of the coils is energized, the iron core moves causing the valve to open or close and compressed air is directed to a different port. Multiple pneumatic valves are linked to a single solenoid manifold base. Both normally open and normally closed valves can be mounted simultaneously, making them extremely flexible in application. The basic configuration of pneumatic valves with the air compressor and the single solenoid manifold base is shown in Figure 7.5.

From the Figure 7.5 single solenoid manifold base is configured with the WAGO 750-337 CANopen field bus coupler module. The fieldbus coupler contains the fieldbus interface, electronics and power supply terminal. The electronics process the data of the bus modules and make it available for fieldbus communication. This fieldbus coupler acts as an interfacing module in between the solenoid valve and PLC to perform its required func-
tionality. With the help of air compressor, solenoid valve would provide the compressed air to the desired pneumatic valves for setting up the complete MML system before the start of its operation.

## 7.2 Appendix B

### 7.2.1 Simulink Design Optimization

Simulink Design Optimization software estimates model parameters by comparing the transient data with simulation data generated from the Simulink model. Simulink design optimization provides a graphical user interface (GUI) called control and estimation tools manager. The control and estimation tools manager is shown in Figure 7.6 and Figure 7.7. Using Figure 7.6 the input and output data of the system is specified in the transient data part of control and estimation tools manager. The parameters which are about to be estimated has to be specified in the variables part of control and estimation tools manager. The initial value and the minimum and maximum range for each parameter need to be defined in the variables part. Depending upon the realized simulink model, an optimization method has to be chosen. Using optimization techniques, the software estimates the parameters and initial conditions of states to minimize a user-selected cost function. The cost function typically calculates a least-square error between the measured and simulated data.
Figure 7.6: Control and estimation tools manager with transient data part
Figure 7.7: Control and estimation tools manager with variables part and optimization options
7.3 Appendix C

7.3.1 Gradient Descent Method

In fitting a function \( \hat{y}(t, \lambda) \) of an independent variable \( t \) and a vector of \( n \) parameters \( \lambda \) to a set of data points \((t_i, y_i)\), it is customary and convenient to minimize the sum of the weighted squares of the errors (or weighted residuals) between the measured data \( y(t_i) \) and the curve-fit function \( y(t_i, \lambda) \). This scalar-valued goodness-of-fit measure is called the chi-squared error criterion.

\[
\chi^2(\lambda) = \frac{1}{2} \sum_{i=1}^{m} \left[ \frac{y(t_i) - \hat{y}(t_i, \lambda)}{w_i} \right]^2
\]  

\[
\chi^2(\lambda) = \frac{1}{2} (y - \hat{y}(\lambda))^T W (y - \hat{y}(\lambda))
\]  

\[
\chi^2(\lambda) = \frac{1}{2} y^T W y - y^T W \hat{y} + \frac{1}{2} \hat{y}^T W \hat{y}
\]  

The value \( w_i \) is a measure of the error in measurement \( y(t_i) \). The weighting matrix \( W \) is diagonal with \( W_{ii} = 1/w_i^2 \). If the function \( \hat{y} \) is nonlinear in the model parameters \( \lambda \), then the minimization of \( \chi^2 \) with respect to the parameters must be carried out iteratively. The goal of each iteration is to find a perturbation \( h \) to the parameters \( \lambda \) that reduces \( \chi^2 \).

The steepest descent method is a general minimization method which updates parameter values in the direction opposite to the gradient of the objective function. It is recognized as a highly convergent algorithm for finding the minimum of simple objective functions \([13, 15]\). For problems with thousands of parameters, gradient descent methods may be the only viable method. The gradient of the chi-squared objective function with respect to the parameters is,

\[
\frac{\partial \chi^2}{\partial \lambda} = (y - \hat{y}(\lambda))^T W \frac{\partial}{\partial \lambda} (y - \hat{y}(\lambda))
\]  

\[
\frac{\partial \chi^2}{\partial \lambda} = -(y - \hat{y}(\lambda))^T W \frac{\partial \hat{y}(\lambda)}{\partial \lambda}
\]  

\[
\frac{\partial \chi^2}{\partial \lambda} = -(y - \hat{y}(\lambda))^T W J
\]  

\[
J = \frac{\partial \hat{y}(\lambda)}{\partial \lambda}
\]

Where the \( m \times n \) Jacobian matrix \( J \) represents the local sensitivity of the function \( \hat{y} \) to variation in the parameters \( \lambda \). The perturbation \( h \) that moves the parameters in the direction of steepest descent is given by,

\[
h_{gd} = -\alpha J^T W (y - \hat{y}(\lambda))^T
\]  

Where the positive scalar \( \alpha \) determines the length of the step in the steepest descent direction.
7.3.2 Gauss-Newton Method

The Gauss-Newton method is a method of minimizing a sum of squares objective function. It presumes that the objective function is approximately quadratic in the parameters near the optimal solution. For more moderately-sized problems the Gauss-Newton method typically converges much faster than gradient-descent methods [15]. The function evaluated with perturbed model parameters may be locally approximated through a first-order Taylor series expansion.

\[ \hat{y}(\lambda + h) \approx \hat{y}(\lambda) + \left[ \frac{\partial \hat{y}(\lambda)}{\partial \lambda} \right] h = \hat{y} + Jh, \quad (7.8) \]

Substituting the approximation for the perturbed function \( y + Jh \) for \( \hat{y} \) in Equation 7.3,

\[ \chi^2(\lambda + h) \approx \frac{1}{2} y^T Wy + \frac{1}{2} \hat{y}^T W \hat{y} - \frac{1}{2} y^T W \hat{y} - (y - \hat{y})^T Jh + \frac{1}{2} h^T J^T W Jh \quad (7.9) \]

This shows that \( \chi^2 \) is approximately quadratic in the perturbation \( h \), and that the Hessian of the chi-squared fit criterion is approximately \( J^T W J \). The perturbation \( h \) that minimizes \( \chi^2 \) is found from \( \partial \chi^2 / \partial \lambda = 0 \).

\[ \frac{\partial \chi^2(\lambda + h)}{\partial h} \approx (y - \hat{y})^T W J + h^T J^T W J, \quad (7.10) \]

And the resulting normal equations for the Gauss-Newton perturbation are,

\[ [J^T W J]h_{gn} = J^T W (y - \hat{y}) \quad (7.11) \]

7.4 Appendix D

7.4.1 Forward and Backward Differentiation

The conversion from continuous-time systems to discrete-time systems are performed through forward and backward differentiation methods. Figure 7.8 illustrates both the forward and backward differentiation methods. The forward differentiation method can be seen as the following approximation of the time derivative of a time-valued function which here is denoted by \( x \).

\[ \dot{x}(t_k) \approx \frac{x(t_{k+1}) - x(t_k)}{T_s} \quad (7.12) \]

\( T_s \) is the time step, i.e., the time interval between two subsequent points of time. The name ”Forward differentiation method” comes from the \( x(t_{k+1}) \) term in Equation 7.12.

The backward differentiation method is based on the following approximation of the time derivative:

\[ \dot{x}(t_k) \approx \frac{x(t_k) - x(t_{k-1})}{T_s} \quad (7.13) \]

The name ”Backward differentiation method” comes from the \( x(t_{k-1}) \) term in Equation 7.13.
The forward and backward differentiation methods can be carried out directly on transfer function if one translates the Equation 7.12 and Equation 7.13 into frequency domain. In the translation, each discrete-time left-shift by \( n \) corresponds to a \( z^n \) multiplying factor in \( z \)-domain, and each \( \frac{d^n}{dt^n} \) in continuous-time domain corresponds to \( s^n \) multiplying factor in Laplace domain. Thus we have:

\[
s \rightarrow \frac{z - 1}{T_s} \Rightarrow \text{(forward rectangular rule)} \quad (7.14)
\]

\[
s \rightarrow \frac{z - 1}{T_s z} \Rightarrow \text{(backward rectangular rule)} \quad (7.15)
\]
Bibliography


