Development of Control Lab Interface for Data Acquisition using Lab VIEW

VIVEK SHARMA

Masters' Degree Project
Stockholm, Sweden Feb 2007

XR-EE-RT 2007:006
Abstract

A lab named ‘System Identification Lab’ is a compulsory part of the course ‘Modeling of Dynamical System’ given by School of Electrical Engineering, Automatic Control at KTH. System identification is an experimental method to derive a mathematical model from the input and the output data. The apparatus used in the lab is a fan and a hinged rectangular plate. Both fan and plate are mounted on a slide. The idea of this lab is to collect the input and output data from this process. The input data is the DC voltage sent to the fan and the output is the angular displacement of the plate due to air stream from the fan. This data is then used to derive a linear model by applying different theoretical methods.

The main focus of this thesis work has been to design a user interface for this lab, and implement it in LabVIEW, which is an easy to use, integrated graphical environment with built-in compatibility across a broad range of data acquisition and control hardware devices. In short, the interface first lets the user choose the sampling time and then one can choose between different input signals. The input and output signals are displayed as plots on the screen and can also be saved to a file. A second, similar interface has also been implemented, where the process is replaced by a simulation model. The simulation model is based on an identified linear model with some added disturbances and non-linear effects. The idea with the simulation model is that the ‘System Identification Lab’ then can be done without using the lab process.

This report also includes an introduction to system identification and a discussion about how to choose appropriate input signals for an identification experiment. These methods are used to derive the simulation model and in order to understand the lab process better, some step responses are done and the process is also modeled from physical principles.
Acknowledgements

This thesis work has been carried out at the lab of Automatic control, School of Electrical Engineering, KTH. I would like to thank the whole Automatic Control group for their help. Many thanks to others those have contributed to this work in many different ways and helped me to finish this project.

I would like to thank specially my supervisor, Jonas Mårtensson for his guidance, advices, tremendous help, time and patience.

I am grateful to my examiner Prof. Bo Wahlberg for giving me an opportunity to carry on this work.

At last, certainly not least, I would like to thank Mummy and Daddy for their support, encouragement and never-ending love.

Vivek Sharma
Stockholm, January 2007
Development of control lab interface for data acquisition using LabVIEW

Content

1 Introduction .............................................................................................................1

2 Introduction to the system .........................................................................................2
  2.1 The Hardware .......................................................................................................2
  2.2 The Software ........................................................................................................3
    2.2.1 LabVIEW Program (VI) ...............................................................................3
    2.2.2 One Example of LabVIEW Program (VI) .......................................................4

3 Data Acquisition .....................................................................................................5
  3.1 Communication with the process .........................................................................5
    3.1.1 Implementation of the NI 6221(68-pin) card .................................................6
    3.1.2 Display output signals on the interface ........................................................6
  3.2 Read and Write Data ..........................................................................................7
  3.3 User Interface and Data acquisition .....................................................................7

4 The design of input signal .........................................................................................9
  4.1 Model Quality .....................................................................................................9
  4.2 The considered factors in the design ..................................................................11
  4.3 Common input signals .......................................................................................12
    4.3.1 Filtered Uniform White Noise .....................................................................12
    4.3.2 Binary Random Signal ................................................................................12
  4.4 The Simulations and the results ..........................................................................12
    4.4.1 Filtered Uniform White Noise .....................................................................13
    4.4.2 Binary Random Signal ................................................................................14

5 Modeling .................................................................................................................16
  5.1 Physical modeling of the system ......................................................................19
  5.2 Linearization of the model ...............................................................................20
  5.3 Parametric identification ...................................................................................21
    5.3.1 Model Structures .........................................................................................21

6 Implementation of system identification software ..................................................23
  6.1 MATLAB-SITB .................................................................................................23
    6.1.1 The derived models ....................................................................................23
      6.1.1.1 ARMAX (4, 4, 3, 1) ...........................................................................24
      6.1.1.2 BJ (3, 3, 5, 7, 1) .............................................................................25
  6.2 Implementation of control design and Simulation module ................................27
    6.2.1 The user interface for the model .................................................................28
    6.2.2 Non-linear effects .......................................................................................28
1 Introduction

The main objective of the ‘System Identification Lab’ is to understand the general concept of system identification by acquiring the input and the output data from the process. The apparatus used in this lab is shown in Fig 1.1. The apparatus consists of a fan and hinged rectangular plate mounted on a slide. The control objective is the angular orientation of the plate. This orientation is controlled by blowing an air stream at the plate with a variable speed fan. The fan is driven by a DC voltage and the rotational angle is measured by a sensor. The plate is under the influence of air from the fan and the stochastic disturbance due to air turbulence around the plate. The DC voltage sent to the fan is the input data and displacement angle of the plate is the output data. This data has to be saved for system identification experiment. [10]

To implement all requirements for the lab an interface has to be designed. The interface should implement the following requirements.

- The design of input signal
- The specification of sampling time
- Plots of input and output data
- Data acquisition
The acquired data will be used to derive the linear model of the process. The report gives the detail of different LabVIEW tools used to design the user interface and share data. This report presents the details of practical application of LabVIEW used to acquire and analyze data from the process. System identification concerns the problem to derive the mathematical model of a process from the experimental data. The method is just to find the mathematical relation between the output and input under the influence of external disturbances without going into details what is actually happening inside the system. There are different theoretical model structures which are applied for the system identification. The parameters of these structures are estimated which are further used to derive the mathematical model of the process.

The derived model is simulated. This derived model is linear. But the process behavior is non linear. So some nonlinearity effects are also modeled in the simulation process in order to get similar output from the model to the process' output. The simulation process is also controlled from an interface similar to the ‘System Identification Lab’ interface. As the simulation reflects the process behavior it can also be used as another System Identification Lab, where no hardware is required.

LabVIEW is a graphical program designed to make interfacing with any measurement hardware. LabVIEW provides assistances which makes data acquisition quite simple. Further LabVIEW provides functions those are designed to extract useful information from the acquired data to analyze measurements and processing signals. LabVIEW tools can be used for data visualization, user interface design, software connectivity and creating reports. Thus LabVIEW can create applications which can be used to collect, analyze and share data with ease and with higher accuracy. LabVIEW make it easier to connect to I/O and integrate with software which makes easier to compare data from a process with the theoretical models.

The report explains also utilization of different LabVIEW tools used for simulation of a linear model derived after parametrical identification and details of how the nonlinearity is modeled. MATLAB System Identification Toolbox has been used for parametric identification. The Black Box model method has been applied. The report also contains the results and conclusions drawn about the process after carrying out different tests specially done for system identification purpose.

### 2 Introduction to the System

#### 2.1 The Hardware

The hardware is a resonant aerodynamic object and is shown in the Fig 2.1. The control objective is the angular orientation of a hinged rectangular plate and the control means is by blowing an air stream at the plate with a variable speed fan.

The fan is driven by a DC motor with pulse width modulation and rotation angle of the plate is measured by low friction potentiometer. The fan and its motor are mounted on a slide, by varying the position of the fan relative to plate the air transportation lag can be changed. The air turbulence around the plate naturally provides the stochastic disturbance.

The distance between the fan and the plate has been fixed. Different input signal are sent to the DC motor and respective output angle has been observed. The input and the output data has been collected during a specified period of time. This data is used to estimate the linear model of the process. Then derived model has been simulated. The derived model is linear. But the process is non linear so the non linear effects has been modeled during simulation.
Development of control lab interface for data acquisition using LabVIEW

2.2 The Software

The system identification process requires some necessary functions e.g. the design of user interface, the input signal, the specification of sampling time, data acquisitions and parametric identification. To implement these requirements some software is needed. LabVIEW is one such software. It is used for designing a user interface, construction of a linear model and for the simulation of the model. MATLAB’s System Identification Toolbox has been used for the parametric identification during the model derivation.

2.2.1 LabVIEW Program (VI)

LabVIEW programs are called virtual instruments or VI because their appearance and operation imitate physical instruments, such as oscilloscopes and multi meters. A LabVIEW VI contains three main components

- Front panel
- Block diagram
- Icon/connector pane

The front panel is the user interface for the VI. Front panel contains the interactive input and output terminals of the VI.

The block diagram contains graphical source codes. These codes are added using the graphical presentation of functions to control the front panel objects.
Icon/connector pane is used to use a VI inside the other VI, which is called a subVI. The upper right corner of the front panel and block diagram displays an icon, which can contain both texts and images. An icon identifies a subVI on the front panel of a VI. To use a VI as a subVI there is a need of a connector pane. Connector pane is a set of terminals that corresponds to controls and indicators of that VI [1].

2.2.2 One example of LabVIEW program(VI)

A simple program ‘Addition and subtraction.VI’ has been written in LabVIEW to illustrate the function of front panel and block diagram. The program will add and subtract two variables and display the results on the front panel.

The front panel of the program is shown in the Fig 2.2. The front panel has two controllers A respectively B and two numerical indicators A+B respectively A-B. The control A is a knob switch.

![Fig 2.2: Front panel of ‘add and subtraction.VI’](image)

There value of A is adjusted by rotating the knob. Control B is a numerical control where the value of B is entered. When the program runs the indicator A+B shows the sum of A and B, simultaneously the indicator A-B shows the subtraction between A and B.

The Fig 2.3 shows the block diagram consisting of graphical codes of the ‘add and subtraction.VI’.

When the program runs these codes executes.
3 Data acquisition

The main function of this lab is to collect the input and the output data. To implement that computer should have a good communication with the process. When the communication is achieved a user interface has been designed. The user interface has all the necessary function as all lab functions are implemented from this interface.

3.1 Communication with the process

To create a communication between the process and the computer National Instruments provides different input/output cards which are further supported by DAQ assistance. DAQ assistance is a simulation of data acquisition device. The DAQ assistance creates different channels for measurement and transfer signals from one form to other so that a computer can process. In this process ‘NI 6221(68-pin) Pin out’ has been used. The card has total 68 contact point. Every contact point has a different function.

Fig.3.1: NI 6221(68-pin)

Fig: 3.1 shows the picture of the card and Fig: 3.2 is the description of different function of the pins.
3.1.1 Implementation of the NI 6221(68-pin) card

AI in the specification is for analog input, AO is for analog output and GND is for ground. A sensor is provided on the hardware whose output voltage is proportional to the plate displacement angle. The output from the sensor is connected to pins 66 & 67 (AI9 and AIGND respectively), as this is an input to the computer. This signal is further assisted by a DAQ assistance provided on the LabVIEW’s block diagram (sec: 2.2.1). The DAQ assistance converts the electrical signal into digital data.

To send the control signal to the DC motor another DAQ is used. This DAQ assistance is connected to a control provided in the LabVIEW. The DAQ assistance in this case converts the digital data to analog signal. This analog signal is sent to the fan’s motor through the pins 22 & 55 (AO 0 and AO GND respectively) as this signal is an output from the computer. The wires have been soldered on the card NI 6221(68-pin).

3.1.2 Display output signal on the interface

Both signals that are received and sent have to be displayed on the interface (Front panel). The LabVIEW provides a waveform chart which is connected to the DAQ to display the output from the sensor. The waveform chart is a special type of numeric indicator that displays the plots of the data typically acquired at a constant rate. Similarly a waveform chart is connected to the control signal to the fan’s motor. Parameters are also set here to display the voltage (volts) in real time. The charts are scaled in a way that at the working point they display input voltage equal to 0 and the constant displacement angle equal to 110°. Fig 3.3 shows the waveform chart of the output signal. The Y axis of the chart shows the output angle and X axis shows the time in seconds. A numeric indicator has also been used for digital display of data. As seen in Fig 3.3 there is a digital indicator above the chart to display the digital value of the data.
3.2 Read and write data

In the function pallet a tool named ‘write to measurement file’ is provided. This tool is used to read and write the data in an ‘lvm (LabVIEW measurement)’ file. The file will write a data in the discrete form. In this work, parameters are set so that the file consists of two columns of synchronized input and output data respectively.

3.3 User interface and data acquisition

Fig.: 3.8 shows the picture of the user interface having all necessary functions of the lab. The appendix section (later in the report) contains the manual of the interface. Data from different inputs signals and their respective outputs have been acquired and saved in separate “lvm files”. These data files have been used for further work.
Development of control lab interface for data acquisition using LabVIEW

Fig 3.8: The user interface
4 The design of input signal

The design of an identification experiment includes several choices, such as, which signals to measure, when to measure, which signals to manipulate and how to manipulate them. It also includes some more practical aspects, such as how to condition the signals before sampling them i.e. choice of filters.

One of the main questions is which input signal should be applied to a physical system whose dynamics is to be identified. Most often the both input and output signals are sampled using a constant sampling interval \( T \) and this quantity \( T \) has to be chosen. So the main factors to be considered during the design are

- Which signals to be measured
- Sampling time
- How many samples
- Wave form of the input signal
- Frequency content of the input signal

4.1 Model Quality

In most cases, the input signals that are used are full of energy so that they can excite the system and force it to show its properties in a short period of time. These kinds of signals can give a lot of information of the process in a short experiment time. Here \( u(t) \) is the input signal and \( y(t) \) is the output signal. By introducing a delay operator

\[
q^{-1} u(t) = u(t - 1) \tag{4.1}
\]

The physical process can be modeled as a linear discrete time model

\[
y(t) = G_o(q)u(t) + v(t) \tag{4.2}
\]

where \( v(t) \) is the disturbance. The experimental data i.e. input and output data is used to construct a model \( G(q, \theta) \), which describes the properties of interest of the system \( G_o \).

The model is parameterized by a vector \( \theta \) and its estimate is denoted by \( \hat{\theta}_N \). The model error \( \theta_o - \hat{\theta}_N \) can be divided into two parts; the variance error and bias error. The variance error is due to noise \( v \) and the bias error is due to the fact that the model \( G(q, \theta) \) is not able to capture the whole dynamics of \( G_o \). Here it is assumed that bias errors are negligible and the model quality is measured by only taking variance error into account. The disturbance \( v(t) \) can be modeled as \( v(t) = H(q, \theta)e(t) \), where \( e(t) \) is zero mean white noise with variance \( \lambda_o \).

The predicted value of the output can be written as

\[
\hat{y}(t) = (1 - H^{-1}(q, \theta))y(t) + H^{-1}(q, \theta)G(q, \theta)u(t) \tag{4.3}
\]
and it is calculated with the help of old input and output data with a given value of $\theta$.

A common approach to determine $\hat{\theta}_N$ is to pick the minimizer of the squared prediction error,

$$\hat{\theta}_N = \arg \min_{\theta} \frac{1}{N} \sum_{t=1}^{N} (y(t) - \hat{y}(t))^2 \quad (4.4)$$

Let $P_N$ denotes the covariance matrix of $\hat{\theta}_N$. An early observation was that $P_N^{-1}$ can be shaped by the inputs. Much attention has been paid to $P_N^{-1}$ because it is an affine function of input spectrum $\phi_u$ and can be shaped by $u$.

If a linear model is flexible enough to capture the true dynamics, the bias error is zero, and then for open loop identification of estimates based on the criterion (4.4) it holds that

$$P^{-1}(\theta_o) = \frac{1}{2\pi \lambda_0} \int_{-\pi}^{\pi} F_u(e^{j\omega}, \theta_o) \phi_u(\omega) F_u^*(e^{j\omega}, \theta_o) d\omega + R_o(\theta_o) \quad (4.5)$$

Where

$$R_o(\theta_o) = \frac{1}{2\pi} \int_{-\pi}^{\pi} F_u(e^{j\omega}, \theta_o) F_u^*(e^{j\omega}, \theta_o) d\omega \quad (4.6)$$

$$F_u(e^{j\omega}, \theta_o) = H^{-1} \frac{dG(\theta_o)}{d\theta}$$

and

$$F_u(e^{j\omega}, \theta_o) = H^{-1} \frac{dH(\theta_o)}{d\theta}.$$ Here $\theta_o$ is the vector that parameterizes $G_o$ and $\lambda_o$ is the noise covariance.

In open loop experiment the cross spectrum between the input and the noise $\phi_{ue}(\omega) = 0$. The estimates $G$ and $H$ are asymptotically uncorrelated, even when they are parameterized with common parameters and for large data lengths, an approximate variance expression is given by:

$$\text{var}(G(e^{j\omega}, \hat{\theta}_N)) \approx \frac{n}{N} \frac{\phi_1(\omega)}{\phi_1(\omega)} \quad (4.6)$$

$$\text{var}(H(e^{j\omega}, \hat{\theta}_N)) \approx \frac{n}{N} \left| H_o(e^{j\omega}) \right|^2 \quad (4.7)$$

Where $n$ is the order of the model and $N$ is the number of samples. $H_o(e^{j\omega})$ is the model disturbance model $H(e^{j\omega}, \theta^*)$ where $\theta^*$ is the set of parameters that yields $G(e^{j\omega}, \theta^*) = G_o(e^{j\omega})$.

So some important observations can be made:

- The inverse of the covariance matrix of $\hat{\theta}_N$ is an affine function of the input spectrum $\phi_u$.
- The covariance of the model estimate will decay as $1/N$, so the estimate improves as the number of data increases.
• If $\phi_\nu$ is large, then the elements of $P$ are small. This implies the more input power is used the better the estimate. Also, from (4.6) we see that the variance of $G(e^{j\omega}, \hat{\Theta}_N)$ is small for frequencies where the input spectrum is large.

In practice, the actual input limitation typically concerns the amplitude constrains $u \leq u(t) \leq \bar{u}$. This desired property of the wave form therefore is defined in terms of the crest factor, which for a zero mean signal is defined as

$$C_r^2 = \frac{\max u^2(t)}{\lim_{N \to \infty} \frac{1}{N} \sum_{t=1}^{N} u^2(t)}$$ (4.8)

A good signal waveform is one that has a small crest factor. A good input signal with a zero mean value usually has a crest factor around one.

The fact that the input spectrum can be used to shape the covariance has been very important from an input design perspective. Another requirement for the input signals it should be persistently exciting which is $u_\nu(\omega) > 0$ for almost all $\omega$. So we can capture the system dynamics.

The open loop experiment with an input that is persistently exciting is sufficiently informative. [12]

4.2 The considered factors in the design of input signal.

The choice of input signal has a very substantial influence on the observed data. The input signal should excite the system during the experiment. The input signals determine the operating point of the system and which parts and modes of the system are excited during the experiment.

• Both input and output signals are sampled using a constant sampling interval $T$, the choice of sampling time is related to the system’s time constant. Sampling which is much faster than the system’s dynamics will lead to data redundancy. On the other hand sampling slower than the system’s dynamics will lead to loss of valuable information and the parameters identified from this data will not describe the dynamics completely. So a faster sampling is a rather better choice than slower sampling. [7]. A general rule for the sampling time is 10 samples per rise time. Rise time is the time taken by a step response to rise from 10% to 90% of its final value.

• The final choice for the identification experiment is the number of collected input-output measurements (N). The model accuracy gets better the more data that is used, but there may be other things that limit the experiment time.

• The waveform of the input signal must have a low crest factor.

• Two different aspects are associated with the choice of input signal. One concerns the second order properties of the input signal $u$, such as its spectrum $\Phi_u(\omega)$ and the cross spectrum $\Phi_{ue}(\omega)$ between the input and the noise. As this lab process is an open loop, the cross spectrum $\Phi_{ue}(\omega) = 0$. The other aspect concerns the shape of the signal. One can work with inputs being sums of sinusoids, or filtered white noise, or pseudorandom signals, or binary signals (assuming only two values), and so on. The signals discussed are inherently
periodic, like PRBS and the sum of sinusoids. All of them can in any case be made periodic by simple repetition.

4.3 Common input signals

So the main requirement for the identification of the linear system is that the input should contain many distinct frequencies, high input energy, low crest factor and it should be persistently exciting. In this section good and typical choices of input signals which can be applied in this lab are discussed.

For the identification of linear systems there are three basic facts that govern the choices:

- The asymptotic properties of the estimate (bias and variance) depend only on the input spectrum.
- The input must have limited amplitude: $u \leq u(t) \leq \bar{u}$
- Periodic inputs may have certain advantages e.g. they have most distinct frequencies in its spectrum, by taking average of the output power signal to noise ratio can be improved, a periodic input allows both formal and informal estimates of the noise level in the system and periodic signals give no leakage when forming the Fourier transform.

The basic issue for input signal design is to achieve a desired input spectrum for a signal with a crest factor as small as possible. It is always advisable to generate the signal and study its properties before using it as an input in an identification experiment. The following signals describe the typical choices of waveforms, and how to achieve the desired spectra. So the following signals have been used in this work to derive the model of the process and in the user interface of the lab.

4.3.1 Filtered uniform white noise

A simple choice is to let the signal be generated as uniform white noise, filtered through a linear filter. These filters are used to choose a frequency content of the signal so it will get the essential energy in that frequency band (the cut off frequency in bode plot) which is important for a system. With this one can achieve virtually any signal spectrum by proper choice of filter. The non-causal filters can be applied and transient effects can be eliminated, which gives a better spectral analysis.

4.3.2 Random Binary Signal

A random binary signal is a random process which assumes only two values. It can be generated in a number of different ways. The *telegraph signal* is generated as a random process which at any given sample has certain probability to change from one current level to the other one.

This is the other signal which has been used for model derivation. The problem with using low pass filter is that it can give the input signal a non-binary character, though, worse crest factor. An alternative is to increase the clock period in the signal generation. [11]

4.4 The simulations and results

The both signals have been simulated in the LabVIEW and used as input signals. The input and the output data have been saved and then used in MATLAB System Identification Toolbox to identify the parameters. MATLAB’s System Identification Toolbox has some function which measures the accuracy of the model. These functions are discussed in detail in section 6.1. The experiments have been done in MATLAB using both input signals with different properties i.e. sampling time, cut-off frequency of the filters. The following section gives the details of the two signals which have been
proved to be appropriate choices for input signals for model derivation. The final derivation of the model is discussed later (sec: 5.3.1 and 6.1.1).

4.4.1 Filtered Uniform white noise

First the data from an unfiltered white noise has been used for parametric identification. The results in MATLAB have not been satisfactory. When the transfer function is estimated in the MATLAB it has been noticed that the transfer function has higher amplitude in the low frequency region. So a input signal with low frequency can give a better model, Fig 4.1 shows the estimated frequency response when unfiltered white noise is used as an input signal.

![Frequency response](image)

Fig 4.1: Estimated frequency response

So the model accuracy can be improved by using the low pass filter. Low pass filters with different cut-off frequencies has been used to improve the results in MATLAB. It is concluded that a uniform white noise filtered with a low pass filter having a cut off frequency $2 \text{rad/s}$ is the a good choice for model derivation. So this signal has been used as one of the input signal to derive the model.

The following figures (Fig 4.2 and Fig 4.3) show the simulated input signal and the power spectral density of the signal. It can be seen in the power spectral density that the energy of the signal decreases as frequency is increased. It shows the input signal has a low frequency character.
It has to be noticed that this signal has been used only for model derivation. In the lab process unfiltered white noise has been used as an input signal.

4.4.2 Random binary signal having a sampling time 0.8 seconds

The random binary signal has also been simulated. The transfer function has been estimated in MATLAB. The same results are received. The transfer function has higher amplitude in low frequency region. The signal can be given a low frequency character by increasing the clock period of the signal generation. In this case sampling time is connected to both signal generation and sampling time. The
detailed information is given in the Appendix. Data has been collected from experiments with different sampling times and parameters have been estimated in MATLAB for the different experiments individually. It is concluded that the sampling time 0.8 seconds has given the best results in MATLAB. So this is the other signal which has been used for model derivation.

The following figures (Fig 4.4 and Fig 4.5) show the simulated signal and the power spectral density of this input signal.

Fig 4.4: The random binary signal having a sampling time 0.8 sec.
It can be seen in the power spectral density the energy of the signal decreases as frequency is increased. It shows the signal has a low frequency character. It has to be noticed that this signal has been used for model derivation. In the lab the user will decide the sampling time.

5 Modeling

The system has to be identified from the collected specified amount of data. Parametric identification has been applied for that. First the physical modeling has been done [5]. Then the different model structure based on Black Box modeling has been tested with the MATLAB System Identification Toolbox.

5.1 Physical modeling of the process

Fig: 5.1 shows a sketch over the process. The input $u(t)$ is the voltage to the fan’s motor and the output $y(t)$ is the displacement angle caused by the air from the fan. This displacement is also under the influence of turbulence caused by the surrounding air.

First the dynamics of the DC motor has been studied. Fig 5.2 shows a sketch over DC motor.
Current through the motor is given by \( i(t) \), R is the resistance and L is the inductance of the motor. When current is passed through DC motor it produces an electro motive force \( e_m \), which causes an angular velocity \( w(t) \). This angular velocity is proportional to the electromotive force so \( e_m = k_m w(t) \), there \( k_m \) is the motor’s constant. \( J_m \) s the moment of inertia around the axis of rotation. The damping due to friction is denoted by the constant \( \gamma \).

The equations (5.1), (5.2) describe the dynamics of the DC motor.

\[
\frac{di}{dt} = u(t) - i(t)R - k_m w(t) \quad (5.1)
\]

\[
J_m \dot{w}(t) = k_m i(t) - \gamma w(t) \quad (5.2)
\]

This equation (5.2) is derived from the different moment acting on the motor. \( J_m \dot{w}(t) \) is the total torque acting (Newton law of rotation), \( k_m i(t) \) is rotational torque produced from the motor and \( \gamma w(t) \) is the torque due to friction.

Assuming that inductance L in the motor is zero. The equation (5.1) becomes

\[
0 = u(t) - i(t)R - k_m w(t) \quad (5.3)
\]

So

\[
i(t) = \frac{u(t)}{R} - \frac{k_m}{R} w(t) \quad (5.4)
\]

By substituting this value in equation

\[
\dot{w}(t) = -\frac{1}{J_m} \left( k_m^2 \frac{R^2}{R} \right) w(t) + \frac{k_m}{J_m R} \cdot u(t) \quad (5.5)
\]
Substituting \[ d = \frac{1}{J_m} \left( \frac{k_m^2}{R} - \gamma \right), \] and \[ k_1 = \frac{k_m}{J_m R}, \] The fan’s angular velocity can be written as first order linear system.

\[
\dot{\omega}(t) = -d \omega(t) + k_1 u(t) \tag{5.6}
\]

Further the air velocity is assumed to be proportional to angular velocity, \( v = k_2 \omega \) so (5.6) can be written as

\[
\dot{v}(t) = -d v(t) + k_2 k_1 u(t) \tag{5.7}
\]

Total torque \( M_{tot} \) acting on the plate is \( J \dot{\phi}(t) \). \( J \) is the moment of inertia of the plate and \( \phi(t) \) is the displacement angle of the plate.

Fig 5.2: The different torques acting on the plate

So the equation of the moment acting on the plate can be defined as

\[
M_{Tot} = -M_{gravitation} - M_{airres} + M_{reaction} \tag{5.8}
\]

This equation is derived from the different moments acting on the plate

- \( M_{Tot} \): Torque derived by Newton’s second law of rotation.
**Development of control lab interface for data acquisition using LabVIEW**

- $M_{\text{gravitational}}$: Torque caused by gravitational force acting on the plate. This torque depends on mass of the plate, component of the gravitational acceleration acting on the plate.

- $M_{\text{airres}}$: Moment caused by air resistance which depends upon the angular velocity which is assumed to be proportional to angular velocity of the plate.

- $M_{\text{reaction}}$: Moment acting on the plate due to air pressure which is assumed to be proportional to the square of air velocity.

The displacement angle of the plate is $\phi(t)$, mass $m$ is reduced to a mass point which exists at a distance $l_g$ from the joint. $a$ is the effective area of the plate which is centralized at a distance $l$ from the axis of rotation. $J$ is the moment of inertia. Further the equation for the delay by air velocity from the fan is

$$T(t) = \frac{L + \sin(\phi(t))}{v(t)} \quad (5.9)$$

$L$ is the distance from fan to the stand. $k_1, k_2$ and $k_3$ are the constants. Where the numerator $L + \sin(\phi(t))$ is the distance covered by air from fan to the plate. The denominator $v(t)$ is the air’s velocity.

So the equation (5.8) can be written as

$$J\ddot{\phi}(t) = -mg l_g \sin(\phi(t)) - b\dot{\phi}(t) + k_3 v^2 (t - T) a l \cos(\phi(t)) \quad (5.10)$$

Equation (5.7) and (5.10) can be written in a vector form $\dot{x}(t) = f(x(t), u(t))$ where every component $x_i(t)$ is a state. So there is one state for the fan and two for the plate. The data of displacement angle $\phi(t)$ and voltage $u(t)$ is sampled and used as the output and the input data respectively.

### 5.2 Linearization of the model

The process has been linearized at a working point, where the system is stationary and approximately linear. It is important the process working does not deviate too much from this point. This is to avoid the nonlinear effects as derived model is linear.

For linearization following substitutions are done

$$\phi(t) = x_1(t) \quad (5.11)$$

$$\phi(t) = x_2(t) \quad (5.12)$$
Development of control lab interface for data acquisition using LabVIEW

\[ v(t) = x_1(t) \]  \hfill (5.13)

And the new equations for the model are

\[ \dot{x}_i(t) = x_i(t) \]  \hfill (5.14)
\[ \dot{x}_2(t) = -A \sin x_1(t) - B x_2(t) + C x_3^2 (t - T) \cos(x_1(t)) \]  \hfill (5.15)
\[ \dot{x}_3(t) = -x_3(t) + k_i u(t) \]  \hfill (5.16)
\[ y(t) = x_1(t) \]  \hfill (5.17)

Where \( A = mg l_s / J \), \( B = b / J \) and \( C = k_3 a l / J \). The output is denoted by \( y(t) \). As the working point is stationary so at this point \( x_i(t) = 0, i = 1, 2, 3 \), \( u(t) = u^0 \), \( x_i(t) = x_i^0 \) and \( y(t) = y^0 \), at all \( t \geq 0 \). The calculated values of \( \begin{pmatrix} x_1^0 & x_2^0 & x_3^0 \end{pmatrix} \) are \( \pi_1 \begin{pmatrix} 0 & k_i & u^0 \end{pmatrix} \), where

\[ \pi_1 = \arctan \left( \frac{C}{A} \left( \frac{k_i u^0}{d} \right)^2 \right) \], it is assumed that \( x_3^0 (t - T) \approx x_3^0 \)

After linearization the equations are

\[
\begin{pmatrix}
\Delta \dot{x}_1(t) \\
\Delta \dot{x}_2(t) \\
\Delta \dot{x}_3(t - T)
\end{pmatrix} = \begin{pmatrix}
0 & 0 & 1 \\
- A \cos(\pi_1) - C \left( \frac{k_i u^0}{d} \right)^2 \sin(\pi_1) & 0 & 2 C \frac{k_i u^0}{d} \cos(\pi_1) \\
0 & 0 & -1
\end{pmatrix} \begin{pmatrix}
\Delta x_1(t) \\
\Delta x_2(t) \\
\Delta x_3(t - T)
\end{pmatrix}
\]
\[ + \begin{pmatrix}
0 \\
0 \\
k_i
\end{pmatrix} \cdot \Delta u(t) \]  \hfill (5.18)
\[ \Delta y(t) = \begin{pmatrix} 1 & 0 \end{pmatrix} \Delta x_1(t) \]  \hfill (5.19)

There \( \Delta x_i = x(t) - x_i^0 \), \( \Delta u(t) = u(t) - u^0 \) and \( \Delta y(t) = y(t) - y^0 \). These equations describe the linear behavior of the process near the working point. Further substitutions are done

\[
\begin{pmatrix}
\Delta \dot{x}_1(t) \\
\Delta \dot{x}_2(t) \\
\Delta \dot{x}_3(t - T)
\end{pmatrix} = \begin{pmatrix}
0 & 1 & 0 \\
- E & - B & D \\
0 & 0 & - d
\end{pmatrix} \begin{pmatrix}
\Delta x_1(t) \\
\Delta x_2(t) \\
\Delta x_3(t - T)
\end{pmatrix} + \begin{pmatrix}
0 \\
0 \\
k_i
\end{pmatrix} \cdot \Delta u(t) \]  \hfill (5.20)

The linearized transfer function has been calculated from these equations.
Development of control lab interface for data acquisition using LabVIEW

\[ \Delta y(t) = \left( \frac{De^{-T_k}}{(s + d)(s^2 + Bs + E)} \right) \Delta u(t) \]  \hspace{1cm} (5.21)

5.3 **Parametric identification**

The received output is a combination of system’s ideal output and the external disturbances

\[ y(k) = y_0(k) + w(k) \]  \hspace{1cm} (5.22)

Let \( y(k) \) and \( u(k) \) is the sampled input and output respectively. The target is to estimate a mathematical model which suits best with the measured data. The ideal output \( y_0(t) \) is modeled as \( y_0(k) = G(q, \theta)u(k) \). \( G(q, \theta) \) is a rational function having a \( q \) operator. The parameters of the model are stacked in the vector \( \theta \). Similarly the external disturbance \( w(k) \) can be modeled as \( w(k) = H(q, \theta) e(k) \). \( e(k) \) is a white noise with a zero mean value.

Now the predicted output can be written as

\[ y(k) = G(q, \theta)u(k) + H(q, \theta)e(k) \]  \hspace{1cm} (5.23)

A general approach is to minimize the square of prediction error \( \epsilon(t) = y - \hat{y} \), \( \hat{y} \) is the predicted output from the model. The loss function which has to minimized can be written as

\[ f_N(\theta) = \frac{1}{N} \sum_{j=1}^{N} (y(t) - \hat{y}(t))^2 \]  \hspace{1cm} (5.24)

\( N \) are the number of samples. The predictor \( \hat{y}(t) \) is calculated with the help of input and output data with a given values of \( \hat{\theta}_N \). A common approach to determine \( \hat{\theta}_N \) is to pick the minimizer of the squared prediction error, Then the coefficients of model \( G(q, \theta) \) and \( H(q, \theta) \) those minimize the loss function (5.24) are estimated [6]:

\[ \hat{\theta}_N = \arg \min_{\theta} f_N(\theta) \]  \hspace{1cm} (5.25)

5.3.1 **Model structures**

To estimate the parameters there are different model structures in theory. The system identification toolboxes are based on these model structures. These parameters do not present any physical quantity instead they are used only to describe properties of the input-output relation. These model structures are called ‘Black-box’ models [6]. In this work the ‘Black box’ modeling has been used.

Following the equation (5.23), \( G(q, \theta) \) and \( H(q, \theta) \) are defined by
Development of control lab interface for data acquisition using LabVIEW

\[ G(q, \theta) = \frac{B(q)}{(F(q) \ast A(q))} \]  
\[ H(q, \theta) = \frac{C(q)}{(D(q) \ast A(q))} \] (5.26) (5.27)

So the Fig 5.3 shows the signal flow diagram of the general linear model. The following polynomial equations define \( A(q), B(q), C(q), D(q) \) and \( f(q) \).

![Signal flow diagram of the models](image)

To find the most suitable model different model structures are used. There are three structures named as ARX, ARMAX and BJ models are commonly used in the real applications.
These model structures are created by setting one or more of $A(q), B(q), C(q), D(q)$ and $f(q)$ equal to 1.

When $C(q), D(q)$ and $F(q)$ is equal to 1, the general linear model reduces to ARX model. Thus $G(q) = B(q)/A(q)$ and $H(q) = 1/A(q)$. The order of the model is decided by deciding the parameters $na, nb$ and $nk$. Where $na$ is the number of poles of both $G(q)$ and $H(q)$. $nb$ is zeros of $G(q)$. $nk$ is delay in number of samples.

When $D(q)$ and $F(q)$ are equal to 1, the general model reduces to ARMAX. Thus $G(q) = B(q)/A(q)$ and $H(q) = C(q)/A(q)$. The order of the model is decided by $na, nb, nc$ and $nk$. $nb$ and $nc$ are number of zeros of $G(q)$ and $H(q)$ respectively. $na$ is the number of poles of $G(q)$ and $H(q)$.

When $A(q)$ is equal to 1, the general linear model reduces to BJ model, which is normally a common approach, has a structure like $G(q) = B(q)/F(q)$ and $H(q) = C(q)/D(q)$ having parameters $nb, nc, nd, nf$ and $nk$. $nf$ and $nb$ are the poles respective zeros of the $G(q)$. $nc$ and $nd$ are the poles respective zeros of the $H(q)$.

Every model structure has been tested and their parameters have been changed until the best model has been achieved [2], [6].

6 Implementation of the system identification software

As discussed earlier although LabVIEW has a System Identification Toolkit, in this work the MATLAB’s System Identification Toolbox has been used to identify the different parameters. The structures discussed in section 5.3.1 have been tested in MATLAB. Then the parameters from the most suitable structure have been used for simulation.

6.1 MATLAB- SITB

MATLAB system identification toolbox enables to build accurate and simplified models of complex systems from time-series data. The features of the toolbox include a ‘graphical user interface’ where different structures can be tested. The data saved has been divided into two parts. One part is used as a working data and other as a validation data. First a simple structure usually ARX is tried.

Then the model structure is tested and validated. As system identification involves many parameters i.e. sampling time, order of the model, so it is extremely hard to find one model which can perfectly reflect all the observed behaviors of the system. So there can be a number of models those are used to describe one process [8].

There are different functions which measures the correctness of the model [7]. These methods are

- The loss function (5.24).
- MATLAB-SITB provides a calculated value FIT which is a measurement of model’s adjustment.
• Prediction error $\epsilon(k) = y(k) - \hat{y}(k)$, autocorrelation function $r_r(l) = \mathbb{E}[\epsilon(k+l)\epsilon(k)]$ should look like a white noise
• Cross correlation $r_{ra}(l) = \mathbb{E}[\epsilon(k+l)u(k)]$ should be approximately zero.

System identification toolbox’s GUI provides windows where these factors can be analyzed. The toolbox calculates the value of loss function (5.24) for the respective functions. The toolbox also sticks out the parameters after identification. Then the parameters from the most suitable structure have been used to derive the model of $G(q)$ and $H(q)$. The parameters are the values of $\theta$ and the variance of white noise $e(k)$.

6.1.1 The derived model

To derive the linear model of the system the input and the output data have been saved. The two input signals discussed in section 4.4 have been used. The data files using both input signals have been used in the ‘ident’ to identify the parameters. After trying the different model structures with different model order, the structures ARMAX (4, 4, 3, 1) i.e. $na = 4, nb = 4, nc = 3, nk = 1$ and BJ (3, 3, 5, 7, 1) i.e. $nb = 3, nc = 3, nd = 5, nf = 7, nk = 1$ have showed the most satisfactory results and fulfilled the conditions discussed in section 6.1 quite well. To derive the mathematical model parameters obtained from these two models have been used.

6.1.1.1 ARMAX (4, 4, 3, 1)

The parameters given by ARMAX (4, 4, 3, 1) are

$$A(q) = 1 + 0.1772q^{-1} - 0.3001q^{-2} - 0.06381q^{-3} - 0.007541q^{-4}$$

$$B(q) = 2.778q^{-1} + 3.467q^{-2} + 0.3627q^{-3} - 0.5276q^{-4}$$

$$C(q) = 1 + 0.2573q^{-1} - 0.09155q^{-2} + 0.1058q^{-3}$$

By using these equations the transfer functions for $G(z)$ and $H(z)$ have been derived. The derived functions are:

$$G(z) = \frac{2.778z^4 + 3.467z^3 + 0.3627z^2 - 0.5276z}{z^5 + 0.17722z^4 - 0.3001z^3 - 0.06381z^2 - 0.007541z}$$

$$H(z) = \frac{z^2 + 0.2573z^4 - 0.09155z^3 + 0.1058z^2}{z^5 + 0.17722z^4 - 0.3001z^3 - 0.06381z^2 - 0.007541z}$$

The variance of the noise $e(k)$ is 0.1961.

Figure 6.2 shows the step response identification of ARMAX (4, 4, 3, 1) done in MATLAB and the Fig 6.3 shows the step response of the process when one step is sent. When the factors like
amplification, rise time and final value are compared it is noticed that the identified step response resembles the process’ step response.

![Fig 6.1: Process’ step response](image1)

![Fig 6.2: Step response for ARMAX (4, 4, 3, 1)](image2)

6.1.1.2 BJ (3, 3, 5, 7, 1)

The parameters given by BJ (3, 3, 5, 7, 1) are

\[
B(q) = 0.01236q^{-1} - 0.02472q^{-2} + 0.02601q^{-3}
\]  

(6.6)
Development of control lab interface for data acquisition using LabVIEW

\[ C(q) = 1 - 0.6061q^{-1} + 0.1511q^{-2} - 0.503q^{-3} \quad (6.7) \]

\[ D(q) = 1 - 2.315q^{-1} + 2.069q^{-2} - 1.791q^{-3} + 1.019q^{-4} - 0.1616q^{-5} \quad (6.8) \]

\[ F(q) = 1 - 2.815q^{-1} + 3.3q^{-2} - 3.088q^{-3} + 2.915q^{-4} \quad \ldots 
\ldots - 1.4532q^{-5} - 0.0375q^{-6} + 0.1812q^{-7} \quad (6.9) \]

By using these equations transfer function of \( G(z) \) and \( H(z) \) are derived. The derived functions are

\[ G(z) = \frac{0.01236z^6 - 0.02472z^5 + 0.02602z^4}{z^7 - 2.8156z^6 + 3.3z^5 - 3.088z^4 + 2.915z^3 - 1.453z^2 - 0.0375z + 0.1812} \quad (6.10) \]

\[ H(z) = \frac{z^5 - 0.6061z^4 + 0.1511z^3 - 0.503z^2}{z^6 - 2.3152z^5 + 2.069z^4 - 1.791z^3 + 1.0192z - 0.1616} \quad (6.11) \]

The variance of the noise \( e(k) \) is 0.020.

Fig 6.3 shows the step response identification for BJ (3, 3, 5, 7, 1) model.
This step response does not resemble completely the process’ step response. So the parameters obtained from the ARMAX (4, 4, 3, 1) has been applied for simulation.

7 Simulation

In the previous section the modeling and parametric identification of the process has been done. The purpose of modeling is to build a model that captures those features of the process which we want to observe. The mathematical models have been derived in section 6.1.1. Simulation requires a simulation environment where this model can be constructed and simulated. Simulation has helped in many ways in this work like the quality of the derived model has been observed. The derived model is linear and the process is nonlinear. So the nonlinear effects have been modeled during simulations. The main objective is that the model’s output should resemble the process’ output. When the model gives the desired results it can be used for another lab. A lab where no hardware is required as the model simulation reflects the process’ behavior. The LabVIEW tools have been used for the construction and simulation of the model. An interface has also been designed for simulation. This interface is quite similar to the Lab interface. The difference is in this interface there are two indicators those are used to draw the transfer functions \(G(z)\) and \(H(z)\) equations and sampling time can not be fed.

7.1 LabVIEW tools for control design and simulation model

LabVIEW provides the ‘control design tool kit and simulation module’ for constructing the mathematical model and simulation.

7.1.1 Control design toolkit

LabVIEW provides a control design toolkit which has control design assistance, a library of VIs and control design MathScript function designed to construct the mathematical model of a dynamical system. There are different kind of models like linear and non linear, continuous and discrete, time variant and time invariant.

Control design toolkit supports only linear and time invariant models. Model type can be both continuous and discrete. The Control design toolkit can represent a dynamic model in three forms.

- Transfer function
- Zero-pole-gain
- State space

In this work ‘transfer function (discrete time)’ has been used to create and simulate the model. Control design toolkit provides even VI to convert the model from one form to other and to connect models in series and in parallel.

Next step is to simulate the constructed model and acquire data. This data has to be compared with the data from the process. The comparison can give a deeper look on the model validation [3].

7.1.2 Simulation Module

LabVIEW provides Simulation module for simulation purposes. Simulation module provides different tools i.e. simulation loop, simulation diagram to create a simulation environment to simulate the both closed and open loop models. After constructing a model as discussed in 5.1, that model is connected to a transfer function model block provided in simulation module. This block is
placed in the simulation loop and by setting simulation parameters the model is simulated. Simulation module even provides the tools like saturation and dead zone to model the nonlinear effects of the process. The more details regarding graphical programming and parameter settings have been given in Appendix section (Appendix C). [3]

7.2 Implementation of Control design and simulation module

The both programs have been installed. These programs provide the tools and sub VIs required for control design and simulation. These tools and sub VIs are used to construct the model, simulate the model and to display the transfer function of $G(z)$ and $H(z)$ on the front panel and to save the results from simulations.

7.2.1 The user interface for the simulation of model

The interface for the simulation has also been designed. This interface is quite similar to the process interface. The main difference is the sampling time can not be controlled when data is measured. There are two blocks in the interface. In these blocks the derived models $G(z)$ and $H(z)$ are displayed. The sampling time is constant which is equal to 0.015 seconds. The rest of the functions are quite similar.

7.2.2 Non linear effects

As mentioned the derived and simulated model is linear but the process behavior is not linear for all input signal values. Some non linear effects are modeled during simulation. To model these effects first the process’ behavior is studied.

To study the behavior of the process the DC signal has been used. DC signals with different amplitudes have been sent and the outputs have been observed. DC signals had 0 initial values and final values have been written in the input column of Table: 7.1.

The results received from these signals have been written other Max Output, Time and Rise Time column of the table. Max output is the final output, Time is time taken by the process to reach the final output value and Rise time is the time taken by step response to rise from 10% to 90% of its peak value.

<table>
<thead>
<tr>
<th>Input</th>
<th>Max Output(degree)</th>
<th>Time(sec)</th>
<th>Rise Time(sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2</td>
<td>1.4</td>
<td>5</td>
<td>2.5</td>
</tr>
<tr>
<td>-1</td>
<td>3.9</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>10-11</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>18-19</td>
<td>4</td>
<td>2.5</td>
</tr>
<tr>
<td>2</td>
<td>24-26</td>
<td>3</td>
<td>2.5</td>
</tr>
<tr>
<td>3</td>
<td>24-26</td>
<td>3</td>
<td>2.3</td>
</tr>
<tr>
<td>4</td>
<td>24-26</td>
<td>3</td>
<td>2.1</td>
</tr>
<tr>
<td>5</td>
<td>24-26</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>25-26</td>
<td>2.8</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>25-26</td>
<td>2.5</td>
<td>1.8</td>
</tr>
<tr>
<td>8</td>
<td>25-26</td>
<td>2.5</td>
<td>1.6</td>
</tr>
<tr>
<td>9</td>
<td>25-26</td>
<td>2.5</td>
<td>1.5</td>
</tr>
<tr>
<td>10</td>
<td>25-26</td>
<td>1.5</td>
<td>1</td>
</tr>
</tbody>
</table>

Table: 7.1
The table: 7.1 shows that the output is almost saturated after 2 volts. The difference after 2 volts is that the time and rise time becomes shorter as the input voltage is increased.

The input and output have been plotted. Fig: 7.1 shows the plot

The process’ output is quite linear between -1 and 2. In order to avoid non linear effects it is important that the input signal values are not deviating too much from this linear interval. Now the process behavior in this linear input region has been studied. DC signals had the initial values 0 and final value is written in the input column of the table: 7.2 and Max Output is the final output.

<table>
<thead>
<tr>
<th>Input</th>
<th>Max Output (degree)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1.00</td>
<td>3.9</td>
</tr>
<tr>
<td>-0.75</td>
<td>6</td>
</tr>
<tr>
<td>-0.50</td>
<td>7.4</td>
</tr>
<tr>
<td>-0.25</td>
<td>8.9</td>
</tr>
<tr>
<td>0.00</td>
<td>10.9</td>
</tr>
<tr>
<td>0.25</td>
<td>12.6</td>
</tr>
<tr>
<td>0.50</td>
<td>14</td>
</tr>
<tr>
<td>0.75</td>
<td>16</td>
</tr>
<tr>
<td>1.00</td>
<td>17.9</td>
</tr>
<tr>
<td>1.25</td>
<td>20.2</td>
</tr>
<tr>
<td>1.50</td>
<td>22.5</td>
</tr>
<tr>
<td>1.75</td>
<td>24.5</td>
</tr>
<tr>
<td>2.0</td>
<td>26.3</td>
</tr>
</tbody>
</table>

Table: 7.2

The input – output plot can be seen in Fig: 7.2
It has been observed from the Table: 7.1 and Fig 7.1 that the process’ output is saturated after input 2 volts and before -1volts. The process is linear in this interval. Saturation has been used. This is a simulation tool which has been provided in LabVIEW which limits the range of signal. The input signal is sent to the saturation. The upper range is set at 2 and lower is set at -1. When the simulation is done the output from the simulated model is quite similar to the process’s output.

7.3 A comparison

The output from the model has been compared with the output from the process, when same signal is sent to both the. The Table: 7.3 shows the output values from the model and the process in the input interval -1 to 2 volts where the process is linear. Input column is the final DC input signal amplitude. Initial values are 0.

<table>
<thead>
<tr>
<th>Input</th>
<th>Max Output(degree) from process</th>
<th>Output from the model</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1.00</td>
<td>3.9</td>
<td>3.6</td>
</tr>
<tr>
<td>-0.75</td>
<td>6</td>
<td>5.3</td>
</tr>
<tr>
<td>-0.50</td>
<td>7.4</td>
<td>7.4</td>
</tr>
<tr>
<td>-0.25</td>
<td>8.9</td>
<td>8.8</td>
</tr>
<tr>
<td>0.00</td>
<td>10.9</td>
<td>10.5</td>
</tr>
<tr>
<td>0.25</td>
<td>12.6</td>
<td>12.8</td>
</tr>
<tr>
<td>0.50</td>
<td>14</td>
<td>14.3</td>
</tr>
<tr>
<td>0.75</td>
<td>16</td>
<td>16.4</td>
</tr>
<tr>
<td>1.00</td>
<td>17.9</td>
<td>17.9</td>
</tr>
<tr>
<td>1.25</td>
<td>20.2</td>
<td>20.0</td>
</tr>
<tr>
<td>1.50</td>
<td>22.5</td>
<td>22.0</td>
</tr>
<tr>
<td>1.75</td>
<td>24.5</td>
<td>24.3</td>
</tr>
<tr>
<td>2.0</td>
<td>26.3</td>
<td>26.0</td>
</tr>
</tbody>
</table>
Development of control lab interface for data acquisition using LabVIEW

Table: 7.3

The fig below shows the plots of both outputs with respect to the input.

Fig 7.4: Plots of the process’ and model’s output

The plot shows that the output from the model is quite similar to the output from the process. So the simulated model reflects the process’ behavior quite well.

8 Conclusion and future work

In this work the user interface for the ‘System Identification Lab’ has been designed. This interface will be used by the students of the course ‘Modeling of dynamical system’. The input and the output data have been acquired. A linear model has been derived and simulated. Later some non-linear effects are modeled and the outputs from the process and the model have been compared. They are quite similar. The interface for the simulation has also been designed. Thus the model simulation can be used as another ‘System Identification Lab’ where no hardware is required.

Some problems are still not solved. When data is acquired from the process the sampling time can be changed from the interface. But when the data is acquired from the model there is a fixed sampling time. The control design toolkit which has been used does not support the nonlinear models. So during simulation the nonlinearity is not modeled in a classical way.

The interface of the model simulation can be put on a website. So that the students do not need to come to the lab room instead they can do at computer room or at home. It is not very complicated. There is a ‘web publishing tool’ under the tools menu can be used to put the VI on the website. At a time only one user can have access to the VI. This problem can be solved by making copies of VI. Then give them different names like 1.VI, 2.VI. There should be a separate password for every student so everybody will access only his own VI. Due to license agreement only 5 students can access their VI during the same time [9].
References

[2] LabVIEW ‘system identification toolkit manual’
   http://phubner.eng.ua.edu/Files/NL_622x_Specifications.pdf
[6] Old exam from course “Modeling of dynamical system (2E1282)” given by
   Department of Automatic control at KTH
[9] NI LabVIEW Internet toolkit
    http://www.kri.com.sg/
Appendix A

Manual of the user interface
Development of control lab interface for data acquisition using LabVIEW
Modeling of dynamical systems (2E1252)

Lab 3: System identification
Lab PM

Vivek Sharma
School of Electrical Engineering
Automatic Control
1. **Introduction**

System identification is an experimental method to derive a mathematical model of a process by collecting input and output data. In this lab you have to collect the appropriate input and output data and derive a linear model which can describe the system’s behavior around one working point. MATLAB’s System Identification Toolbox has been used to derive the model. The objective of the Lab is to gain practical experience of how a system identification experiment is carried out. The main tasks are as follows

i. To collect data and analyze it for system identification.

ii. To choose an appropriate choice of the model.

iii. To derive a mathematical model from the acquired data.

iv. To assess the functionality of the derived model.

2. **Presentation**

The given lab can be done individually or in groups of two. The results have to be presented in the form of a report, written either in English or Swedish. The report should essentially consist of all the experimental details along with technical and scientific terms.

**The format of the report should be as follows:**

i. A good title, Author’s name, and e-mail must be written on the front page of the report.

ii. One or maximum two authors.

iii. The report should include all necessary graphs and plots. The report should not be longer than six sides of A4 pages.

The report must be handed in STEX and is not accepted by E-mail or fax. After the assessment the grades pass or fail are awarded. However one will be given an opportunity to correct the mistakes in case the report is not fulfilling the required demands.

**The report must explain the following technical concepts:**

i. Choice of sampling time.

ii. Detailed explanation of the applied input signal, how the data is acquired from the process.

iii. Number of samples used for identification and validation.

iv. Use more than one model structure and compare results with the help of the validation data.

v. Present results in the form of Bode plot, poles/ zeros map.

vi. Describe validation with simulation and correlation analysis.

vii. Comment why the selected model is more convincing than other alternatives.
viii. Derive a simple physical modeling of the process and compare that with the experimentally derived model.

3. **The process**

Process to be modeled consists of a fan and a plate mounted on the stand. Input is the voltage applied to the fan’s motor and output is the angular displacement angle of the plate caused by the air from the fan. Figure 1 is the picture of the process.

![Fig 1: The process](image)

4 **Procedure**

In the lab mathematical model has to be derived of this process by acquiring input and output data. The model should take into account small variations around the working point. The variations should be large enough so that the disturbances could not influence the output signal.

A constant voltage is used to drive the motor which in turn causes the plate to be displaced by a certain angle. This angle is the point where variations are done. The data thus collected is used to derive a model.

The following parameters are to be specified:

i. Input signal

ii. Collection of data
iii. Iteration and processing of data
iv. Spectral analysis
v. Parametrical estimation of the model
vi. Model validation

The last two points are to be repeated until a satisfactory model has been received.

4.1 *Start the computer*

On the desktop you will find a folder ‘Modeling of dynamical system’. Double click on the folder so you can see the directory. Now double click on the file ‘Lab 3’ from the directory. The window with an interface showing Figure 2 will appear on the screen.

![Fig 2: The Interface](image)

The different blocks of the interface have specific functions and described as follows:

**Sampling time:**
Feed the time in seconds for sampling.

**Random signal/Step response:**

Here is a selector between Random Insignal and Step response.

When you want to see the step response you have to select the Step response. Later when you want to send a random input signal select Random Insignal.

**Step Response:**

To check the step response a step is sent from the block Step response block. The value of amplitude of a step is entered in this block.

**Random Insignal:**

When you want to send a random input you can make a choice between two different sorts of random signals. In the Random Insignal block there is a selector between two different sorts. One is Telegraphic Insignal which is an input varying randomly between two levels depending on offset and amplitude value. Other is Uniform white noise also varying randomly between two levels depending on amplitude and offset.

**Scopes:**

In interface there are two scopes named as Voltage and Angle. In Voltage the input (voltage to the fan’s motor) will be plotted and in angle the output (displacement angle) will be plotted. There is a digital display above the scopes where digital values can be read.

**Warning:**

One has to work in around a linear point but if your maximum input value (offset + amplitude) is out side the limit $\pm 10$ then no signal will go to the process and a lamp will glow in the Warning block indicating the incorrect value. But it is highly recommended to work in the positive area i.e. Offset +Amplitude $\geq 0$.

**Save data:**

To save the data the switch in the block save data has is pressed.

**Stop:**

Stop button is used to stop execution.

### 4.2 Start experiment

Switch on the power. Do not change anything on the fan and plate’s control panel. A constant input is sent to the motor which causes a constant displacement angle. This is the point where one has to work.

**4.2.1 Step response**

To choose the appropriate sampling time step response has to be plotted. For that first of all feed 0.05 in the Sampling time block. Select the step response and run the program by clicking the
arrow on the menu bar as shown in fig 3. This arrow has to be pressed every single time to start a new execution.

The amplitude of the step response is 0 in the beginning. To send a signal enters a positive value in step amplitude and double click on Reset signal or press Enter. When step response has been plotted stop execution. Step response can be checked by zooming with the help of tools from the graph palette as shown in fig 4. Try different steps to calculate the average rise time which is the time taken by step response to rise from 10% to 90% of its peak value.

4.2.2 Acquiring data

Now is the time to acquire data. Sampling time is decided from the step response experiment. A general rule for sampling time is

\[ \text{Sampling time} = \frac{\text{risetime}}{10} \]

Feed this value in sampling time. Select Random Insignal instead of Step Response and also select either Telegraphic Insignal or Uniform White Noise from the block Random Insignal. Now enter Amplitude and Offset value. Run the program and press the button once in Save data as will glow green as shown in Fig 4.
When you think that the enough data has been saved press the **Save Data** button again and then **Stop** execution. Go the folder ‘**Modeling of the Dynamical Systems**’ and open the file ‘**data.lvm**’ in word pad. This is data that you have acquired. The first column is the input and second column is the output. Save this file on your floppy or USB memory stick.

### 5 The choice of the model structure

To get a good model the first step is to start with a simple structure like ARX model. If it seems that the derived model is not good enough then its time to try other models like ARMAX and BJ. Theory for this in MATLAB can be found in the computer exercises. For that first of all one has to download the file in MATLAB by command: `load ('filename.lvm');` After opening the

*identification toolbox* write the command:

```
load filename.lvm
```

By writing `who` in the command window your variables are going to be `filename u  y`.

If you want to download the data in an ordinary MATLAB file. You can use the following commands:

```
filename=load('filename.lvm');
```

For input you can use the command:

```
u= filename(:,1);
```

For output you can use the command:

```
y=filename(:,2);
```

### 6 Software

The required software MATLAB and System Identification Toolbox for the further process is available on the computers in the computer room.
Appendix B

Example of a Report
Lab 3

Parametric Identification

2E1282 Modeling of dynamical systems

Vivek Sharma
Vivek@kth.se

18 Oct 2006

Summary

In this lab a process has been modeled both from the physical principles and parametrical identification from the acquired sampled data. Three different model structures have been tested.

1. Introduction

From a limited data acquired from the process a model has been estimated. The process consists of a fan and a plate hanging on the stand. Input $u(t)$ is the dc voltage sent to the fan’s motor and output $y(t)$ is the displacement angle caused by the air from the fan.

2. Theory

2.1 Physical modeling of the process

Firstly, the physical model of the system has been derived [1]. As the input $u(t)$ is the voltage to the fan’s motor and output $y(t)$ is the displacement angle caused by air from the fan.

The fan’s angular velocity can be written as first order linear system.

$$\dot{\omega}(t) = -d\omega(t) + k_1 u(t)$$

(2.1)

It is assumed that the DC motor’s electrical dynamics are much faster than mechanical so one equation could be skipped. Further the air velocity is assumed to be proportional to angular velocity, $v = k_v \omega$ so (2.1) can be written as

$$\dot{v}(t) = -dv(t) + k_v k_1 u(t)$$

(2.2)

So the equation of the moment acting on the plate can be defined as

$$J \ddot{\phi}(t) = -mg l_s \sin(\phi(t)) - b \dot{\phi}(t) + k_v v^2(t - T) Al \cos(\phi(t))$$

(2.3)

$\phi(t)$ is the displacement angle of the plate, mass $m$ is reduced to a mass point which exists at distance $l_s$ from the joint. $A$ is the effective area of the plate which is centralized at a distance $l$ from the axis of rotation. Further the equation for the delay by air velocity from the fan is
$$T = \frac{L + \sin(\varphi(t))}{v(t)}$$

(2.4)

$L$ is the distance from fan to the stand. $k_1, k_2$ and $k_3$ are the constants. Equation (2) and (3) can be written in a vector form $\dot{x}(t) = f(x(t), u(t))$ where every component $x_j(t)$ is a state. So there is one state for fan and two for the plate. The displacement angle $\varphi(t)$ is converted into voltage. This voltage is sampled and used as output data.

### 2.2 Parametric identification

The received output is a combination of system’s ideal output and the external disturbances

$$y(k) = y_0(k) + w(k).$$

(2.5)

Let $y(k)$ and $u(k)$ is the sampled output and input respectively. The target is to estimate a mathematical model which suits best with the measured data. $y_0(t)$ is the ideal output which can be written as $y_0(k) = G(q, \theta)u(k)$.

$G(q, \theta)$ is a rational function having a displacement operator $q$. Similarly the external disturbance $w(k)$ can be described as

$$w(k) = H(q, \theta) e(k).$$

$e(k)$ is white noise with a zero mean value.

Now the output can be written as

$$y(k) = G(q, \theta)u(k) + H(q, \theta)e(k)$$

(2.6)

A general approach is to minimize the square of prediction error $\varepsilon(t) = y - \hat{y}$, $\hat{y}$ is the predicted output from the model. The loss function which has to minimized can be written as

$$f_N = \frac{1}{N} \sum_{i=1}^{N} (y(t) - \hat{y}(t))^2$$

(2.7)

$N$ are the number of samples. $\hat{y}(t)$ is calculated with the help of input and output data with a given values of coefficients of $G(q, \theta)$ and $H(q, \theta)$. Then the known number of coefficients of $G(q, \theta)$ and $H(q, \theta)$ those minimizes the loss function (2.6) are estimated [2]. The mean value of the noise is zero.

### 3. Practical

#### 3.1 Data acquisition

The measured output has to be sampled. The sampling time $T$ should be enough fast that it will not miss any information from the systems dynamics and simultaneously not that fast either that unnecessary noise turbulence starts influencing. A general rule to 10 samples per rise time is used for sampling time.
To get an accurate value of rise time different steps have been sent to the process and sampling time $T$ used is 0.05 seconds. So it is found that the rise time is around 1.8 seconds. By using the rule it is decided to set sampling time $T = 0.2$ sec.

Next step is to collect input and output data for system identification process. To get many frequencies different a signal varying between two levels has been used. Out of two alternatives for random input signal the telegraphic signal has been used. The offset and amplitude value used is Offset= 0.5 and amplitude= 0.5. So the telegraphic signal was varying between 0 and 1 randomly. Total 604 samples have been saved where first 302 are used for working data and last 302 are used for validation data.

Pre processing and parametrical estimation has been done in the computer program MATLAB-SITB.

### 3.2 Parametrical identification

Three different structures ARX, ARMAX and BJ have been used for the parametrical identification. Section 2.1 describes that $G(q, \theta)$ has at least three poles and estimated delay is around 2-5 samples. The different general rules for measurement of the correctness of model are:

i. The loss function $f_N$

ii. MATLAB-SITB provides a calculated value FIT which is a measurement of model’s adjustment.

iii. Prediction error $\varepsilon(k) = y(k) - \hat{y}(k)$.

iv. Autocorrelation function $r_e(l) = E[\varepsilon(k + l)\varepsilon(k)]$ should look like a white noise.

v. Cross correlation $r_{eu}(l) = E[\varepsilon(k + l)u(k)]$ should be equal to 0.

Moreover the Bode plot of the model can be compared with the a non-parametrical estimation of $G(e^{iw})$.

The first structure which is tried is ARX. Where $G(q) = B(q)/A(q)$ and $H(q) = 1/A(q)$. The order of the model is decided by deciding the parameters $na, nb$ and $nk$. Where $na$ are the number of poles of both $G(q)$ and $H(q)$. $nb$ are zeros of $G(q)$. $nk$ is delay in number of samples. The optimized value after several iterations of $na, nb$ and $nc$ is 5, 5 and 3 respectively.

Next structure is ARMAX, where $G(q) = B(q)/A(q)$ and $H(q) = C(q)/A(q)$. The order of the model is decided by deciding the parameters $na, nb, nc$ and $nk$. $na$ and $nb$ are number of zeros of $G(q)$ and $H(q)$ respectively. $na, nb$ and $nc$ are number of zeros of $G(q)$ and $H(q)$. The optimized values of $na, nb, nc$ and $nc$ are 4, 4, 4 and 2 respectively.

Next structure is BJ, where $G(q) = B(q)/A(q)$ and $H(q) = C(q)/A(q)$. Having parameters $na, nb, nc, nd$ and $nk$. $na$ and $nb$ are the poles respective zeros of the $G(q)$. $nc$ and $nd$ are the poles respective zeros of the $H(q)$. The optimized values of $na, nb, nc, nd$ and $nk$ are 3, 3, 7, 5 and 2 respectively.
After these structures and parameters are more suitable for the process, further these structures are compared with each other. The following table describes the values of $f_N$ and FIT gained from these structures.

<table>
<thead>
<tr>
<th>Model</th>
<th>$f_N$</th>
<th>FIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARX(5,5,3)</td>
<td>0.110356</td>
<td>57.17</td>
</tr>
<tr>
<td>ARMAX(4,4,4,2)</td>
<td>0.10992</td>
<td>57.17</td>
</tr>
<tr>
<td>BJ(3,3,7,5,2)</td>
<td>0.103731</td>
<td>57.32</td>
</tr>
</tbody>
</table>

The table above shows the values of $f_N$ and FIT obtained from the different structures. They results are quite similar, but for the further work only ARMAX and BJ model has been selected because of less loss function $f_N$ value than ARX model.

### 3.2.1 ARMAX (4, 4, 4, 2)

To analyze this model first zero and pole positions as been checked. The fig below shows the zeroes and pole position of ARMAX model.

![Poles and Zeros](image)

As seen all poles are inside the unit circle and there is one zero outside.

Fig below shows the estimated autocorrelation $r_c(l)$ and cross correlation $r_m(l)$ for the ARMAX process. As seen
The fig below shows a comparison between frequency responses of ARMAX model and the spectral model.

And the figure below shows the plot of both predicted output from the model and the process’s output.
3.2.2 *BJ (3, 3, 5, 2)*

Similarly features of BJ model have been analyzed. The fig below shows the zero and pole position of the model.

Fig shows that all poles are inside the unit circle and two zeros are outside.
Development of control lab interface for data acquisition using LabVIEW

Fig below shows the estimated autocorrelation $r_e(l)$ and cross correlation $r_{en}(l)$ for the BJ process.

Fig below shows a comparison between the frequency response from the BJ model and the spectral model.
Fig below shows the predicted output from the model and the process output.

![Graph showing Measured and simulated model output over time](image)

4 Conclusion

Now the process has been modeled both from the physical methods and the parametrical identification.

It is noticed that the ARX is a good structure to start with, because of less parameters. Later ARMAX, BJ can be applied to achieve the more accuracy. In this experiment both ARMAX and BJ has given quite similar results so both models can be a good choice. BJ model has shown a little better in case of FIT value, auto and cross correlation as seen in the graphs.

References

[1] Old exam from course “Modeling of dynamical system (2E1282)” given by Department of Automatic control at KTH
Appendix C

Parameter Settings in the LabVIEW codes
Development of control lab interface for data acquisition using LabVIEW
C.a The block diagram of User interface for the process

LabVIEW is a graphical language. So the graphical programming has been done on the block diagram. Different parameters are set according to the requirement. Different LabVIEW tools provided on the ‘function palette’ have been used for functions like communications, sampling time, input signal, plots and save data.

- Communication

Two DAQ assistances have been used. The first ‘DAQ assistance’ is to receive the output signal. It will convert the electrical signal into the analog value. In the properties of ‘DAQ assistance’ there is a task timing section. In this section parameters are set in ‘acquisition mode’ for signal generation. Acquisition mode has been set equal to 1 sample (HW timed). So the signal will update after every sampling time interval. Same settings have been done for ‘DAQ Assistance 2’ which is to convert the input analog signal into the electrical signal.

- Sampling time

The sampling time has to be controlled by the user. In the loop section on the ‘while loop’ has been provided. All tools have been placed on this loop. A millisecond delay named ‘wait till next ms’ has been provided. All codes executes after the time interval (ms) specified in this delay. This delay is controlled from the interface. But the value entered in the control is multiplied by 1000 to convert delay from milliseconds to seconds (Fig: C.1). So the value entered in the controller is sampling time in seconds.

![Fig: C.1: Control, constant (1000) and wait until next ms](image)

- Input signals

In the input section a tool ‘simulate signal’ is provided to simulate signal. This tool has been used to simulate all three signals. The table C.1 shows the configuration done for these three signals.

<table>
<thead>
<tr>
<th>Signal</th>
<th>DC Input signal</th>
<th>Telegraphic random</th>
<th>Uniform white noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signal type</td>
<td>DC</td>
<td>Square</td>
<td>Square with uniform white noise</td>
</tr>
<tr>
<td>Frequency</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Samples per second</td>
<td>1000</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

Table: C.1

The amplitude of the DC input is controlled from the interface (Front panel), the amplitude and offset of the square is also controlled from the interface and the duty cycle is varying randomly between 0 and 100. In uniform white noise the amplitude of the square wave is set at 0 in order to generate uniform white noise. The offset and noise amplitude is controlled from the interface. Duty cycle of frequency is also varying from 0 to 100.

- Plot data
Development of control lab interface for data acquisition using LabVIEW

Both the input and output analog signal has been contacted to voltage charts. In order to plot data in real time an example VI ‘real time chart.VI’ has been used. This example VI is provided in ‘help’ in the menu bar. This VI explains how to configure chart to plot data in real time.

- **Save data**

In ‘read and write section’ tool ‘write LabVIEW measurement’ is provided. This tool is used to save the data. The file name where data will be stored is ‘data.lvm’. Both input and output signal has been sent to this file. The data is saved in the text form. There are two columns in the file. The first column is the input signal and second column is the output signal.

In properties of the file configurations shown in Fig: C.2 is done to save data.

![Fig.C.2: configuration setting for lvm file](image)

In **action** ‘save to one file’ is to write the data only in one file named ‘data.lvm’. This file is saved in the folder modeling of the dynamical system.

In **If the file already exists** ‘Overwrite file’ is to overwrite the recently saved data on the old data.

In **File format** ‘Text (LVM)’ is to save data in text form.

In **segment header** ‘No header’ to avoid any header in the data files.

In **X value columns** ‘Empty time column’ to avoid time column in the saved data.
C.b The block diagram of User interface for the model

Block diagram from the model has many similar functions which have already been used in the block diagram of the process. The main difference is that as no hardware is used so there is no need of ‘DAQ assistance’ and during input and output data acquisition the sampling time is not specified at the user interface. To generate input signals, plot and save data the similar LabVIEW tools and parameters have been used but to construct and simulate model other LabVIEW tools have been used. These tools are provided on the ‘control and simulation toolkit’. The details of these tools are as follows

- **Model construction**

On the function pallet, there is a block named ‘control design’. Different LabVIEW tools are provided on this block to construct the model. Some of these are used in this work

- ‘CD construct transfer function.vi’ has been used to construct the function. In this block transfer function’s denominator and nominator values are fed. Output from this tool is fed to two tools named ‘CD draw transfer function equation.vi’ and ‘convert TF to SIM siso transfer function.vi’.

- ‘CD draw transfer function equation.vi’ is used to draw the transfer function equation. By connecting a indicator to its output the equations can be seen on the Front Panel (user interface).

- ‘convert TF to SIM siso transfer function.vi’ converts the control design model to a model which can be used to simulate.

- **Simulate model**

On the function pallet, there is a block named ‘simulation’. Different LabVIEW tools are provided on this block to simulate the model. Some of these are used in this work

- A tool named ‘simulation loop’ is used to simulate the model. All simulation functions and subsystems are placed within this loop. It has to be noticed that this loop is not a ‘while loop’. By right clicking on the ‘simulation loop’ a list of different option is appeared. These options are used to set simulation and timing parameters. These parameters can not be specified on the Front Panel. This is reason that the sampling time can not be specified on the user interface.

- Tool ‘discrete transfer function’ is placed within this loop. This is used to simulate. Output from ‘convert TF to SIM SISO transfer function’ and simulated input signal is fed to this tool. Output from this tool is fed to indicator and lvm file to plot and save data.

- A tool ‘saturation’ which is a simulation of saturation is used a saturation in order to saturate the input signal in a desired interval.
Development of control lab interface for data acquisition using LabVIEW