MEEN 364
Vijay Alladi, 2001. John Mangan, 2002. Andrew Rynn, 2002.
Justin Mlcak, 2005.
Last Update: 22 August 2006

Lab 1 – Introduction to MATLAB & LabVIEW

Introduction

Computers are increasingly important tools for mechanical engineers as they become more powerful and less expensive. They are often used to design and implement control systems because programming offers great flexibility. By simply changing the value of a parameter, or by just changing a few lines of code, the characteristics of the controller can be changed. The computer also offers tools for instant feedback of the results and means for storing the data. The use of computers and different software as tools for making measurements on dynamic systems and their control will be emphasized throughout the course.

LabVIEW is a graphical programming language that uses icons instead of text to create applications. In contrast to text based programming languages, where instructions determine the program execution, LabVIEW uses dataflow programming where data determines execution

In LabVIEW you build applications using a set of objects and tools. The user interface is known as the front panel. Using graphical representation of functions, the actual program is represented in the form of a flow chart. This makes program development and debugging much easier.

LabVIEW has the capability to communicate with hardware such as Data Acquisition boards. LabVIEW contains libraries for data collection, analysis and storage. LabVIEW also has traditional program development tools. LabVIEW is designed to facilitate hardware instrumentation, data acquisition and control systems programming.

Objective

The objective of this lab session is

1. To become familiar with creating Virtual Instruments (VI's) in LabVIEW and using it to build your own applications.
2. To learn how to solve a system of differential equations in MATLAB.

PreLab

Go through the handout on LabVIEW. Also, review the MATLAB tutorial.

Lab Procedure for LabVIEW

1. Build a Virtual Instrument (VI) that will convert a temperature input in °C to °F.
    a. Add a numeric control to the Front Panel for the user to input the temperature in °C.
    b. Add a numeric indicator to the Front Panel to display the temperature in °F.
    c. Add two thermometer indicators to the Front Panel to display the input and output temperatures.
    d. To convert from °C to °F, use the relationship $^\circ F = \frac{9}{5}{}^\circ C + 32$.

   e. Only allow the VI to work in the range of –200 °C to 2000 °C.
      i. Use an In Range and Coerce block in conjunction with a case structure.
      ii. Display an error message if the input temperature is out of range.

Tip: The last values entered into LabVIEW controls are not automatically saved. Upon reopening a previously saved LabVIEW file, the default values for the controls appear rather than the values that appeared when the file was last saved. To prevent this from happening, select **Operate»Make Current Values Default** each time before saving.

2. Build a VI that generates a noisy sinusoid and displays it on a graph.
   a. Generate a 100 Hz sinusoidal waveform.
   b. Add Gaussian white noise with a standard deviation of 0.1 to the sine wave.
   c. Use sub VIs to calculate the mean value and the RMS value of the noisy sine wave.
   d. Display the mean and RMS values on the front panel using digital indicators.
   e. Use a while structure for the VI so that you will see a continuous display of the waveform.
3. Replace the sine waveform sub VI in the previous VI with a basic function generator VI.
   a. Modify the VI to rectify the signal (take the absolute value).
   b. Add an additional waveform graph to display the rectified signal along with the original signal.
4. Modify the waveform generated and note the changes on the waveform graphs and in the mean and RMS values. Compare observed values with the "exact" mean and RMS of the theoretical signals.

Lab Procedure for Matlab

The following is a slider crank mechanism.

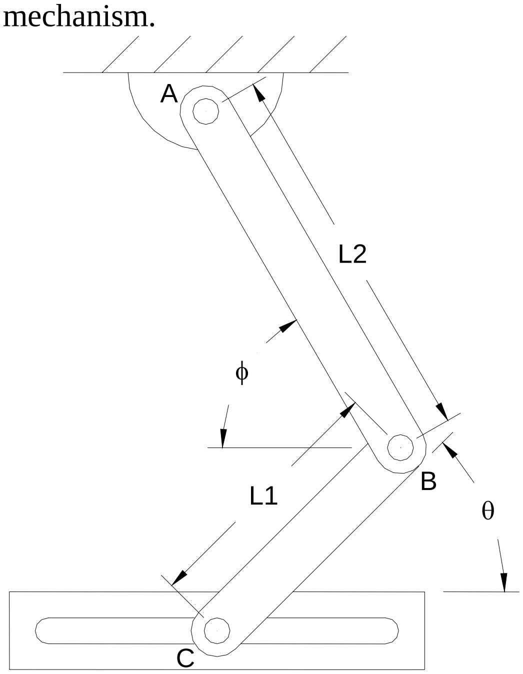

Figure 1-1 - Slider Crank Mechanism (drawn by Christopher Cullum)

Vijay Alladi, 2001. John Mangan, 2002. Andrew Rynn, 2002.
Justin Mlcak, 2005.
Last Update: 22 August 2006

Consider a coordinate system with origin at point A of the slider crank mechanism. Furthermore, consider lateral constrains to the slider so it can only translate in the vertical direction (no translation in the horizontal direction and no rotation).

It can be modeled by the following non-linear equations

$$M(y,t)\dot{y} = f(y,t) \tag{1}$$

where

$$M(y,t) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{ml_1^2}{12} & 0 & 0 & \frac{l_1\sin(\theta)}{2} & -\frac{l_1\cos(\theta)}{2} & -\frac{l_1\sin(\theta)}{2} & \frac{l_1\cos(\theta)}{2} \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{Ml_2^2}{3} & -l_2\sin(\phi) & -l_2\cos(\phi) & 0 & 0 \\ 0 & -\frac{ml_1\cos(\theta)}{2} & 0 & 0 & 0 & -1 & 0 & -1 \\ 0 & -\frac{ml_1\sin(\theta)}{2} & 0 & -ml_2\sin(\phi) & -1 & 0 & -1 & 0 \\ 0 & m_b l_1\sin(\theta) & 0 & -m_b l_1\sin(\phi) & 0 & 0 & 1 & 0 \\ 0 & l_1\cos(\theta) & 0 & l_2\cos(\phi) & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2}$$

$$f(y,t) = \begin{bmatrix} \dot{\theta} \\ 0 \\ \dot{\phi} \\ \frac{Mgl_2\cos(\phi)}{2} \\ -mg + \frac{ml_1\sin(\theta)\dot{\theta}^2}{2} \\ ml_2\cos(\phi)\dot{\phi}^2 - \frac{ml_1\cos(\theta)\dot{\theta}^2}{2} \\ m_b l_2\cos(\phi)\dot{\phi}^2 - m_b l_1\cos(\theta)\dot{\theta}^2 \\ l_2\sin(\phi)\dot{\phi}^2 + l_1\sin(\theta)\dot{\theta}^2 \end{bmatrix} \tag{3}$$

and $$y = \begin{bmatrix} y(1) \\ y(2) \\ y(3) \\ y(4) \\ y(5) \\ y(6) \\ y(7) \\ y(8) \end{bmatrix} = \begin{bmatrix} \theta \\ \dot{\theta} \\ \varphi \\ \dot{\phi} \\ B_x \\ B_y \\ N_x \\ N_y \end{bmatrix} \tag{4}$$

The position and the velocity of the slider (point C) are given by

$$x = l_2 \cos(\phi) - l_1 \cos(\theta) \tag{5}$$

$$\dot{x} = l_1 \sin(\theta)\dot{\theta} - l_2 \sin(\phi)\dot{\phi} \tag{6}$$

Parameter convention

m –	mass of bottom link, assume a value of 5 [Kg].
M –	mass of top link, assume a value of 5 [Kg].
m_b -	mass of horizontal slider assume, a value of 2.5 [Kg].
l_1 -	length of bottom link, assume it as 1 [m].
l_2 -	length of top link, assume it as 1.5 [m].
θ $and \phi$ -	are from the figure [rad].
B_x, B_y –	Reaction forces between top and bottom links [N].
N_x, N_y –	Reaction between the bottom link and the horizontal slider [N].

1. Simulate the position and velocity of the slider using an M-file in MATLAB.
2. Refer to the MATLAB Tutorial for examples.
3. Solve the differential equations for θ and ϕ.
4. Use the kinematic constraint equations (5) and (6) to find the position and velocity of the slider.
5. Plot the position and velocity of the slider.

Things you learned in this Lab

1 Building front panel
2 Building the diagram for the VIs
3 Using basic controls and indicators
4 Using of structures and loops-case structures, for structures, while loops
5 Using Graphs and charts
6 Solving ODE's using MATLAB.