

ME340: Lab Manual 7

Double Arm Pendulum, Lagrange's Equations, and Linearization

PRINT, STAPLE, and READ all materials concerning Lab 7. Printing lab materials in lab is not allowed. A stapler may not be available in lab. Unstapled material will not be accepted.

Introduction

Linear equations have a very specific form. This form makes linear systems relatively simple to use as models, analyze, and describe in state space (matrix) form. However, many mechanical systems are inherently non-linear. In order to make it possible to apply numerous linear analysis methods to non-linear systems, we must exercise an approximation process of *linearizing* non-linear system equations.

This lab will demonstrate the significance of linearization techniques and investigate the accuracy and limit of such approximation techniques on an actual double pendulum system. The main purpose of this lab is to acquaint you with the main methods of analyzing non-linear dynamical systems: non-linear numerical simulation and analytical linear analysis via a linearized model.

A Simple Introduction to Linearization

An example of linearization is the *small angle approximation*, $\sin(\theta) \approx \theta$ for small θ . In this case, $\sin(\theta)$ is linearized about the point $\theta = 0$. Figure 1 shows this approximation.

Linearization produces a linear equation from the non-linear equation. This linear equation will have the form

$$y = m\theta + b \tag{7.1}$$

The linear equation will be tangent to the function at the point of linearization (in this case zero). For the small angle approximation, $m = \left. \frac{d}{d\theta}(\sin \theta) \right|_{\theta=0} = 1$ and $b = (\sin \theta - m\theta) \Big|_{\theta=0} = 0$. Note that the linearization only holds for values close to the point of linearization.

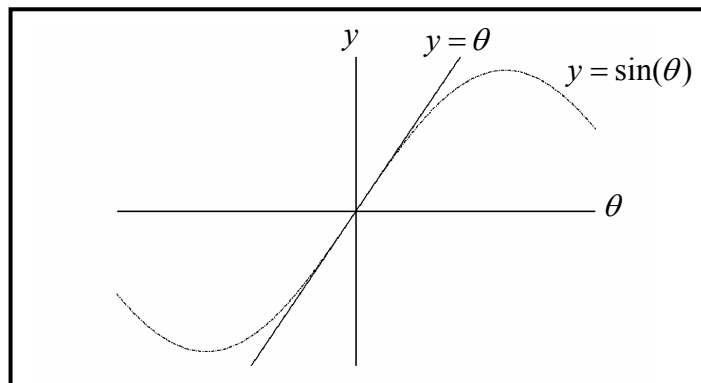


Figure 1: A sine wave and its linearization

Linearizing Functions of Multiple Variables

Suppose we wish to linearize a function of the form $f(x_1, \dots, x_m)$ where x_1, \dots, x_m are the m variables of the function f . From the previous example, a linearized equation only holds for values close to specific points called *operating points*. Define the operating points of the function as $\bar{x}_1, \dots, \bar{x}_m$. Using Taylor series expansion of f about the operating points gives

$$f(x_1, \dots, x_m) = \underbrace{f(\bar{x}_1, \dots, \bar{x}_m) + \sum_{k=1}^m \left\{ (x_k - \bar{x}_k) \frac{\partial f}{\partial x_k} \bigg|_{(\bar{x}_1, \dots, \bar{x}_m)} \right\}}_{\text{Linear Terms (KEEP!)}} + \underbrace{\frac{1}{2!} \sum_{k=1}^m \left\{ (x_k - \bar{x}_k)^2 \frac{\partial^2 f}{\partial x_k^2} \bigg|_{(\bar{x}_1, \dots, \bar{x}_m)} \right\}}_{\text{Non-Linear Terms (DISCARD!)}} + \underbrace{H.O.T.}_{\text{Higher Summations}} \quad (7.2)$$

Don't worry; it's not as complicated as it looks. When using Taylor series, a linear approximation assumes that the non-linear summation terms are small about the operating point and can be neglected. The series reduces to the linear *approximation*

$$f(x_1, \dots, x_m) \approx f(\bar{x}_1, \dots, \bar{x}_m) + \sum_{k=1}^m \left\{ (x_k - \bar{x}_k) \frac{\partial f}{\partial x_k} \bigg|_{(\bar{x}_1, \dots, \bar{x}_m)} \right\} \quad (7.3)$$

This is simply a “fill-in-the-blank” answer to linearization; solve for the differential and then evaluate at the operating point. Realize that the linearized approximation *may not* be a good approximation near the operating point. The approximation accuracy depends on the function shape *and* the range of values that the linearized equation needs to approximate.

Pendubot Model

The Pendubot is a double pendulum found at the College of Engineering Control Systems Laboratory. It is used for educational purposes in several controls courses. The Pendubot consists of two rigid aluminum links. There are optical encoders at each of the joints of the Pendubot and the whole system is actuated at the first link's origin. The system is controlled via Matlab commands. These will be introduced as needed. A simplified representation of the Pendubot is shown in Figure 2.

You will not be using the actuator in this lab, but a demonstration of what can be done with simple controls will use both the position encoders and the actuator.

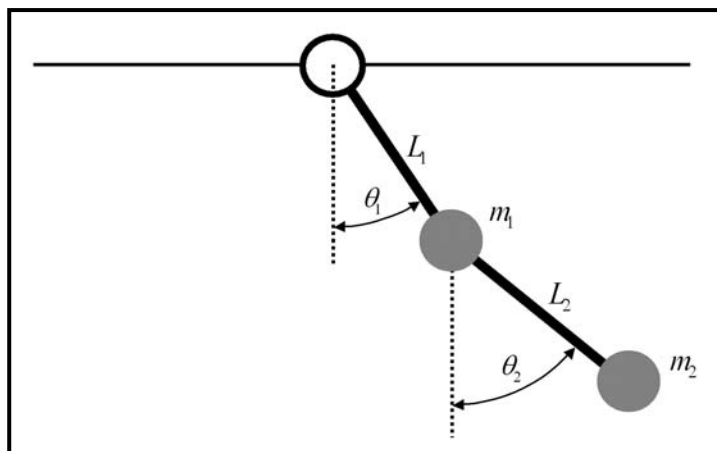


Figure 2: The double pendulum system

Pendubot's Non-Linear Equations

of Motion

Lagrange's equations give the most convenient method for derivation of the equations of motion. After considerable application of simple calculus and algebraic manipulation, the two equations of motion for the system are

$$\begin{aligned}(m_1 + m_2)L_1^2\ddot{\theta}_1 + m_2L_1L_2\ddot{\theta}_2 \cos(\theta_2 - \theta_1) - m_2L_1L_2\dot{\theta}_2^2 \sin(\theta_2 - \theta_1) + (m_1 + m_2)gL_1 \sin \theta_1 &= 0 \\ m_2L_2^2\ddot{\theta}_2 + m_2L_1L_2\ddot{\theta}_1 \cos(\theta_2 - \theta_1) + m_2L_1L_2\dot{\theta}_1^2 \sin(\theta_2 - \theta_1) + m_2gL_2 \sin \theta_2 &= 0\end{aligned}\quad (7.4)$$

Note that *Pre-Lab 7* asks for a derivation of these equations. They are given here as a *checkpoint* for your work. To simplify Equations (7.4), define the following constants

$$\begin{aligned}p_1 &= (m_1 + m_2)L_1^2 \\ p_2 &= m_2L_2^2 \\ p_3 &= m_2L_1L_2 \\ p_4 &= (m_2 + m_1)L_1 \\ p_5 &= m_2L_2\end{aligned}\quad (7.5)$$

and rewrite the equations of motion as

$$\begin{aligned}p_1\ddot{\theta}_1 + p_3\ddot{\theta}_2 \cos(\theta_2 - \theta_1) - p_3\dot{\theta}_2^2 \sin(\theta_2 - \theta_1) + p_4g \sin \theta_1 &= 0 \\ p_2\ddot{\theta}_2 + p_3\ddot{\theta}_1 \cos(\theta_2 - \theta_1) + p_3\dot{\theta}_1^2 \sin(\theta_2 - \theta_1) + p_5g \sin \theta_2 &= 0\end{aligned}\quad (7.6)$$

Linearizing the Equations of Motion

Before proceeding with the linearization of the non-linear pendulum model, we must choose an *equilibrium point* (our operating point) about which to linearize. If a system is at an equilibrium point, it will remain at that point unless an outside force acts upon the system. One obvious equilibrium point is the "down-down" position (i.e. $\theta_1 = \theta_2 = 0$). The Pendubot has three other equilibrium points.

The non-linear terms of Equation (7.6) are

$$\begin{aligned}f_1(\theta_1, \theta_2) &= \cos(\theta_2 - \theta_1) \\ f_2(\theta_1, \theta_2, \dot{\theta}_2) &= \dot{\theta}_2^2 \sin(\theta_2 - \theta_1) \\ f_3(\theta_1) &= \sin \theta_1 \\ f_4(\theta_1, \theta_2, \dot{\theta}_1) &= \dot{\theta}_1^2 \sin(\theta_2 - \theta_1) \\ f_5(\theta_2) &= \sin \theta_2\end{aligned}\quad (7.7)$$

Linearizing Equations (7.7) with Equation (7.3) and replacing the respective terms in Equations (7.6) gives the following linearized equations of motion

$$\begin{aligned} p_1\ddot{\theta}_1 + p_3\ddot{\theta}_2 + p_4g\theta_1 &= 0 \\ p_2\ddot{\theta}_2 + p_3\ddot{\theta}_1 + p_5g\theta_2 &= 0 \end{aligned} \quad (7.8)$$

Now, the equations can be conveniently put into 2nd-order matrix form as

$$\begin{bmatrix} p_1 & p_3 \\ p_3 & p_2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} p_4g & 0 \\ 0 & p_5g \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \bar{0} \quad (7.9)$$

System Identification

After extensive efforts at plant identification, the following equivalent values have been experimentally determined:

$$L_1 = 0.1508 \text{ m} \quad L_2 = 0.1683 \text{ m} \quad m_1 = 0.4373 \text{ kg} \quad m_2 = 0.1615 \text{ kg}$$

Explanation of the Simulink Model

The non-linear *Simulink* model is provided for this lab. The *Simulink* model is shown in Figure 3. This simulation file will still need to be linearized in lab. To do this, you will need to know and understand the structure of the simulation file.

Recall that the non-linear equations of motion are

$$\begin{aligned} p_1\ddot{\theta}_1 + p_3\ddot{\theta}_2 \cos(\theta_2 - \theta_1) - p_3\dot{\theta}_2^2 \sin(\theta_2 - \theta_1) + p_4g \sin \theta_1 &= 0 \\ p_2\ddot{\theta}_2 + p_3\ddot{\theta}_1 \cos(\theta_2 - \theta_1) + p_3\dot{\theta}_1^2 \sin(\theta_2 - \theta_1) + p_5g \sin \theta_2 &= 0 \end{aligned} \quad (7.10)$$

Putting this into matrix form gives

$$\underbrace{\begin{bmatrix} p_1 & p_3 \cos(\theta_2 - \theta_1) \\ p_3 \cos(\theta_2 - \theta_1) & p_2 \end{bmatrix}}_M \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} p_3\dot{\theta}_2^2 \sin(\theta_2 - \theta_1) - p_4g \sin \theta_1 \\ -p_3\dot{\theta}_1^2 \sin(\theta_2 - \theta_1) - p_5g \sin \theta_2 \end{bmatrix} \quad (7.11)$$

Solving for M^{-1} with $M^{-1} = \text{adj } M / \det M$ gives (*adj* M stand for *adjoint* of matrix M)

$$M^{-1} = \frac{1}{\det(M)} \begin{bmatrix} p_2 & -p_3 \cos(\theta_2 - \theta_1) \\ -p_3 \cos(\theta_2 - \theta_1) & p_1 \end{bmatrix} \quad (7.12)$$

Pre-multiplying Equation (7.11) by M^{-1} gives a solution for the highest derivative

$$\begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = \frac{1}{\det(M)} \begin{bmatrix} p_2 & -p_3 \cos(\theta_2 - \theta_1) \\ -p_3 \cos(\theta_2 - \theta_1) & p_1 \end{bmatrix} \begin{bmatrix} p_3\dot{\theta}_2^2 \sin(\theta_2 - \theta_1) - p_4g \sin \theta_1 \\ -p_3\dot{\theta}_1^2 \sin(\theta_2 - \theta_1) - p_5g \sin \theta_2 \end{bmatrix} \quad (7.13)$$

This equation can be rewritten compactly as

$$\begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} M_{11}^{-1} & M_{12}^{-1} \\ M_{21}^{-1} & M_{22}^{-1} \end{bmatrix} \begin{bmatrix} q_1(\theta_1, \theta_2, \dot{\theta}_2) \\ q_2(\theta_1, \theta_2, \dot{\theta}_1) \end{bmatrix} \quad \text{Note that } M_{12}^{-1} = M_{21}^{-1} \quad (7.14)$$

From Equations (7.13) and (7.14), the *Simulink* block diagram can be created. Follow the description below.

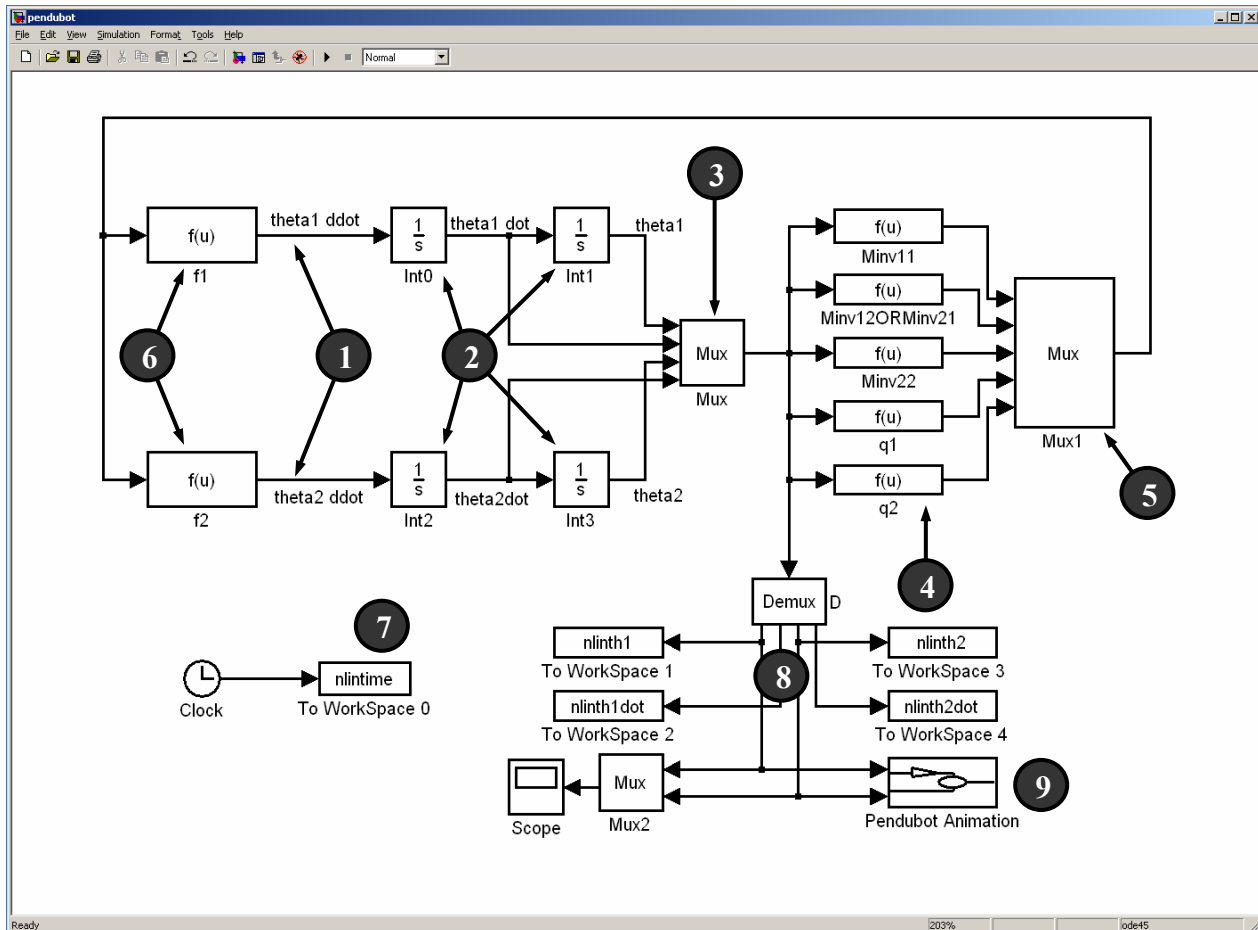


Figure 3: Nonlinear *Simulink* model.

Description of Simulation Loop

1. Begin with $\dot{\theta}_1$ and $\dot{\theta}_2$, represented by the indicated lines.
2. Integrate $\ddot{\theta}_1$ and $\ddot{\theta}_2$ twice to obtain $\dot{\theta}_1$, $\dot{\theta}_2$, θ_1 , and θ_2 . These will be used to calculate the non-linear terms.
3. *Mux* the data into one line. This *mux* simply represents 4 data lines as 1 line.
4. Five function blocks to calculate the five unique non-linear terms of Equation (7.14). For example, the bottom-most function block labeled q_2 calculates the term $q_2(\theta_1, \theta_2, \dot{\theta}_1) = -p_3 \dot{\theta}_1^2 \sin(\theta_2 - \theta_1) - p_3 g \sin \theta_2$.
5. *Mux* the five calculated values into a single line and then feed them back to close the loop.
6. These are functions that do the point-by-point multiplication of the non-linear terms in Equation (7.14) to solve for $\ddot{\theta}_1$ and $\ddot{\theta}_2$.
7. Outputs the time vector to the *Matlab* workspace for plotting.
8. Output $\dot{\theta}_1$, $\dot{\theta}_2$, θ_1 , and θ_2 to the workspace for plotting.
9. Animation function that will show the motion of the pendulum during simulation.

Experiment I: Linearizing the Non-Linear Simulink Model

1. Open the non-linear Simulink model. The file is located in $f:\text{labs}\backslash\text{me340}\backslash\text{pendubot}\backslash$ and it is called pendubot.mdl . Start *Simulink* and open this file. The file should resemble Figure 3.
2. Re-save pendubot.mdl as nonlinear.mdl in your folder on the local drive.
3. Then, re-save pendubot.mdl as linear.mdl in your folder on the local drive.
4. Now we will linearize the nonlinear model. In linear.mdl , change the appropriate blocks to linearize the equations (Minv11, Minv12, Minv22, q1(t), q2(t), f1, f2).
5. Rename the variables in the *To Workspace* blocks as linth1 (linth1 stands for linearized theta1), linth2 , linth1dot , and linth2dot by double clicking on the blocks. Change the time output in the *To Workspace* time block to lintime , (stands for *linear model time*). Save your block diagram.
6. Calculate the previously defined p parameters using the numerical values for L and m above. At the *Matlab* prompt, enter the values of the p parameters. For example, for p_1 :

```
>> p1=0.0136;
```

7. Do the same for the g variable.
8. At this point, the block diagrams are ready to run. Give the same initial condition, $\theta_1 = 0.1$, in both the non-linear and linear block diagrams and run them both. Plot the results with the following command and compare the dynamics.

```
>> plot(lintime,linth1,'- ',nlintime,nlinth1,'--')  
>> legend('Theta 1, Lin. Sys.', 'Theta 1, Nonlin. Sys.',0)
```

There is no need to print this plot; the plot is simply to verify that your linearization is correct. The resulting plots of the linear and non-linear system response should be very similar (overlapping?). If the responses are obviously different, you may have done something wrong during the linearization of the *Simulink* model. (Try again).

Experiment II: Data Collection & Simulink Model Comparison

The Pendubot is an expensive piece of precision equipment. Please do not throw the arm wildly during experimentation.

In this section, the actual dynamics of the Pendubot will be recorded for two different initial conditions (one large IC and the one small IC). The initial conditions that you give the Pendubot will be determined exactly, and they will be used as the initial conditions for the non-linear and linear *Simulink* models. After running the *Simulink* models, you will evaluate the ability of the mathematical model to reproduce the actual motion of the Pendubot.

- actth1 , the angle of the first linkage as a function of time, (which arbitrarily stands for *actual theta #1*) will be recorded by *Matlab* commands. Have your TA verify that the Pendubot is ready to record data.
- The first set of data to be collected is for small initial displacements of θ_1 (about 20 degrees) with no displacement for θ_2 (the second link will hang vertically at the start of data)

collection). The following command will start the data collection routine. Follow the on-screen instructions. Note: do not hold the Pendubot by the encoder when displacing θ_1 .

```
>>[acttime,actth1,a,b,c,d]=pend_collect(zeros(1000,1));
```

- Now verify the measurement by plotting the experimental data

```
>>plot(acttime,actth1);          (do not print this plot).
```

- Zoom in on the plot to determine the exact initial conditions (positions) for your experimental data. Note that initial velocities are zero. Also record the time at which the Pendubot was dropped. This will be referred to as the *time shift*.
- Run the linear and non-linear *Simulink* simulations using the initial condition for θ_1 that you just determined.
- *You may want to create an m-file for the following.* Do the following to plot your results. Title and label your plots accordingly. The `subplot(2,2,x)` command simply places four plots on one sheet of paper.

```
%CREATE A NEW FIGURE
figure(43)
%PLOT THE EXPERIMENTAL RESPONSE
subplot(2,2,1)
plot(acttime,actth1)
%!!add your xlabel, ylabel, and title here!!
%PLOT THE EXP. DATA WITH THE NON-LINEAR SIM.
subplot(2,2,2)
plot(acttime,actth1,'-','nlintime+timeshift,nlinth1,'--')
legend('Actual \theta_1','Nonlinear Sim. \theta_1',0)
%!!add your xlabel, ylabel, and title here!!
%PLOT THE EXP. DATA WITH THE LINEAR SIM.
subplot(2,2,4)
plot(acttime,actth1,'-','lintime+timeshift,linth1,'- -')
legend('Actual \theta_1','Linear Sim. \theta_1',0)
%!!add your xlabel, ylabel, and title here!!
%PLOT THE NON-LINEAR AND LINEAR SIMULATIONS TOGETHER
subplot(2,2,3)
plot(nlintime,nlinth1,'-','lintime,linth1,'- -')
legend('Nonlinear Simulation','Linear Simulation',0);
%!!add your xlabel, ylabel, and title here!!
%CHANGE THE PAPER ORIENTATION FOR PRINTING
orient landscape
```

- The first plot above is simply the data collected in the experiment. The next two plots compare the actual dynamics to the linear and nonlinear simulations. The fourth plot compares the linear and nonlinear simulations. Print this figure.
- Do the above steps of *Experiment II* again for a larger initial condition. θ_1 should be less than 90 degrees, but larger than that used in the first experiment. A good target is 60 to 70 degrees for θ_1 . Recall that θ_2 remains at zero.