UNIVERSITY OF BAHRAIN
COLLEGE OF ENGINEERING
DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

EENG 485: NONLINEAR CONTROL SYSTEM

LABORATORY MANUAL

- NONLINEAR CONTROL SYSTEM SIMULATION.
- VAN DER POL OSCILLATOR.
- LYAPUNOV POTENTIAL FUNCTION ANALYSIS (MAGNETIC LEVITATION).
- TWO NONLINEAR SYSTEMS AND LIMIT CYCLES.
- NONLINEAR CONTROL DESIGN (STATE FEEDBACK).
- MP CALCULATIONS FOR A LINEAR SYSTEM.
- VIA LYAPUNOV-BASED DESIGN AND MODEL REFERENCE ADAPTIVE CONTROLLER.

Prof. Ebrahim Mattar.

UNIVERSITY OF BAHRAIN
COLLEGE OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

NONLINEAR CONTROL SYSTEM
EENG 485

COMPUTER LAB NO.1
PRACTICAL: MAXIMUM OF (10) MARKS

NONLINEAR CONTROL SYSTEM SIMULATION

Objectives:

- To learn how to simulate a nonlinear system.
- To study in depth the time domain behavior of a nonlinear pendulum system.

Equipment:
- Laptop or Desktop Personal Computer.
- Printer.
- Matlab as simulation environment.

Steps:

The motion of a frictionless pendulum of length L is governed by the differential equation.

$$y'' = -\frac{g}{L}\sin y.$$

where y is the angle (in radians) that the pendulum makes with the vertical, g = 32:2ft=sec2, and L is the length of the pendulum. Although this a non-linear system, as you will see below, linear systems are very important in its analysis. This is an example of a technique called linearization" that will be studied at greater depth later in the class.

Assume that L = g so that the equation simplifies to :

$$y'' = -\sin y$$

This equation is equivalent with the system.

$$\begin{aligned} y' &= v \\ v' &= -\sin y \end{aligned}$$

(1) Show that the only equilibrium points for this system are (y; v) = (k_; 0) where k runs over the set of integers.

(2) Use pplane to construct the phase plane portrait for the system (2) using the range. $-4 \le y \le 10,\ -6 \le v \le 6.$

(2) For small y, siny _ y. Thus, we expect that for small y, the solutions to system (2) should have similar behavior to the solutions to the following linear system.

(3) To verify this, open a new MATLAB Command window by clicking again on the icon you use to start MATLAB. In this new window, start pplane and plot the phase plane portrait for system (3) using the same range on v and y as

in step 1, to demonstrate that the behavior of the linear system approximates the nonlinear one near (0; 0). According to the classification in you class notes, what kind of equilibrium does (0; 0) seem to be? Prove that you are right by finding the eigenvalues and eigenvectors. Get this plot printed.

Hint: You can get MATLAB to compute the eigenvalues and eigenvectors. Type \help eig" in the MATLAB Command widow for instructions.

(4) There is also an equilibrium at (π; 0). To analyze this equilibrium, let u = y − π. Then, $u_0 = y_0$ so system is equivalent with

$$u' = v$$
$$v' = -\sin(u + \pi) = \sin u$$

Reasoning similarly to previous steps, construct a linear system which approximates this system near u = 0. Plot its phase plane portrait in the second pplane window used in previous steps and classify the type of singularity according to the classification in your class notes. Prove your guess by computing the eigenvalues of the corresponding system. Get this plot printed. Then close this window.

(5) Using the pplane plot, in the pplane Solutions pull down menu, select \Find an Equilibrium Point". A cross will appear on the screen. Using the mouse, place the cross close to (0; 0) and click.

A large red dot will appear indicating the position of the equilibrium point. Note that the type of equilibrium points as well as the eigenvalues for the linear system are indicated. Similarly mark the equilibria at (π; 0), (2π; 0) and (3π; 0).

(6) In the pplane Solutions pull down menu, select \Plot Stable and Unstable Orbits". A cross will appear on the screen. Using the mouse, place the cross on the equilibrium point at (π; 0) and click.

What is produced is the orbits that are tangent to the eigenvectors of the linear approximation to the system (3) at the given point. Demonstrate this by plotting the eigenvectors. To plot the vector from [0; 0] to [a; b] enter plot([0,a],[0,b]).

The vectors should appear in the pplane display window. Indicate which of these orbits is stable and which is unstable. (Here, stable means that the orbit moves toward the equilibrium point and unstable means that it moves away from the equilibrium point.)

How, from the eigen-values, can you tell which is which?

(7) Plot the stable and unstable orbits for each of the remaining equilibrium, still using the same display window. You need not plot the other eigenvalues. The plots for the points (2kπ; 0) will be single points.

Remark:
The equilibria at the points ((2k+1)π; 0) in this system are all saddle points. At each saddle point for an autonomous system, there are four special orbits, called separatrices, whose tangent lines are parallel to the eigenvectors of the linear approximation. (Each one is a separatrix.)

(8) The reason for the term "separatrix" is that the separatrices separate the phase plane into regions where the solutions behave very differently. Demonstrate this by plotting the orbits corresponding to several initial conditions lying inside the regions bounded by the separatrices and several initial conditions lying outside these regions. Describe the motion of the pendulum for each type of initial condition.

UNIVERSITY OF BAHRAIN
COLLEGE OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

NONLINEAR CONTROL SYSTEM
EENG 485

LABORATORY NO.2
PRACTICAL: MAXIMUM OF (10) MARKS

VAN DER POL OSCILLATOR

Experiment Objectives:

- The main objective of this experiment is to make further analysis of the Van der Pol oscillator .
- To achieve few simulation of this dynamic system and compare them with some theoretical handwork.
- To study the limit cycle of oscillation and the associated phase plan analysis.

Equipment:
- Laptop or Desktop Personal Computer.
- Hand Calculator.
- Printer.

A Brief Introduction to The VAN DER POL OSCILLATOR

The labels mathematician, engineer, and physicist have all been used in reference to Balthasar van der Pol. The van der Pol oscillator, which we study in this notebook, is a model developed by him to describe the behaviour of nonlinear vacuum tube circuits in the relatively early days of the development of electronics technology. A little more detail on his work, taken from the Exploratorium web site (http://www.exploratorium.edu), is given below. A brief description of a circuit described by the van der Pol equation is given in Nonlinear Dynamics and Chaos, Steven Strogatz, Addison-Wesley 1994, p. 228. Chapter 7 of Strogatz' book contains a very readable discussion of the equation. Our study in this notebook will be based entirely on numerical solutions. The rigorous foundations for the analysis (e.g., the proof that the equation has a limit cycle solution which is a global attractor) date back to the work of Lienard in 1928, with later more general analysis by Levinson and others. Perturbation techniques are also useful for the case of large parameter, but we will not consider them here.

"Balthazar van der Pol was a Dutch electrical engineer who initiated modern experimental dynamics in the laboratory during the 1920's and 1930's. Van der Pol investigated electrical circuits employing vacuum tubes and found that they have stable oscillations, now called limit cycles. When these circuits are driven with a signal whose frequency is near that of the limit cycle, the resulting periodic response shifts its frequency to that of the driving signal. That is to say, the circuit becomes "entrained" to the driving signal. The waveform, or signal shape, however, can be quite complicated and contain a rich structure of harmonics and sub-harmonics. In the September 1927 issue of the British journal Nature, he and his colleague van der Mark reported that an "irregular noise" was heard at certain driving frequencies between the natural entrainment frequencies. By reconstructing his electronic tube circuit, we now know that they had discovered deterministic chaos. Their paper is probably one of the first experimental reports of chaos --- something that they failed to pursue in more detail. Van der Pol built a number of electronic circuit models of the human heart to study the range of stability of heart dynamics. His investigations with adding an external driving signal were analogous to the situation in which a real heart is driven by a pacemaker. He was interested in finding out, using his entrainment work, how to stabilize a heart's irregular beating or "arrhythmias"."

The van der Pol Equation

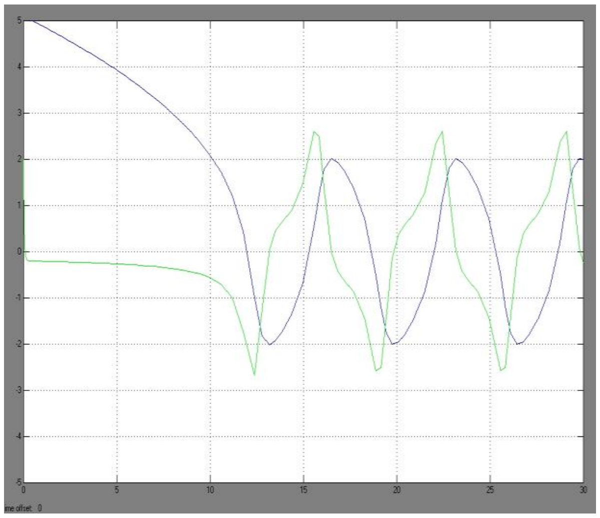
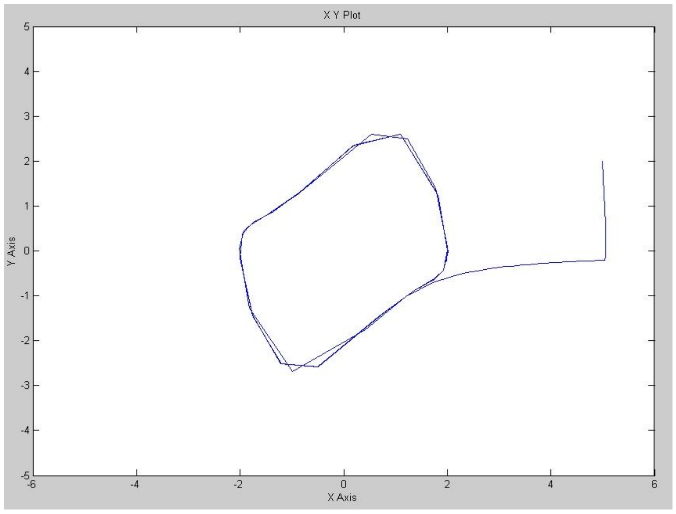
The van der Pol equation, in what is now considered to be standard form, is given by

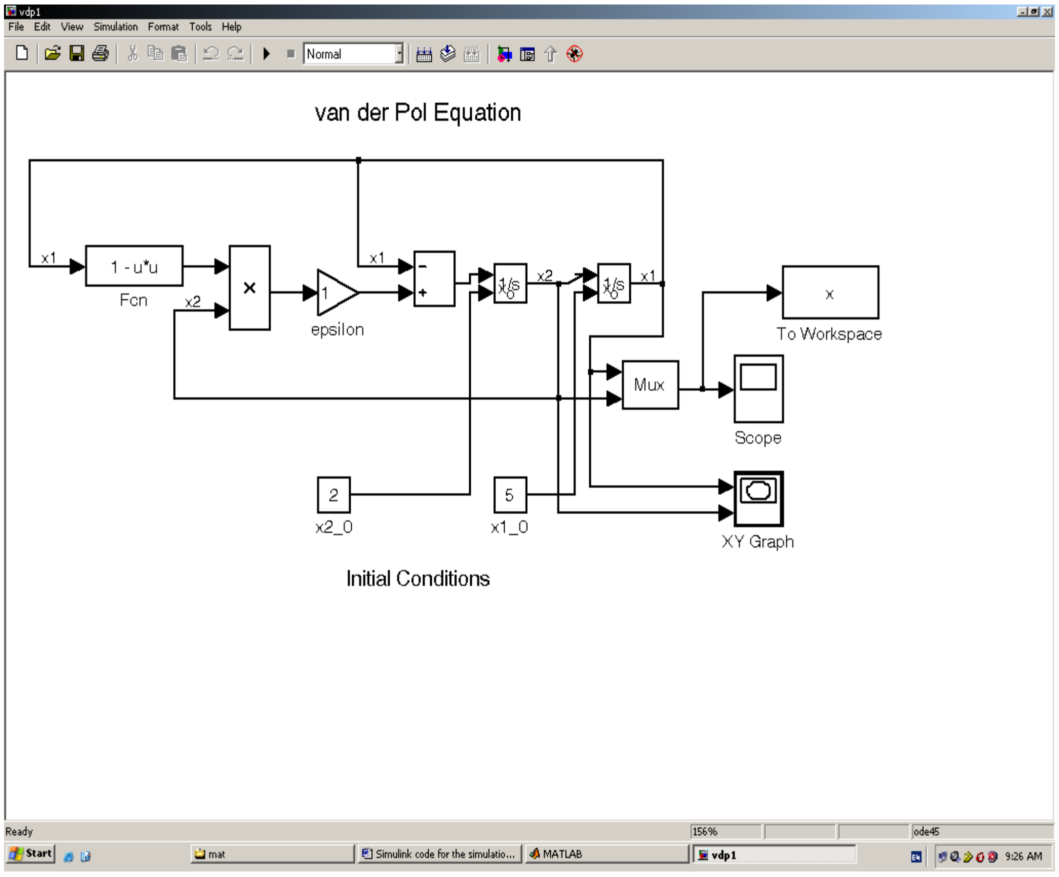
$$\ddot{x} + \mu(x^2 - 1)\dot{x} + x = 0 .$$

Steps:

Time response:

- Start Matlab on your PC and make sure you have created your own working sub-directory.
- In the Matlab environment, make sure the control toolbox has been installed, type help control.
- Simulink code for the simulation of the Van der Pol oscillator, vdp1.mdl .
- Run the simulink code for different initial conditions x_{1_0} and x_{2_0}.
- Drive the basic equation for the Van der Pol oscillator and prove the above results you got by simulation.
- Do any changes to the Van der Pol oscillator parameters … Observe the effect of such changes !



Equilibrium and Stability

There is an equilibrium at the origin, and it is obvious from the slope vector that there are no other equilibria. We give this equilibrium a name, and then look at the eigenvalues of the derivative matrix eqmat at the equilibrium.

```
eq = {0, 0};

eqmat = dermat /. Thread[statevec → eq]

{{0, 1}, {-1, μ}}

Eigenvalues[eqmat]
```

$$\left\{\frac{1}{2}\left(\mu - \sqrt{-4+\mu^2}\right),\ \frac{1}{2}\left(\mu + \sqrt{-4+\mu^2}\right)\right\}$$

We see that if $0 < \mu < 2$, the equilibrium is an unstable spiral. For $\mu > 2$, the equilibrium is an unstable node. Because there are no other equilibria, the only possible attractors for this system are the point at infinity, or periodic solutions. We will let the computer tell us which.

The Limit Cycle

We begin our numerical work with a phase portrait based on four selected initial conditions.

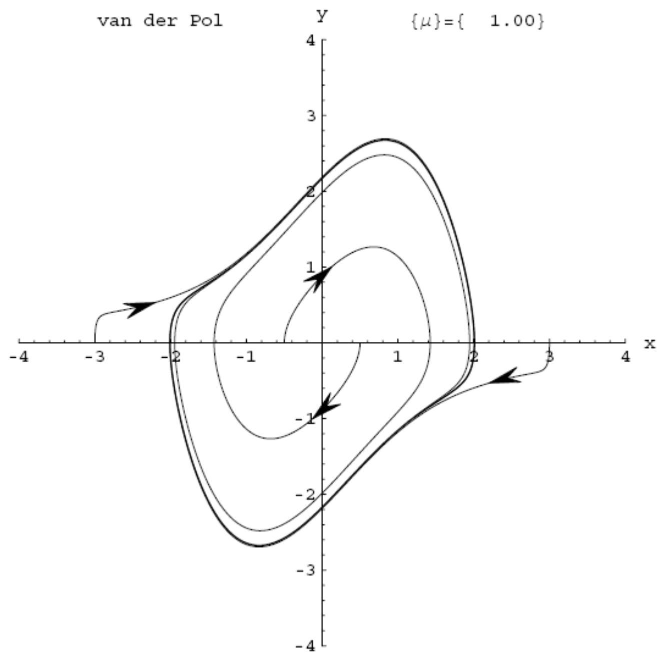
```
initset = {{0.5, 0}, {-0.5, 0}, {3.0, 0}, {-3.0, 0}};
```

We set the integration parameters.

```
t0 = 0.0; h = 0.02; nsteps = 500;
```

We set the plotting parameters.

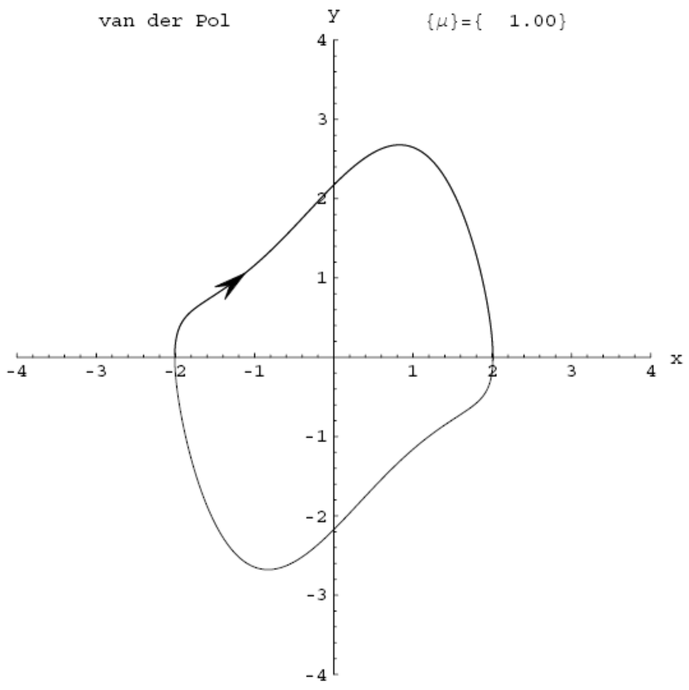
```
asprat = 1.0; plrange = {{-4, 4}, {-4, 4}}; labshift = 15;

arrowflag = True; arrowvec = {1 / 15};

graph1 = portrait[initset, t0, h, nsteps, 1, 2];
```



This looks very much like a globally attracting limit cycle. Let's try to construct the pure limit cycle.

```
sol2 = limcyc[{2, 0}, t0, h, nsteps];
```

```
graph2 = phaser[sol2];
```



```
period[sol2]
```

```
6.66
```

The remarkable fact about this oscillator is that every initial condition in the phase plane ultimately leads to this periodic motion with a period of 6.66. (Our calculations only suggest the truth of that statement. A proof requires a rigorous mathematical analysis which is given in some of the references mentioned above.)

Conclusions :

- What is the Van der Pol oscillator ? How it was originated ? Make few concluding remarks regarding this issue.
- Write a laboratory report describing your results and comment on the system behavior.
- What do think of the limit cycle ? and the System Phase Plan !! Make any Conclusions !

Prof. Ebrahim A. Mattar

UNIVERSITY OF BAHRAIN
COLLEGE OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

NONLINEAR CONTROL SYSTEM
EENG 485

LABORATORY NO.3
PRACTICAL: MAXIMUM OF (10) MARKS

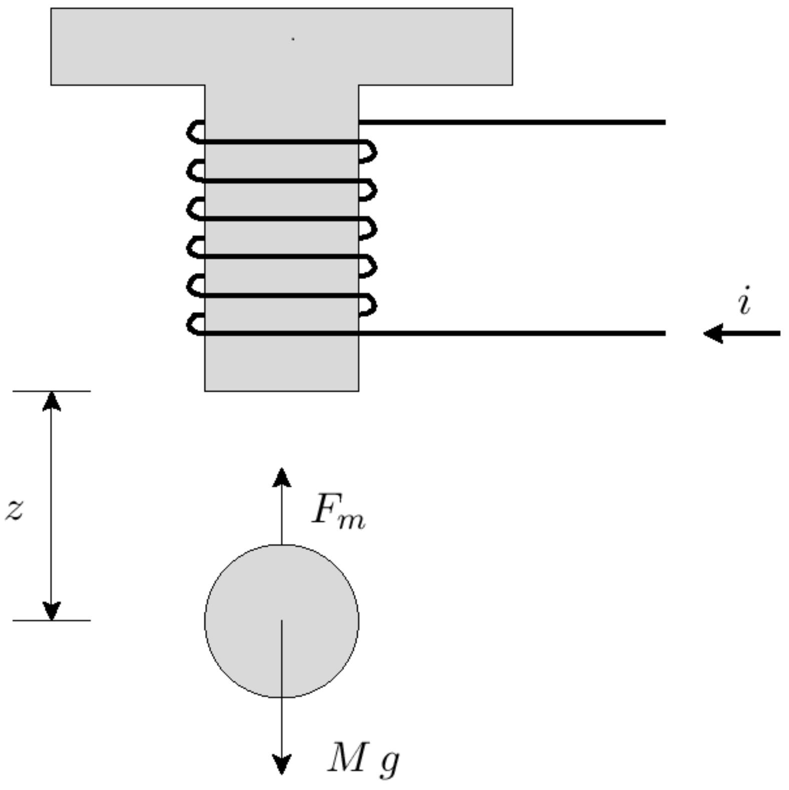
LYAPUNOV POTENTIAL FUNCTION ANALYSIS (MAGNETIC LEVITATION)

Objectives:

- To look in detail at the issue of Lyapunov Theorem for Global Asymptotic Stability.
- To simulate this via Matlab and do detailed analysis of the nonlinear dynamic system.

Equipment:

- Laptop or Desktop Personal Computer.
- Hand Calculator.
- Printer.



Quoted Figure from, "Magnetic Levitation Systems"

Steps :

Potential Function Computation:

For the given dynamic system investigate if the candidate potential Function V(x) is a Lyapunov function or not.

Figure below shows the diagram of a magnetic ball-suspension system, where the objective of the system is to control the position of the steel ball by adjusting the current.

- Advice any Potential function V(x).
- Investigate the stability.
- Simulate the nonlinear system behavior.
- Drive the transfer function of the system relating the input voltage e(t) with the steel ball position y(t).
- Compare the two results.

Time response Analysis:

- V(x) is the given candidate for the Lyapunov function. Hence what a potential function are you suggesting ?
- Make the simulink simulation for the given dynamics and show how the dynamic system will be behaving at different initial condition. (i.e to show that global stability cannot be guaranteed).
- Obtain the Radial Unbounded-ness for the magnetic ball-suspension. Example is shown below.

Servo System Stability (Via Lyapunov Candidate function)

- Go to the Position sevro and advise any potential function.
- Investigate the stability.
- Run the system and prove it via measurement of the servo output. (Transfer function of the servo system are found within the lab manuals)
- All experiments parts are found in the control laboratory.
- Show all your results.

Prof. Ebrahim A. Mattar

FRT 075 Nonlinear Control and Servo Systems
Laboration 2 – Nonlinear Control
The Furuta Pendulum

Bo Lincoln, Dept. of Automatic Control

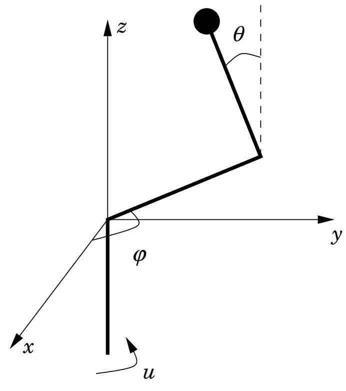


Figure 1 A schematic picture of the Furuta pendulum

1. Introduction

In this laboratory session we will design a nonlinear controller for an inverted pendulum on a rotating base (known as a “Furuta” pendulum). This is done by **Lyapunov design**, i.e. by inventing a Lyapunov function and making sure its derivative is negative by control. In the end of the lab there is also a short section on **friction compensation**, which makes a huge difference for our pendulum.

Important! There are 5 assignments in the lab. Number 2 and 3 has **home assignment** parts, which you will have to do before the lab.

The lab will be performed in Matlab Simulink, which you are now familiar with. We will both run pure simulations with a model of the pendulum in Simulink, and we will also run it with “hardware-in-the-loop”. The latter means that we will replace the pendulum model with a special Simulink block which communicates with the real pendulum in real time, and actually run our algorithms on the real pendulum.

The Matlab files you need to copy are `pendlib.mdl`, `pendbasic.mdl`, `estimatefriction.mdl`, and `pendinit.m`. Start Matlab and run `pendinit`.

2. The Furuta Pendulum

The Furuta pendulum is a free pendulum on a rotating arm, see Figure 1. We can control the torque for rotating the arm, and if it is done right, we can balance the pendulum in the upright position. To be able to do this, we need a correct mathematical model of the nonlinear dynamics of the pendulum.

2.1 Nonlinear Model

The dynamics of the Furuta pendulum are rather complicated due to the rotation of the pendulum. Some Lagrange theory “simplifies” matters, and after some algebra we end up with:

$$\begin{aligned}(J_p + Ml^2)(\ddot\theta - \dot\varphi^2 \sin\theta\cos\theta) + Mrl\ddot\varphi\cos\theta - gl(M+m/2)\sin\theta = 0\\ Mrl\ddot\theta\cos\theta - Mrl\dot\theta^2\sin\theta + 2(J_p + ml^2)\dot\theta\dot\varphi\sin\theta\cos\theta\\ +(J + mr^2 + Mr^2 + (J_p + ml^2)\sin^2\theta)\ddot\varphi = u.\end{aligned} \tag{1}$$

To make it more readable, we introduce

$$\begin{aligned} a &= J_p + Ml^2 & b &= J + Mr^2 + mr^2\\ c &= Mrl & d &= lg(M + m/2)\end{aligned}$$

and the equations of motion can be rewritten:

$$\begin{aligned} a\ddot\theta - a\dot\varphi^2\sin\theta\cos\theta + c\ddot\varphi\cos\theta - d\sin\theta = 0\\ c\ddot\theta\cos\theta - c\dot\theta^2\sin\theta + 2a\dot\theta\dot\varphi\sin\theta\cos\theta + (b + a\sin^2\theta)\ddot\varphi = u,\end{aligned} \tag{2}$$

where θ is the angle of the pendulum, and φ is the angle of the arm. We control u, the motor torque on the arm.

Approximate coefficients for the pendulum used in the laboration are:

$$\begin{aligned} l &= 0.413\ m & r &= 0.235\ m\\ M &= 0.01\ kg & J &= 0.05\ kgm^2\\ J_p &= 0.0009\ kgm^2 & m &= 0.02\ kg\end{aligned}$$

2.2 Linearized Model

To be able to stabilize the pendulum in the upright position, we need a controller. The most commonly used method to design a controller is to linearize the nonlinear model and form at linear controller for this model.

To do this, we introduce the state

$$x = \begin{pmatrix}\theta & \dot\theta & \varphi & \dot\varphi\end{pmatrix}$$

and linearization of the system (2) around

$$x = \begin{pmatrix}0 & 0 & 0 & 0\end{pmatrix},$$

which is the upright position of the pendulum with zero velocity, gives

$$\dot x = Ax + Bu = \begin{pmatrix} 0 & 1 & 0 & 0\\ \frac{bd}{ab-c^2} & 0 & 0 & 0\\ 0 & 0 & 0 & 1\\ \frac{-cd}{ab-c^2} & 0 & 0 & 0\end{pmatrix} x + \begin{pmatrix} 0\\ \frac{-c}{ab-c^2}\\ 0\\ \frac{a}{ab-c^2}\end{pmatrix} u. \tag{3}$$

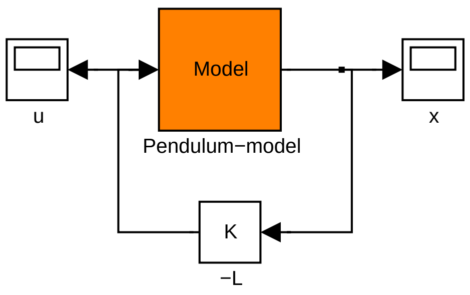


Figure 2 The set-up to simulate your linear controller.

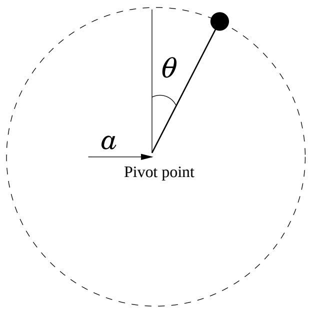


Figure 3 The planar pendulum, with only two states. The dashed circle is the possible "orbit" for the pendulum mass.

ASSIGNMENT 1
Calculate the linearized model in Matlab (i.e. calculate the A and B matrices). Show that the linear feedback

$$u = -Lx$$

with $L = \begin{pmatrix} -7.5343 & -1.3465 & 0 & -0.2216 \end{pmatrix}$ almost stabilizes the linear model. Which state is not stabilized?
Open Simulink and get the "Pendulum-model" block from `pendlib.mdl`. Insert this controller and simulate it (see Figure 2). Does it work? For which initial states? Why does it not work for all initial states? (Oscillations around zero are due to friction.) □

3. Swinging Up the Pendulum

Now that we have a stabilizing controller for the upright position, we would like to desing a controller which gets the pendulum close enough for our linear controller to work (a "swing-up"-controller). To be able to solve the problem, we will approximate the Furuta pendulum model by a planar pendulum (see Figure 3).

The equations of motion now simplify to

$$\ddot{\theta} = \frac{Mgl}{J_p}\sin\theta - \frac{Ml}{J_p}a\cos\theta, \tag{4}$$

where a is our control signal. It is the acceleration sideways of the pivot point. We can create this acceleration with our usual control signal u, i.e. the torque to the pendulum arm.
All parameters of the pendulum can actually be described by one parameter, which we call ω_0:

$$\omega_0 = \sqrt{\frac{Mgl}{J_p}}$$

and the equation simplifies to

$$\ddot{\theta} = \omega_0^2 \sin\theta - \omega_0^2 \frac{a}{g}\cos\theta$$

ASSIGNMENT 2
Home assignment: The name ω_0 suggests that it denotes a frequency. Show why, and suggest a way to estimate this parameter from experiments. *Hint:* Keep $a = 0$ and linearize around $\theta = \pi$ (down position).
At the lab: Do the experiment on the Pendulum-model in Simulink or on the real pendulum. Which value of ω_0 did you get? □

3.1 Energy Control

A very useful method to get the pendulum to the upright position is to control its *energy* to be the energy of the top position. The total energy (potential energy + kinetic energy) of the pendulum is:

$$E = Mgl(\cos\theta - 1) + \frac{J_p}{2}\dot{\theta}^2$$

or normalized by ω_0^2

$$E_n = \cos\theta - 1 + \frac{1}{2\omega_0^2}\dot{\theta}^2$$

If the pendulum has energy $E_n = 0$, it will either be moving along the orbit in Figure 3 or be at the top position, with zero velocity (check this by inserting $\theta = 0$, $\dot{\theta} = 0$). This is due to the fact that no energy leaves the pendulum in our model. When the pendulum is close to the “up-position” it will have low speed, and it is very easy for the linear controller to “catch” it.

ASSIGNMENT 3
Home assignment: We want the energy $E_n \to 0$ both if the pendulum has too much or too little energy. Let $V(x,a) = \big(E_n(x,a)\big)^2$ be a Lyapunov function candidate. Find a simple controller $a = F(x)$ such that $\frac{d}{dt}V \leq 0$ for all x.
Hints:

- Differentiate and simplify V first, and see how you can choose a to make the derivative negative.
- Don't use control signals larger than 1.
- Make the controller non-linear!

At the lab: Simulate the controller with the Pendulum-model in Simulink. Insert the "Theta, thetadot observer"-block in front of your controller. It is needed to give good measurements of θ and $\dot{\theta}$ for the full 360° motion.
If it does not work properly, modify it until it does. □

4. The Full Monty

Now it is time to put it all together. We want the controller to have two modes:

1. Swinging up the pendulum (your controller).
2. Catching and balancing it in the top position.

In the Simulink model `pendbasic.mdl` a working top-position-controller is already given.

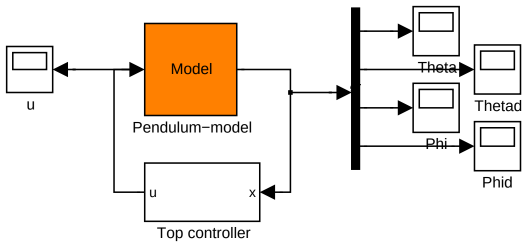


Figure 4 The Simulink setup with pendulum model and a top-position controller.

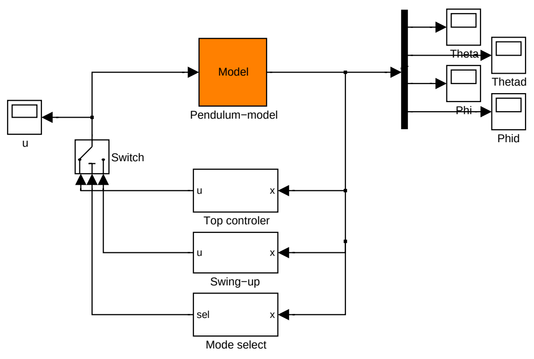


Figure 5 The full pendulum controller.

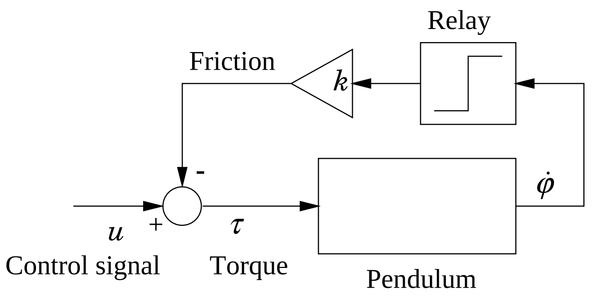


Figure 6 A model of the φ friction. The friction will work in the opposite direction of the velocity $\dot{\varphi}$. The constant k is the friction torque.

ASSIGNMENT 4
Construct (in Simulink) a switching method, which switches between your swing-up controller and the given top-position controller. See Figure 5 for a useful structure. *Hint:* For example, switch to top controller when θ and $\dot{\theta}$ are both "small", i.e. close to the top position with low velocity.
When it works with the Pendulum-model, go to the real pendulum and run your controller there. Modify it until it works. □

5. Friction Compensation

When the pendulum is in its upright position, it oscillates in φ (the arm position). This is due to friction which makes the arm stick although $u \neq 0$. A simple way of modelling the friction is shown in Figure 6.

A simple but efficient way of compensating for this is to estimate k offline ($\hat{k}$) and give the control signal a extra "push" in the right direction depending on $\dot{\varphi}$ – see Figre 7.

In the Simulink model `estimatefriction.mdl` (see Figure 8) a simple PI-controller has been implemented to keep the arm velocity $\dot{\varphi}$ constant (with the pendulum hanging down).

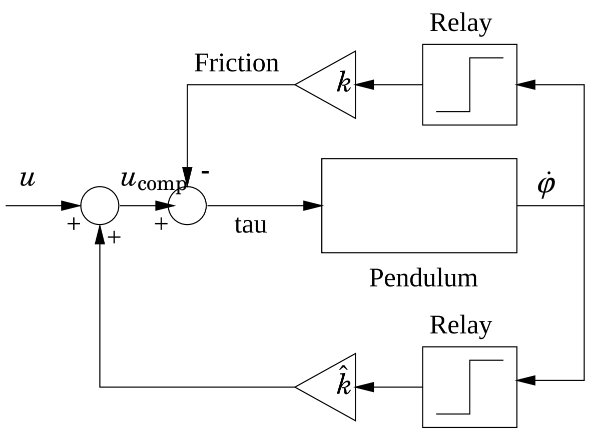


Figure 7 A simple friction compensation where an extra torque $\hat{k}$ is added in the right direction.

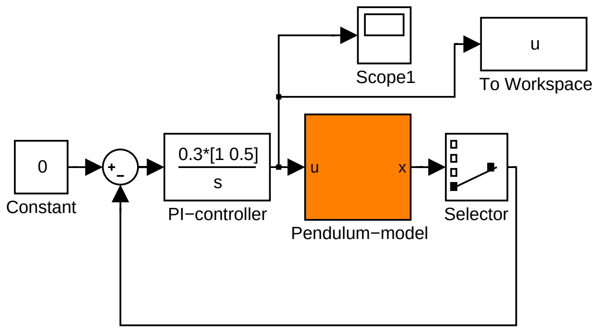


Figure 8 The Simulink setup to estimate friction.

ASSIGNMENT 5
How can you use this to get an estimate $\hat{k}$ of the friction torque? Estimate it for the pendulum-model, and if there is time, on the real pendulum.
Insert the “Friction-compensation” block from `pendlib.mdl` into your controller. Insert the estimate $\hat{k}$ and simulate. Does it perform better in the top position?
If there is time, run it on the real pendulum too. □

UNIVERSITY OF BAHRAIN
COLLEGE OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

NONLINEAR CONTROL SYSTEM
EENG 485

LABORATORY NO. 4
PRACTICAL: MAXIMUM OF (10) MARKS

TWO NONLINEAR SYSTEMS AND LIMIT CYCLES

Objectives :

- The work that will be done is mainly concerned with two nonlinear systems describing functions and limit cycles.
- Over the nonlinear Control System, a student is supposed to gain some knowledge and experience with using and analyzing:

  – Nyquist Plots
  – Phase Diagrams
  – Time Histories
  – The Describing Function Method
  – MATLAB, a tool to achieve all the needed analysis.

Expected Lessons Learned from the Experiment:

- Have some understanding of how feedback control works and methods to deal with nonlinearities.
- The student should have the ability to use tools such as Nyquist plots, phase diagrams, time histories, the describing function method, and Matlab to study nonlinear systems.

- There is much more to learn about linear and nonlinear systems such as:
  – How good an approximation is the describing function?
  – Does the concavity of the describing function affect limit cycle stability?
  – What other methods are available for studying nonlinear systems?
  – What other information does the Nyquist plot provide about a system?
  – How can a student write equations that govern a system if he only knew its input(s) and output(s)?
  – How does a systems engineer make sure a subsystem works in such a way that does not adversely affect the
  whole system?

Equipment:
- Laptop or Desktop Personal Computer.
- Hand Calculator.
- Printer.

Problem Setup
Systems under Study : We are going to analyze two systems as:

Van der Pol Oscillator

A model of an electronic circuit involving vacuum tubes which acts like a normal resistor when current is high, and a negative resistor when the current is low.
Circuit amplifies small oscillations but reduces large ones.
Relatively simple nonlinear system.

Super-Cavitating Body

Body is enveloped in a cavity while it travels underwater.
Cavitation allows the body to travel at faster than normal speeds.
Feedback control is needed to make sure body stays on course.
Nonlinearity occurs due to the forces which act on the body when it leaves the cavity.

Why A Student is Studying These Systems?

- Would like to control these systems so that they have stable dynamics.
- If you disturb the system from its equilibrium values, you want it to be able to return to that state.
- A limit cycle is an <u>isolated periodic orbit around one</u> or more equilibrium points.
- Hence , if you can find stable limit cycles and stable equilibrium points, a student is closer to the goal of having a stable system.

<u>Example (1) : Van der Pol Oscillator</u>

How You Can find Limit Cycles?

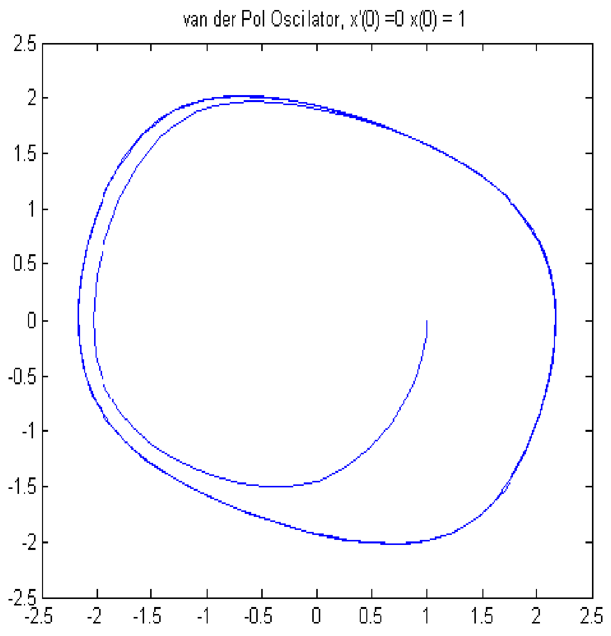
The Van der Pol Oscillator is described by this equation:

$$\ddot{x} = \varepsilon\left(\dot{x} - \left(\frac{1}{3}\dot{x}\right)^3\right) - x$$

A student is needed to write a <u>MATLAB</u> program which solved the equation for x and x' which then plotted x' versus x.   The phase plot shows that there is a limit cycle.   Hence, you should get same results as shown below:

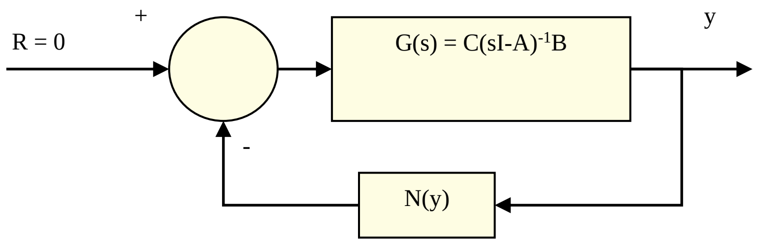
```
clear;

[x1,x2] = ode45('vdp1b',[0 80],[1; 0]);

[z1,z2] = ode45('vdp1c',[0 80],[1; 0]);

[y1,y2] = ode45('vdp1',[0 40],[1; 0]);

figure(1);
subplot(1,3,1)
plot(x2(:,1),x2(:,2));
title('\epsilon = 0.1');
%figure(2);
subplot(1,3,2)
plot(y2(:,1),y2(:,2));
title('\epsilon = 1');
%figure(3);
subplot(1,3,3)
plot(z2(:,1),z2(:,2));
title('\epsilon = 10');
```

```
function dy = vdp1(t,y)
dy = [y(2); (y(2) - (1/3)*y(2)^3 ) - y(1)];
```



The Describing Function Method

Here, the original system's output is nonlinear for some reason, in the case of the SCB, due to the planning force of the water when it exits the cavity. Describing Function Method is an analysis technique that allows a user to think of nonlinearities as a feedback to a dominant linear system.



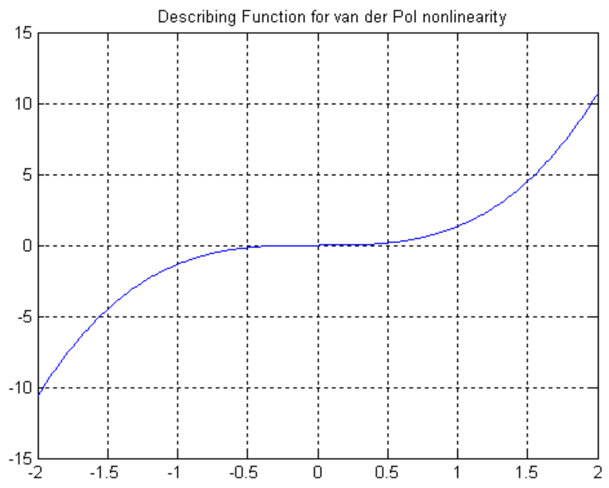
How to Get the Van der Pol Limit Cycle?

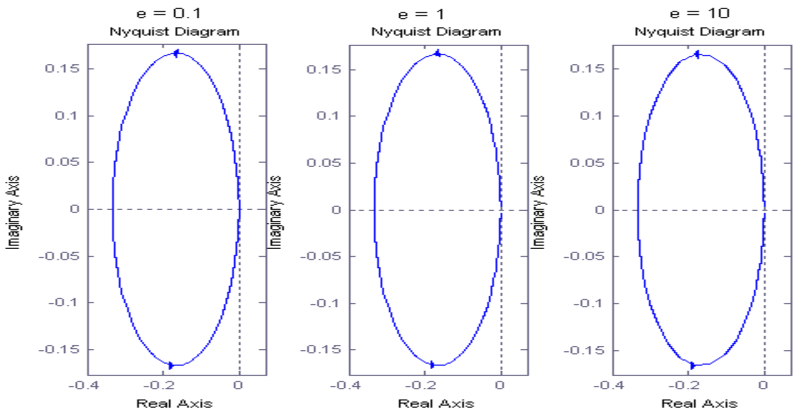
The student needs to write a program that calculates the describing function for the nonlinear term. The describing function's intersection with the transfer function on the Nyquist plot tells that a limit cycle exists at that point, as well as the amplitude and frequency of the limit cycle oscillation. Using the general Nyquist stability criterion, the limit cycle can be determined to be stable or unstable.

This is can be achieved via the following Matlab code:

```
clear

count = 1;

for a = -2:.01:2;
    xvar(count) = a;
    [t, x] = ode45(@nlvan, [0 pi], 0, [], a);
    yvar(count) = x(length(x));

    count = count + 1;
end

figure(1);
subplot(1,3,2);
plot(xvar,yvar);
grid on;
```

```
function result = nlvan(t, x, a);
result = [(a*sin(t))^3];
```





Van der Pol Limit Cycle

Next, the student is needed to put the equation in linear state variable form, so you could make use of the feedback model: Solve for the transfer function and from the transfer function, get the Nyquist plot. (Make use of the followings).

$$x = \begin{bmatrix} x1 \\ x2 \end{bmatrix}$$

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -1 & \varepsilon \end{bmatrix} x + \begin{bmatrix} 0 \\ 1/3\varepsilon \end{bmatrix} u$$

$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} x$$

$$u = -N(y)$$

$$G(s) = C(sI - A)^{-1} B$$

$$G(s) = \begin{bmatrix} 0 & 1 \end{bmatrix} \left(\begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} + \begin{bmatrix} 0 & -1 \\ 1 & -\varepsilon \end{bmatrix} \right)^{-1} \begin{bmatrix} 0 \\ 1/3\varepsilon \end{bmatrix}$$

$$G(s) = \frac{\varepsilon s}{3s^2 - 3\varepsilon s + 3}$$

So have you seen the describing function?

<u>Example (2) : Super-Cavitating Body</u>

The second system to be considered is the Super-Cavitating Body (SCB). The SCB model, in state variable form is:

$$\begin{bmatrix} \dot{z} \\ \dot{w} \\ \dot{\theta} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 & 1 & -V & 0 \\ 0 & a_{22} & 0 & a_{24} \\ 0 & 0 & 0 & 1 \\ 0 & a_{42} & 0 & a_{44} \end{bmatrix} \begin{bmatrix} z \\ w \\ \theta \\ q \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ b_{21} & b_{22} \\ 0 & 0 \\ b_{41} & b_{42} \end{bmatrix} \begin{bmatrix} \delta_e \\ \delta_c \end{bmatrix} + \begin{bmatrix} 0 \\ c_2 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ d_2 \\ 0 \\ d_4 \end{bmatrix} u$$

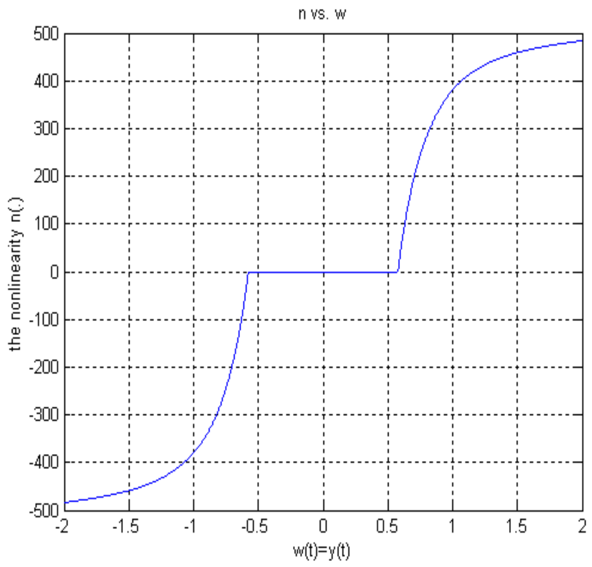
z :	depth
w :	transverse displacement rate
θ :	pitch angle
q :	pitch rate

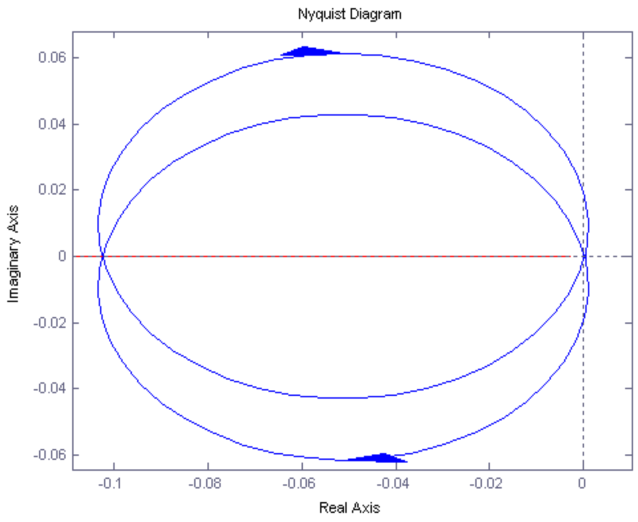
where the Nonlinearity is governed by the following identities :

$$n(w) = V^2 \left(1 - \left(\frac{R'}{h' + R'}\right)^2\right) \frac{1+h'}{1+2h'} \alpha$$

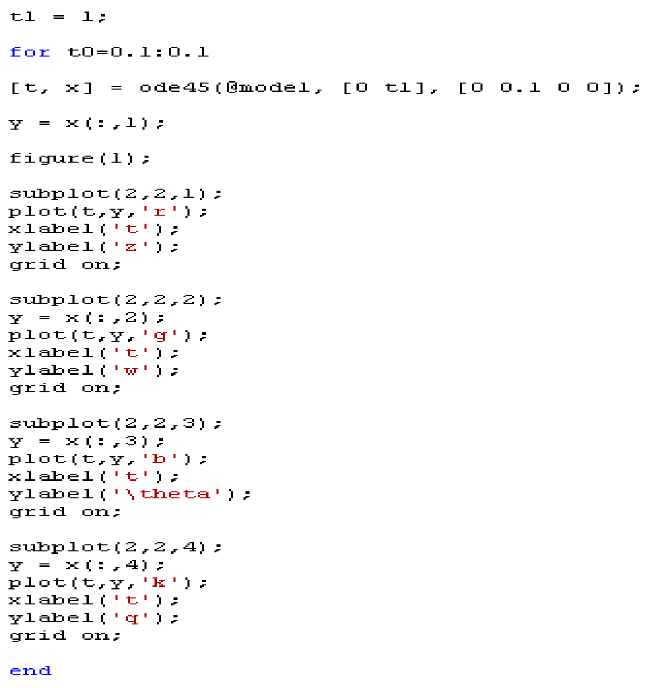
$$h' = \begin{cases} 0 & \frac{L}{R}\left|\frac{w}{V}\right| < R' \\ \frac{L}{R}\left|\frac{w}{V}\right| - R' & \text{otherwise} \end{cases} \quad \text{and} \quad \alpha = \begin{cases} \frac{w}{V} - \frac{\dot{R}_c}{V} & \frac{w}{V} > 0 \\ \frac{w}{V} + \frac{\dot{R}_c}{V} & \text{otherwise} \end{cases}$$

Again after your analysis, you should get the describing function as follows, and the associated Nyquist plot.

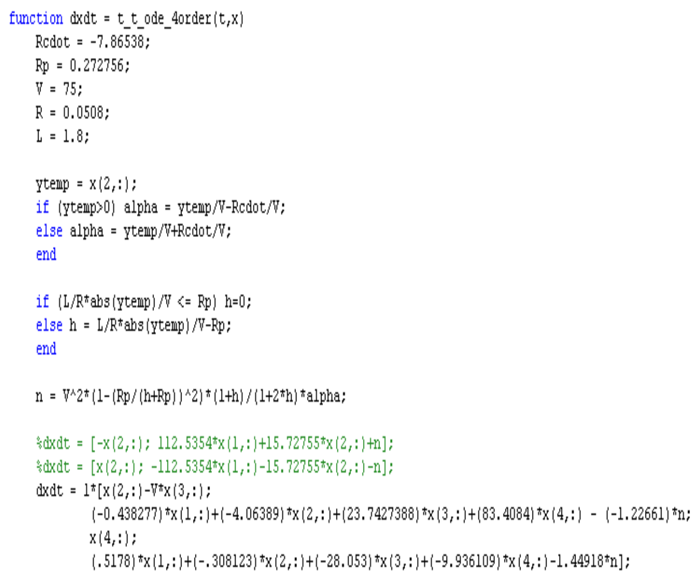




Below are plots of the transfer function (in the complex plane) and describing function for the SCB.   Intersections of the two are points about which we apply the general Nyquist criterion to determine the stability of limit cycles.   When stable limit cycles could not be found for the system using both control inputs, I wrote a program which simulated the system and plotted the state variables as a function of time.

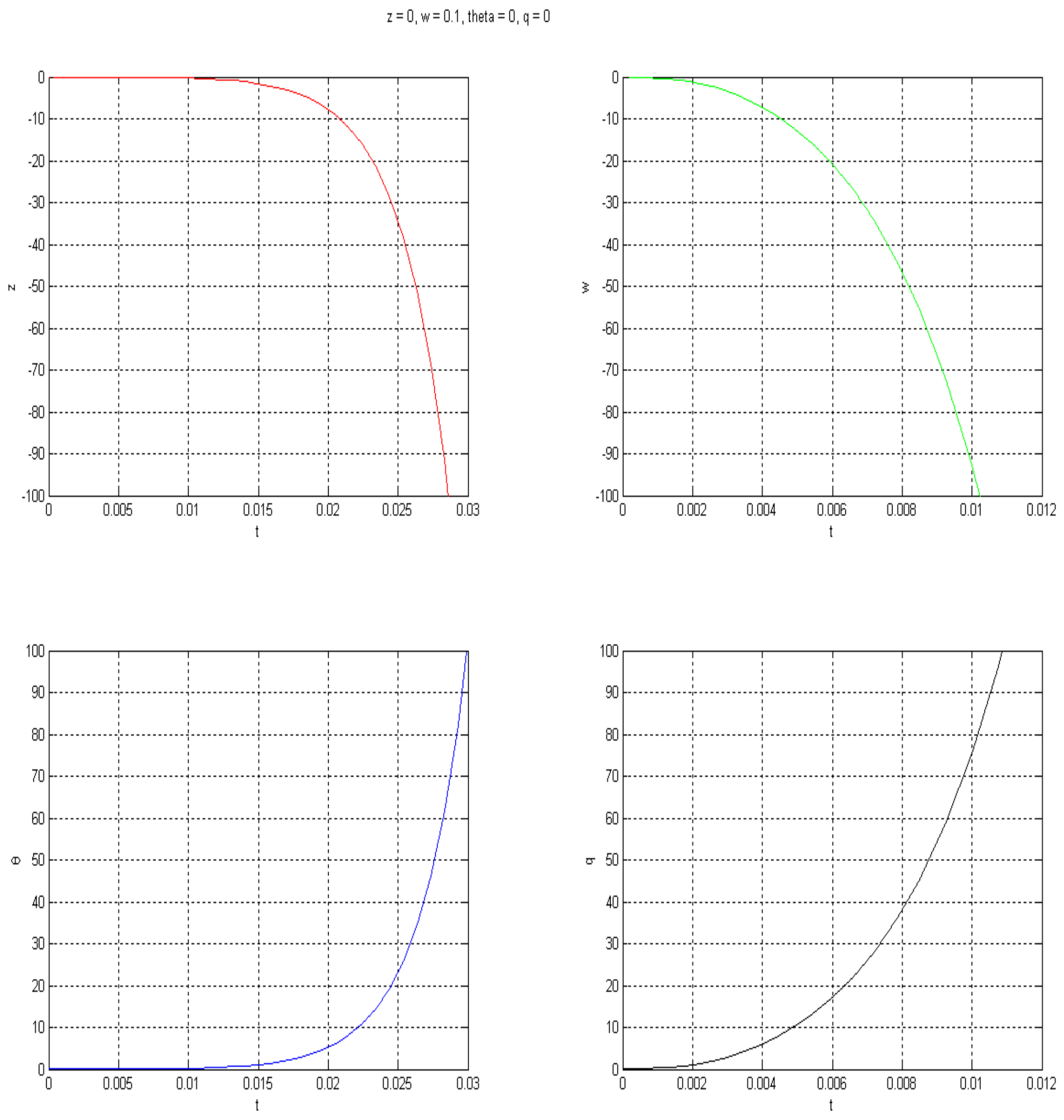


```
t1 = 1;

for t0=0.1:0.1

[t, x] = ode45(@model, [0 t1], [0 0.1 0 0]);

y = x(:,1);

figure(1);

subplot(2,2,1);
plot(t,y,'r');
xlabel('t');
ylabel('z');
grid on;

subplot(2,2,2);
y = x(:,2);
plot(t,y,'g');
xlabel('t');
ylabel('w');
grid on;

subplot(2,2,3);
y = x(:,3);
plot(t,y,'b');
xlabel('t');
ylabel('\theta');
grid on;

subplot(2,2,4);
y = x(:,4);
plot(t,y,'k');
xlabel('t');
ylabel('q');
grid on;

end
```



```
function dxdt = t_t_ode_4order(t,x)
    Rcdot = -7.86538;
    Rp = 0.272756;
    V = 75;
    R = 0.0508;
    L = 1.8;

    ytemp = x(2,:);
    if (ytemp>0) alpha = ytemp/V-Rcdot/V;
    else alpha = ytemp/V+Rcdot/V;
    end

    if (L/R*abs(ytemp)/V <= Rp) h=0;
    else h = L/R*abs(ytemp)/V-Rp;
    end

    n = V^2*(1-(Rp/(h+Rp))^2)*(1+h)/(1+2*h)*alpha;

    %dxdt = [-x(2,:); 112.5354*x(1,:)+15.72755*x(2,:)+n];
    %dxdt = [x(2,:); -112.5354*x(1,:)-15.72755*x(2,:)-n];
    dxdt = 1*[x(2,:)-V*x(3,:);
            (-0.438277)*x(1,:)+(-4.06389)*x(2,:)+(23.7427388)*x(3,:)+(83.4084)*x(4,:) - (-1.22661)*n;
            x(4,:);
            (.5178)*x(1,:)+(-.308123)*x(2,:)+(-28.053)*x(3,:)+(-9.936109)*x(4,:)-1.44918*n];
```

The program calculates the value of the describing function at each time interval and passes that value to the input of the feedback system.



Next, we shall took a look at the system with only one control input. Again, the system was stabilized. Using MATLAB, you need to solve the system of nonlinear equations for the K matrix and from there, calculated the values of the coefficients for the transfer function. This would allow you to get the Nyquist plot, however, you will still not be able to find any stable limit cycles

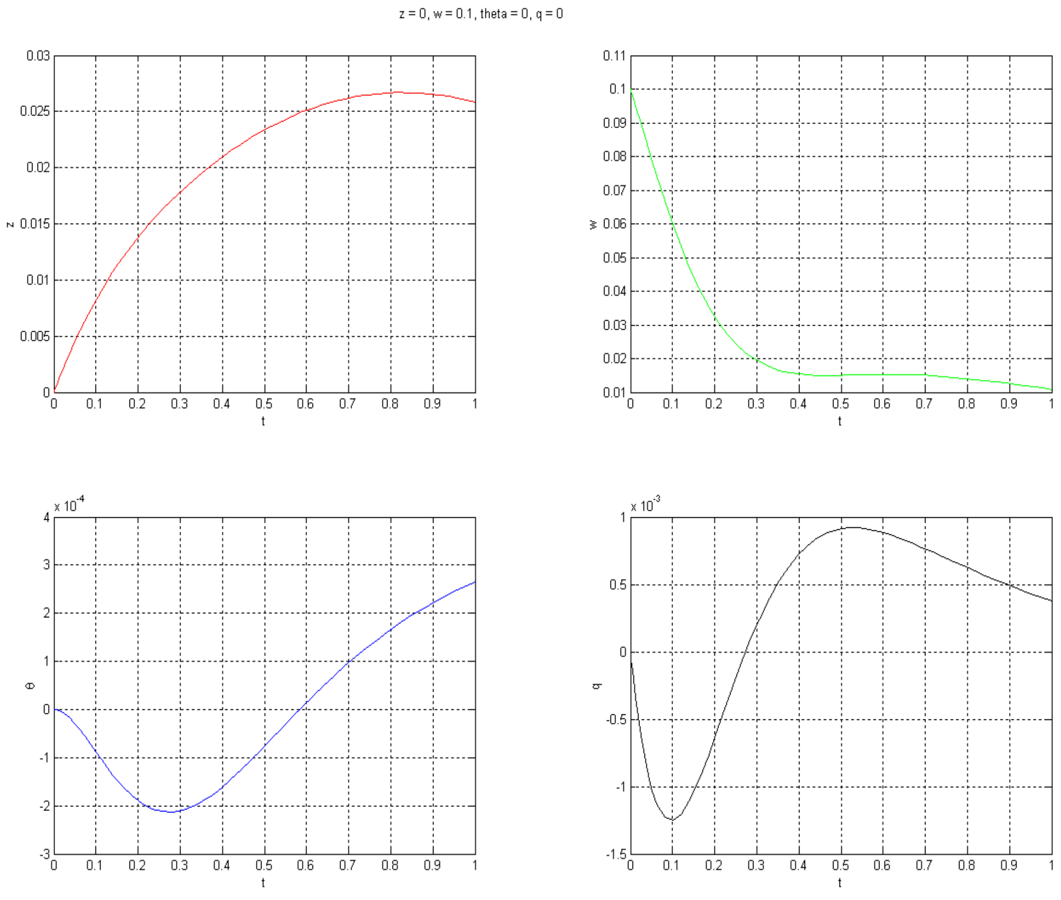
$$\begin{bmatrix} \delta_e \\ \delta_c \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ k_{21} & k_{22} & k_{23} & k_{24} \end{bmatrix} \begin{bmatrix} z \\ w \\ \theta \\ q \end{bmatrix}$$

$$\begin{bmatrix} \dot{z} \\ \dot{w} \\ \dot{\theta} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 & 1 & -V & 0 \\ a_{21}' & a_{22}' & a_{23}' & a_{24}' \\ 0 & 0 & 0 & 1 \\ a_{41}' & a_{42}' & a_{43}' & a_{44}' \end{bmatrix} \begin{bmatrix} z \\ w \\ \theta \\ q \end{bmatrix} + \begin{bmatrix} 0 \\ c_2 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ d_2 \\ 0 \\ d_4 \end{bmatrix} (-n) \text{ where}$$

$$a_{21}' = b_{22}k_{21},\ a_{22}' = a_{22} + b_{22}k_{22},\ a_{23}' = b_{22}k_{23},\ a_{24}' = a_{24} + b_{22}k_{24},$$
$$a_{41}' = b_{42}k_{21},\ a_{42}' = a_{42} + b_{42}k_{22},\ a_{43}' = b_{42}k_{23},\ a_{44}' = a_{44} + b_{42}k_{24}$$

$$G(s) = C(sI - A)^{-1}B + D = \frac{d_2 s^3 + \beta_2 s^2 + \beta_1 s + \beta_0}{s^4 + \gamma_3 s^3 + \gamma_2 s^2 + \gamma_1 s + \gamma_0}\ where$$
$$\beta_2 = d_4 a_{24}' - d_2 a_{44}',\ \beta_1 = d_4 a_{23}' - d_2 a_{43}',\ \beta_0 = d_2 a_{41}' - d_4 a_{21}',$$
$$\gamma_3 = -a_{22}' - a_{44}',\ \gamma_2 = a_{22}' a_{44}' - a_{43}' - a_{24}' a_{42}',\ \gamma_1 = -a_{21}' + a_{43}' a_{22}' - a_{23}' a_{42}' + V a_{41}'$$
$$\gamma_0 = a_{21}' a_{44}' - a_{24}' a_{41}' + a_{21}' a_{43}' - a_{23}' a_{41}' + V a_{21}' a_{42}' - V a_{41}' a_{22}'$$

Again, a simulation of the system is to be done. This would yield interesting results!! Now, instead of solving for the K matrix, just pick its values and in that manner try to find a stable limit cycle.



Prof. Ebrahim A. Mattar

UNIVERSITY OF BAHRAIN
COLLEGE OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

NONLINEAR CONTROL SYSTEM
EENG 485

LABORATORY NO.5
PRACTICAL: MAXIMUM OF (10) MARKS

NONLINEAR CONTROL DESIGN (STATE FEEDBACK)

Objectives:

- To investigate the characteristics of nonlinear system design via a defined Lyapunov function.
- To design a state feedback controller, hence to simulate this via matlab and do detailed analysis of the nonlinear dynamic system.

Equipment:
- Laptop or Desktop Personal Computer.
- Hand Calculator.
- Printer.

Steps :

- Follow the given below steps:
  A nonlinear plant is described by.

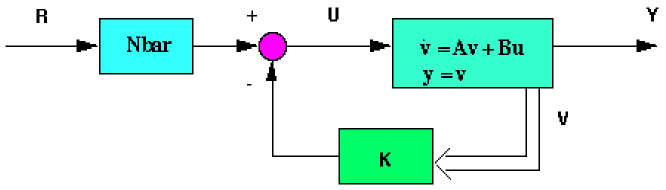
$$\begin{aligned}\dot{x}_1(t) &= -x_1(t) - x_2(t) + x_3(t) + u(t)\\ \dot{x}_2(t) &= x_1(t)\\ \dot{x}_3(t) &= x_2(t)\end{aligned}$$

- Check the Equilibrium points.
- Simulate the system with no input. Check the beaviour.
- For the following input signal:

$$u(t) = x_1(t) - x_3(t) - (2x_1(t) + x_3(t))^3$$

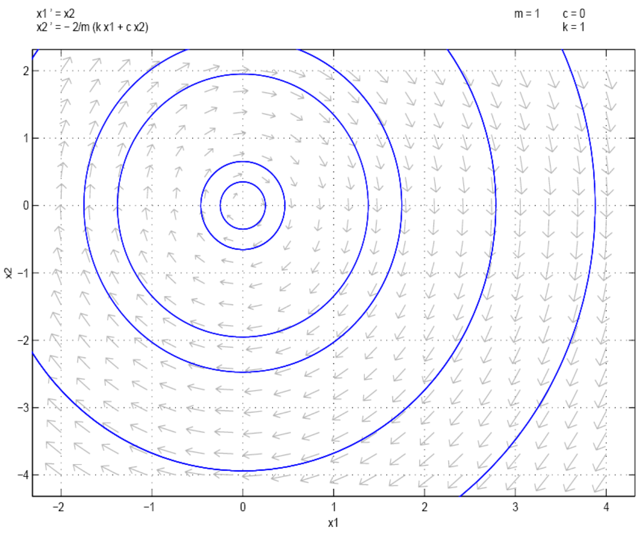
- Compute via the Use of the following Lyapunov function.

  - Via the computation of the eigen values of the linearized A matrix.
  - Compute the value of Am
  - Construct a state feedback, observe the controller behavior of the system accordingly.
  - Get the phase portrait for the controlled system.



(a) [6p] Show that $x = 0$ is a global asymptotically stable equilibrium point for the closed loop system.
(*Hint:* Use
$$V(x) = x_1^2 + \frac{1}{2}x_2^2 + \frac{1}{2}x_3^2 + x_1 x_3 \qquad)$$

(b) [4p] In order to evaluate the gain margin of the closed loop system, study the control law
$$u(t) = A_m(x_1(t) - x_3(t) - (2x_1(t) + x_3(t))^3)$$
where $A_m \geq 0$ is a gain. Decide for which A_m the closed loop system is stable or unstable.
(*Hint:* Study the stability properties of the linearized closed loop system.)



Observations :

- Hence design a state feedback and show the stability of the designed system.
- Do think the controller has achieved a better performance ? Justify this.
- Measure the associated rise time and the overshoot % .

Conclusion :

This laboratory has showed us that the dynamics of a linear system of differential equations are completely determined by the eigenvalues of the coefficient matrix. The three possible forms are described, and examples are given. Lastly, we show how to compute the parameters of our controller model based upon the dynamics exhibited by the phase portraits.

UNIVERSITY OF BAHRAIN
COLLEGE OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

NONLINEAR CONTROL SYSTEM
EENG 485

LABORATORY NO.6
PRACTICAL: MAXIMUM OF (10) MARKS

MP CALCULATIONS FOR A LINEAR SYSTEM

Objectives:

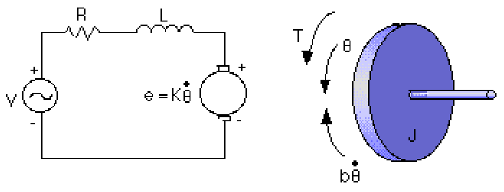
- To investigate the Nyquist design via the use of M circle. This gives a relation to the small gain theory for the nonlinear case.
- To simulate this via matlab and do detailed analysis of the nonlinear dynamic system.

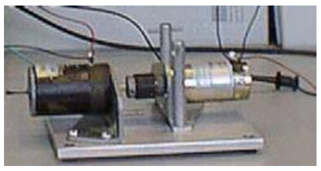
Equipment:

- Laptop or Desktop Personal Computer.
- Printer.

Problem Setup

A common actuator in control systems is the DC motor. It directly provides rotary motion and, coupled with wheels or drums and cables, can provide transitional motion. The electric circuit of the armature and the free body diagram of the rotor are shown in the following figure:





For this example, we will assume the following values for the physical parameters. These values were derived by experiment from an actual motor in Carnegie Mellon's undergraduate controls lab.

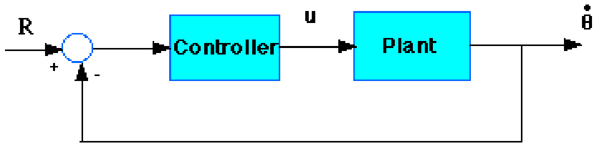
- moment of inertia of the rotor (J) = 3.2284E-6 kg.m^2/s^2
- damping ratio of the mechanical system (b) = 3.5077E-6 Nms
- electromotive force constant (K=Ke=Kt) = 0.0274 Nm/Amp
- electric resistance (R) = 4 ohm
- electric inductance (L) = 2.75E-6 H
- input (V): Source Voltage
- output (theta): position of shaft
- The rotor and shaft are assumed to be rigid.

Model is taken from the Well known Matlab Education site, please visit it at my web-site, please

From the main problem, the dynamic equations in transfer function form are the following:

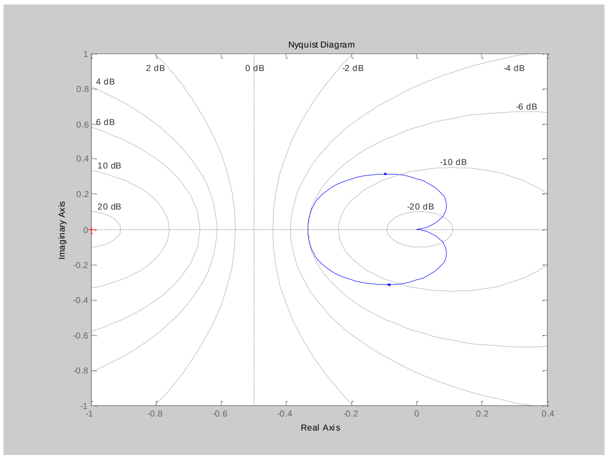
$$\frac{\theta}{V} = \frac{K}{s((Js+b)(Ls+R)+K^2)}$$

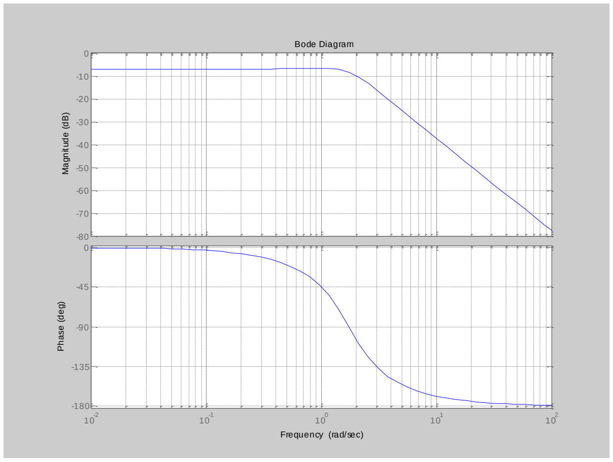
and the system schematic looks like:



For the original problem setup and the derivation of the above equations, please refer to the Modeling a DC Motor page. With a 1 rad/sec step reference, the design criteria are:

- Settling time less than 0.04 seconds
- Overshoot less than 16%
- No steady-state error
- No steady-state error due to a disturbance





Steps:

- With simplification of the DC. Motor, use the following transfer function: n=1; d=[(s+1)(s+2)].
- Compute the required system to gain value for an Mp of 2 and 2.5.
- Construct the Nyquist for this system.
- Verify the stability of the system via simulation and via building up the system.
- Check the Mp value via Bode analysis. Make your comments.
- Make your own conclusions about the experiment and values computed.

UNIVERSITY OF BAHRAIN
COLLEGE OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

NONLINEAR CONTROL SYSTEM
EENG 485

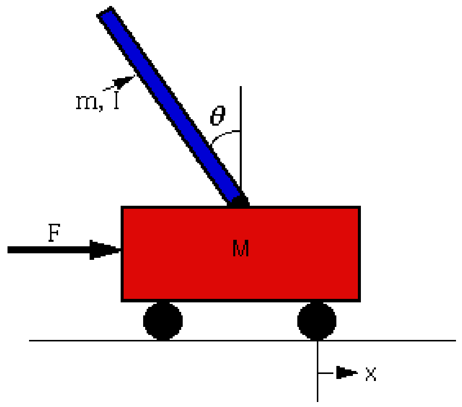
LABORATORY NO.7
PRACTICAL: MAXIMUM OF (10) MARKS

VIA LYAPUNOV-BASED DESIGN AND MODEL REFERENCE ADAPTIVE CONTROLLER

Project Objectives:

To design a nonlinear controller for the inverted pendulum system. The design specifications are as :

Settling time of less than 5 seconds.
Pendulum angle never more than 0.05 radians from the vertical.
Settling time for x and theta of less than 5 seconds.
Rise time for x of less than 0.5 seconds.
Overshoot of theta less than 20 degrees (0.35 radians).



M	mass of the cart	0.5 kg
m	mass of the pendulum	0.5 kg
b	friction of the cart	0.1 N/m/sec
l	length to pendulum centre of mass	0.3 m
I	inertia of the pendulum	0.006 kg*m^2
F	force applied to the cart	
x	cart position coordinate	

theta pendulum angle from vertical

Problem setup and design requirements

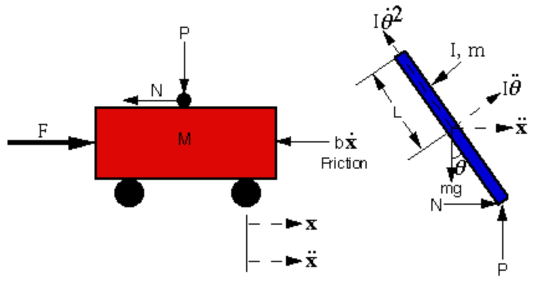
The cart with an inverted pendulum, shown below, is "bumped" with an impulse force, F. Determine the dynamic equations of motion for the system, and linearize about the pendulum's angle, theta = Pi (in other words, assume that pendulum does not move more than a few degrees away from the vertical, chosen to be at an angle of Pi).

A classical PID, root locus, and frequency response sections of this problem, we will be only interested in the control of the pendulums position.   For these sections we will assume that the system starts at equilibrium, and experiences an impulse force of 1N. The pendulum should return to its upright position within 5 seconds, and never move more than 0.05 radians away from the vertical.

However, with the state-space method we are more readily able to deal with a multi-output system. Therefore, for this section of the Inverted Pendulum example we will attempt to control both the pendulum's angle and the cart's position. To make the design more challenging we will be applying a step input to the cart. The cart should achieve it's desired position within 5 seconds and have a rise time under 0.5 seconds. We will also limit the pendulum's overshoot to 20 degrees (0.35 radians), and it should also settle in under 5 seconds.

System Modeling and Force analysis and Nonlinear system equations

Below are the two Free Body Diagrams of the system.



Summing the forces in the Free Body Diagram of the cart in the horizontal direction, you get the following equation of motion:

$$M\ddot{x}+b\dot{x}+N=F$$

Note that you could also sum the forces in the vertical direction, but no useful information would be gained.

Summing the forces in the Free Body Diagram of the pendulum in the horizontal direction, you can get an equation for N:

$$N=m\ddot{x}+ml\ddot{\theta}\cos\theta-ml\dot{\theta}^2\sin\theta$$

If you substitute this equation into the first equation, you get the first equation of motion for this system:

$$(M+m)\ddot{x}+b\dot{x}+ml\ddot{\theta}\cos\theta-ml\dot{\theta}^2\sin\theta=F$$

To get the second equation of motion, sum the forces perpendicular to the pendulum. Solving the system along this axis ends up saving you a lot of algebra. You should get the following equation:

$$P\sin\theta+N\cos\theta-mg\sin\theta=ml\ddot{\theta}+m\ddot{x}\cos\theta$$

To get rid of the P and N terms in the equation above, sum the moments around the centroid of the pendulum to get the following equation:

$$-Pl\sin\theta-Nl\cos\theta=I\ddot{\theta}$$

Combining these last two equations, you get the second dynamic equation:

$$(I+ml^2)\ddot{\theta}+mgl\sin\theta=-ml\ddot{x}\cos\theta$$

Since Matlab can only work with linear functions, this set of equations should be linearized about theta = Pi. Assume that theta = Pi + ø (ø represents a small angle from the vertical upward direction). Therefore, cos(theta) = -1, sin(theta) = -ø, and (d(theta)/dt)^2 = 0. After linearization the two equations of motion become (where u represents the input):

$$\left(\mathbf{I}+\mathbf{ml^2}\right)\ddot{\phi}-\mathbf{mgl}\phi=\mathbf{ml}\ddot{\mathbf{x}}$$
$$(\mathbf{M}+\mathbf{m})\ddot{\mathbf{x}}+\mathbf{b}\dot{\mathbf{x}}-\mathbf{ml}\ddot{\phi}=\mathbf{u}$$

Linear Transfer Function Approach

To obtain the transfer function of the linearized system equations analytically, we must first take the Laplace transform of the system equations. The Laplace transforms are:

$$(\mathbf{I}+\mathbf{ml}^2)\Phi(s)s^2-\mathbf{mgl}\Phi(s)=\mathbf{ml}\mathbf{X}(s)s^2$$
$$(\mathbf{M}+\mathbf{m})\mathbf{X}(s)s^2+\mathbf{b}\mathbf{X}(s)s-\mathbf{ml}\Phi(s)s^2=\mathbf{U}(s)$$

Since we will be looking at the angle Phi as the output of interest, solve the first equation for X(s),

$$\mathbf{X}(s)=\left[\frac{(\mathbf{I}+\mathbf{ml}^2)}{\mathbf{ml}}-\frac{\mathbf{g}}{\mathbf{s}^2}\right]\Phi(s)$$

then substituting into the second equation:

$$(\mathbf{M}+\mathbf{m})\left[\frac{(\mathbf{I}+\mathbf{ml}^2)}{\mathbf{ml}}+\frac{\mathbf{g}}{\mathbf{s}}\right]\Phi(s)s^2+\mathbf{b}\left[\frac{(\mathbf{I}+\mathbf{ml}^2)}{\mathbf{ml}}+\frac{\mathbf{g}}{\mathbf{s}}\right]\Phi(s)s-\mathbf{ml}\Phi(s)s^2=\mathbf{U}(s)$$

Re-arranging, the transfer function is:

$$\frac{\Phi(s)}{\mathbf{U}(s)}=\frac{\frac{\mathbf{ml}}{\mathbf{q}}s^2}{s^4+\frac{\mathbf{b}(\mathbf{I}+\mathbf{ml}^2)}{\mathbf{q}}s^3-\frac{(\mathbf{M}+\mathbf{m})\mathbf{mgl}}{\mathbf{q}}s^2-\frac{\mathbf{bmgl}}{\mathbf{q}}s}$$

where,

$$\mathbf{q}=\left[(\mathbf{M}+\mathbf{m})(\mathbf{I}+\mathbf{ml}^2)-(\mathbf{ml})^2\right]$$

From the transfer function above it can be seen that there is both a pole and a zero at the origin. These can be canceled and the transfer function becomes:

$$\frac{\Phi(s)}{\mathbf{U}(s)}=\frac{\frac{\mathbf{ml}}{\mathbf{q}}s}{s^3+\frac{\mathbf{b}(\mathbf{I}+\mathbf{ml}^2)}{\mathbf{q}}s^2-\frac{(\mathbf{M}+\mathbf{m})\mathbf{mgl}}{\mathbf{q}}s-\frac{\mathbf{bmgl}}{\mathbf{q}}}$$

Linear State-Space

After a little algebra, the linearized system equations can also be represented in state-space form (Prove that !!):

$$\begin{bmatrix}\dot{x}\\ \ddot{x}\\ \dot{\Phi}\\ \ddot{\Phi}\end{bmatrix}=\begin{bmatrix}0 & 1 & 0 & 0\\ 0 & \frac{-(I+ml^2)b}{I(M+m)+Mml^2} & \frac{m^2gl^2}{I(M+m)+Mml^2} & 0\\ 0 & 0 & 0 & 1\\ 0 & \frac{-mlb}{I(M+m)+Mml^2} & \frac{mgl(M+m)}{I(M+m)+Mml^2} & 0\end{bmatrix}\begin{bmatrix}x\\ \dot{x}\\ \Phi\\ \dot{\Phi}\end{bmatrix}+\begin{bmatrix}0\\ \frac{I+ml^2}{I(M+m)+Mml^2}\\ 0\\ \frac{ml}{I(M+m)+Mml^2}\end{bmatrix}u$$

$$y=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}x\\ \dot{x}\\ \Phi\\ \dot{\Phi}\end{bmatrix}+\begin{bmatrix}0\\ 0\end{bmatrix}u$$

The **C** matrix is 2 by 4, because both the cart's position and the pendulum's position are part of the output. For the state-space design problem we will be controlling a multi-output system so we will be observing the cart's position from the first row of output and the pendulum's with the second row.

Matlab representation and the open-loop response
Transfer Function Approach

The transfer function found from the Laplace transforms can be set up using Matlab by inputting the numerator and denominator as vectors. Create an m-file and copy the following text to model the transfer function:

```
M = .5;
m = 0.2;
b = 0.1;
i = 0.006;
g = 9.8;
l = 0.3;

q = (M+m)*(i+m*l^2)-(m*l)^2;   %simplifies input

num = [m*l/q  0]
den = [1  b*(i+m*l^2)/q  -(M+m)*m*g*l/q  -b*m*g*l/q]
```

To observe the system's velocity response to an impulse force applied to the cart add the following lines at the end of your m-file:

```
t=0:0.01:5;
impulse(num,den,t)
axis([0 1 0 60])
```

Verify the open loop response.   Hence, discuss it.

Matlab representation and the open-loop response
State-Space Approach

```
M =  0.5;
m =  0.2;
```

```
b  =  0.1;
i  =   0.006;
g  =  9.8;
l  =  0.3;

p = i*(M+m)+M*m*l^2;     denominator for the A and B matrices :

A = [0     1                       0                        0
      0    -(i+m*l^2)*b/p     (m^2*g*l^2)/p      0
      0     0                     0                        1
      0    -(m*l*b)/p            m*g*l*(M+m)/p   0]

B = [     0
           (i+m*l^2)/p
           0
            m*l/p]

C = [ 1   0   0   0
        0   0   1   0]

D = [ 0
       0]

T=0:0.05:10;
U=0.2*ones(size(T));
[Y,X]=lsim(A,B,C,D,U,T);
plot(T,Y)
axis([0 2 0 100])
```

Observe the system outputs after running the m-file.   If you turn to linear state space, the following matrices of state space are defined.   Show how you have get them via linearization.

```
A =
      0    1.0000        0               0
      0   -0.1818       2.6727         0
      0        0           0               1.0000
      0   -0.4545     31.1818         0

B =
      0
      1.8182
      0
      4.5455

C =
    1    0    0    0
    0    0    1    0

D =
    0
    0
```

Discuss the open loop response.   Show the cart's position and the pendulum's angle .

Laypunov Design Approach :

It is desired to stabilize the inverted pendulum at the position $\theta = \delta_1$.

Design a state feedback controller using Laypunov redesign.   Verify that the tracking error converges to zero.
Design a state feedback controller using (continuous) sliding mode control.   Verify that the tracking error converges to zero.
Using simulation, compare the performance of the two controllers.   In the simulation, you may use different parameters constants. Like (K=10, q = 0.1 and a =10).

Model reference adaptive controller Design :

Hence, design a model reference adaptive controller for the inverted pendulum system. Propose a transfer function for P(s) and M(s).
Simulate the closed-loop system. In your simulation, use r= 1,   r= 1 + sin(t)   and   r = sin(t) + sin(2t).
Show in a clear way your design steps.
There will be a presentation day, through which you (as a group of two) are requested to present the design and results.

Note :   -   This assignment carries a weight, please achieve it fully.
A group could divide the task among themselves.
Just in case of any un understandable items, do not hesitate to ask me.
Date of Assignment submission to students is : 11/04/2006.
To be submitted back by 4/6/2006 (after the class presentation on same day)   and no longer.

NOTE :   Refer to my Web-site for more information about MATLAB Carnegie Mellon Modeling Part.