

Modelling and Control of Ball-Plate System

Final Project Report

Mohammad Nokhbeh and Daniel Khashabi

Under the supervision of Dr.H.A.Talebi

Amirkabir University of Technology, 2011

Abstract

Abstract—In this report we will have a detailed look into controlling a Ball-on-Plate system apparatus. We will first derive the exact non-linear differential equations of Ball-on-Plate apparatus by the use of Lagrange-Euler equations. At the rest we derive several useful linearised model for our system. We use the linearised model to design compensator for the system. We simulate the results using non-linear model of system and evaluate the results by step responses and trajectory tracking.

Contents

1	Introduction	2
2	Review on Previous Works	2
3	Mathematical Modelling	3
3.1	Plant Values:	6
3.2	Motor Model	6
3.3	Model Linearisation	7
3.3.1	Linearised Model of a Ball-on-Plate	7
3.3.2	State-space Model of System	8
3.3.3	Complete Linearisation of Ball-on-Plate and DC Motor	9
4	Controller Design	10
4.1	Single Compensator in Open-Loop	10
4.2	Compensator With Prefilter	11
4.2.1	Multi-loop controller 1	12
4.2.2	Multi-loop controller 2	12
5	Result Evaluation	13
5.1	Non-linear Simulation	15
5.1.1	Trajectory Tracking Using Non-linear Simulator	16

5.1.2	Coulomb Friction	16
5.2	Two-Dimensional Overshoot	17
6	Conclusions	18
7	Acknowledgement	18

1 Introduction

Balancing systems are one of most challenging problems in *Control* field. There are lots of platforms for this sake like cart-pole system(inverted pendulum), ball-beam system, double and multiple inverted pendulums[1].

The ball-on-plate system is a promoted version of the traditional ball-on-beam control problem. The problem consists of a plate which its deviation can be manipulated in two perpendicular directions. The goal is to carry the ball moving on the plate to a desired position, that is to control a freely rolling ball on a specific position or moving on a trajectory on the plate. In more precise way, one can do *stabilization* control in which the goal is to carry the ball from one specific position and hold it in the desired position or do *trajectory tracking* control in which the goal is to make the ball follow a predefined trajectory with the least error and in a the minimum time.

2 Review on Previous Works

We can classify the previous works on ball-on-plate system into to general categories. Those based in mathematical modelling and those proposing model-free controllers.

The [2] is devoted to a mechatronic design of the system which benefits a touch sensor feedback. In [1] a Fuzzy Reinforcement Learning method is proposed. The previous paper benefits a high accuracy stepping motor with 0.9 degree size of steps which is reducible by the drivers down to 1/200 of step size, i.e. $4.5 * 10^{-3}$. It uses precise incremental encoder (3600 ppr) which is coupled to motor's shaft. Altogether by taking the mechanical limitations, it benefits the smallest measurable and applicable amount of rotation 0.1 degrees. It uses a 15 inches touch sensor that its output is a message packet being sent through RS-232 serial communication with 19200 bps. Thus the fastest sampling rate of the whole sensor block is 384 samples per second. This implies that the maximum available time for decision making is $1/384 \cong 2.6 * 10^{-3}$ seconds. The area of the surface of the touch sensor on which pressure can be sensed $30.41 \times 22.81cm$. Sensor resolution is 1900*1900 pixels so if the sensor sensitivity is uniformly distributed on its sensitive area, then each pixel is assigned to an area approximately equals to $1.8*1.2 mm^2$ of the surface of the sensor.

In [3] a PID neural network controller(PIDNN) based on genetic algorithm(GA) is used for control. In this approach GA is applied in training weighting factor of multilayer feed-forward neural network in order to overcome the disadvantage

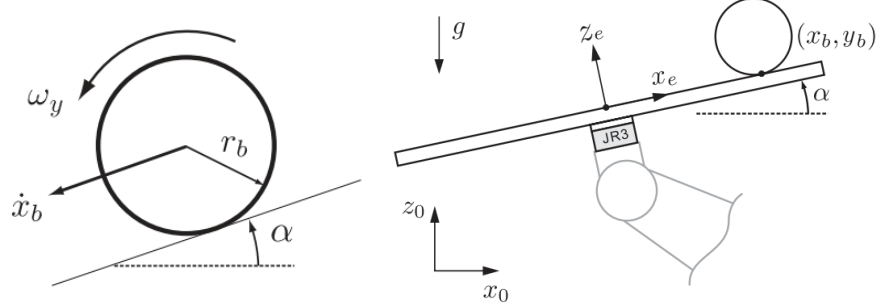


Figure 1: Side view of ball on plate system.

of back-propagation algorithm which falls into local minima. It is claimed that the proposed method is robust, adaptive and strong control performance.

In [2] ball position measured by a touch pad rather than a camera. In the previous paper a linear quadratic state feedback regulator was designed after the system was linearised around a few operating points. Accuracy of stabilization 5 mm. Steady state error in average was 18 mm in circle following while trajectory velocity was less than 4.2 mm/s

In [4] a hierarchical fuzzy control scheme with a TS type fuzzy tracking controller was proposed. In [5] sliding mode control with uncertain items observe compensation, is used to control a ball-on-plate system.

3 Mathematical Modelling

In this section we are going to derive the motion equations of system. In these part we assume following simplifications:

- There is no slipping for ball.
- The ball is completely symmetric and homogeneous.
- All frictions are neglected.
- The ball and plate are in contact all the time.

Here we derive dynamical equations of ball-on-plate system by the help of Lagrangian. The following precise mathematical equations are based on [6][4][7] ([4, 6, 7]). The Euler-Lagrange equation of ball-plate system is as followings:

$$\frac{d}{dt} \frac{\partial T}{\partial \dot{q}_i} - \frac{\partial T}{\partial q_i} + \frac{\partial V}{\partial q_i} = Q_i \quad (1)$$

Where q_i stands for i -direction coordinate, T is kinetic energy of the system, V is potential energy of system and Q is composite force.

The system has 4 degree of freedom; two in ball motion and two in inclination of plate. Here we assume the generalized coordinates of system to be (is) x_b and y_b ball's position on plate and α and β the inclination of the plate. It is important to note that we assume the center of $x - y$ coordinates be at center of plate. The kinetic energy of ball consists of its both rotational with respect to its center of mass and translational energy:

$$T_b = \frac{1}{2}m_b (\dot{x}_b^2 + \dot{y}_b^2) + \frac{1}{2}I_b (\omega_x^2 + \omega_y^2) \quad (2)$$

Where m_b is mass of the ball and I_b is moment of inertia of the ball; \dot{x}_b and \dot{y}_b are ball's translational velocities along x -axis and y -axis; ω_x and ω_y are ball's rotational velocities along x -axis and y -axis. The following relations between translational velocities and rotational velocities:

$$\dot{x}_b = r_b \omega_y \quad , \quad \dot{y}_b = r_b \omega_x \quad (3)$$

In which r_b denotes ball's radius. By substituting equations 3 into equations 4 we will have:

$$T_b = \frac{1}{2} \left[m_b (\dot{x}_b^2 + \dot{y}_b^2) + \frac{I_b}{r_b^2} (\dot{x}_b^2 + \dot{y}_b^2) \right] = \frac{1}{2} \left(m_b + \frac{I_b}{r_b^2} \right) (\dot{x}_b^2 + \dot{y}_b^2) \quad (4)$$

The kinetic energy of the plate, by considering ball as a point mass which is placed in (x_b, y_b) , consists of the (ezafi) its rotational energy with respect to its center of mass:

$$\begin{aligned} T_p &= \frac{1}{2} (I_p + I_b) (\dot{\alpha}^2 + \dot{\beta}^2) + \frac{1}{2} m_b (x_b \dot{\alpha} + y_b \dot{\beta})^2 \\ &= \frac{1}{2} (I_p + I_b) (\dot{\alpha}^2 + \dot{\beta}^2) + \frac{1}{2} m_b (x_b^2 \dot{\alpha}^2 + 2x_b \dot{\alpha} y_b \dot{\beta} + y_b^2 \dot{\beta}^2) \end{aligned} \quad (5)$$

Where α and β are plate's angle of inclination along x -axis and y -axis, respectively. Therefore $\dot{\alpha}$ and $\dot{\beta}$ are plate's rotational velocity. Here we can calculate the kinetic energy of system as followings:

Where α and β are plate's angle of inclination along x -axis and y -axis, respectively. Therefore $\dot{\alpha}$ and $\dot{\beta}$ are plate's rotational velocity. Here we can calculate the kinetic energy of system as followings: (paragraf tekrari)

$$\begin{aligned} T &= T_b + T_p \\ &= \frac{1}{2} \left(m_b + \frac{I_b}{r_b^2} \right) (\dot{x}_b^2 + \dot{y}_b^2) + \frac{1}{2} (I_p + I_b) (\dot{\alpha}^2 + \dot{\beta}^2) \\ &\quad + \frac{1}{2} m_b (x_b^2 \dot{\alpha}^2 + 2x_b \dot{\alpha} y_b \dot{\beta} + y_b^2 \dot{\beta}^2) \end{aligned} \quad (6)$$

The potential energy of the ball relative (the relative potential) to horizontal plane in the center of the inclined plate can be calculated as:

$$V_b = m_b g h = m_b g (x_b \sin \alpha + y_b \sin \beta) \quad (7)$$

Here we can derive the system's equation by Lagrangian and equations 4-7:

$$L = T_b + T_p - V_b \quad (8)$$

We use L to derive system's equations:

$$\frac{\partial T}{\partial \dot{\alpha}} = (I_p + I_a) \dot{\alpha}_x + m_b x_b (x_b \dot{\alpha} + y_b \dot{\beta}) \quad , \quad \frac{\partial L}{\partial \alpha} = m g \cos \alpha \quad (9)$$

$$\frac{\partial T}{\partial \dot{\beta}} = (I_p + I_a) \dot{\beta}_x + m_b y_b (y_b \dot{\beta} + x_b \dot{\alpha}) \quad , \quad \frac{\partial L}{\partial \beta} = m g \cos \beta \quad (10)$$

$$\frac{\partial T}{\partial \dot{x}_b} = \left(m_b + \frac{I_b}{r_b^2} \right) \dot{x}_b \quad , \quad \frac{\partial L}{\partial x_b} = m_b (x_b \dot{\alpha} + y_b \dot{\beta}) \dot{\alpha} \quad (11)$$

$$\frac{\partial T}{\partial \dot{y}_b} = \left(m_b + \frac{I_b}{r_b^2} \right) \dot{y}_b \quad , \quad \frac{\partial L}{\partial y_b} = m_b (x_b \dot{\alpha} + y_b \dot{\beta}) \dot{\beta} \quad (12)$$

We assume generalized toques as τ_x and τ_y which are exerted torques on the plate. From Lagrange-Euler equation we can write:

$$\begin{aligned} \frac{d}{dt} \frac{\partial T}{\partial \dot{\alpha}} - \frac{\partial L}{\partial \alpha} &= (I_p + I_b) \ddot{\alpha} + m_b x_b^2 \ddot{\alpha} + 2m_b x_b \dot{x}_b \dot{\alpha} + m_b x_b y_b \ddot{\alpha} \\ &+ m_b \dot{x}_b y_b \dot{\beta} + m_b x_b \dot{y}_b \dot{\beta} - m_b g \cos \alpha = \tau_x \end{aligned} \quad (13)$$

$$\begin{aligned} \frac{d}{dt} \frac{\partial T}{\partial \dot{\beta}} - \frac{\partial L}{\partial \beta} &= (I_p + I_b) \ddot{\beta} + m_b y_b^2 \ddot{\beta} + 2m_b y_b \dot{y}_b \dot{\beta} + m_b x_b y_b \ddot{\beta} \\ &+ m_b \dot{y}_b x_b \dot{\alpha} + m_b y_b \dot{x}_b \dot{\alpha} - m_b g \cos \beta = \tau_y \end{aligned} \quad (14)$$

$$\frac{d}{dt} \frac{\partial T}{\partial \dot{x}_b} - \frac{\partial L}{\partial x_b} = \left(m_b + \frac{I_b}{r_b^2} \right) \ddot{x}_b - m_b (x_b \dot{\alpha} + y_b \dot{\beta}) \dot{\alpha} + m_b g \sin \alpha = 0 \quad (15)$$

$$\frac{d}{dt} \frac{\partial T}{\partial \dot{y}_b} - \frac{\partial L}{\partial y_b} = \left(m_b + \frac{I_b}{r_b^2} \right) \ddot{y}_b - m_b (y_b \dot{\beta} + x_b \dot{\alpha}) \dot{\beta} + m_b g \sin \beta = 0 \quad (16)$$

So the non-linear differential equations for the ball-plate-system as follows:

$$\left(m_b + \frac{I_b}{r_b^2} \right) \ddot{x}_b - m_b (x_b \dot{\alpha}^2 + y_b \dot{\alpha} \dot{\beta}) + m_b g \sin \alpha = 0 \quad (17)$$

$$\left(m_b + \frac{I_b}{r_b^2} \right) \ddot{y}_b - m_b (y_b \dot{\beta}^2 + x_b \dot{\alpha} \dot{\beta}) + m_b g \sin \beta = 0 \quad (18)$$

$$\begin{aligned} \tau_x &= (I_p + I_b + m_b x_b^2) \ddot{\alpha} + 2m_b x_b \dot{x}_b \dot{\alpha} + m_b x_b y_b \ddot{\alpha} \\ &+ m_b \dot{x}_b y_b \dot{\beta} + m_b x_b \dot{y}_b \dot{\beta} + m_b g x_b \cos \alpha \end{aligned} \quad (19)$$

$$\begin{aligned} \tau_y &= (I_p + I_b + m_b y_b^2) \ddot{\beta} + 2m_b y_b \dot{y}_b \dot{\beta} + m_b y_b x_b \ddot{\beta} \\ &+ m_b \dot{x}_b y_b \dot{\alpha} + m_b x_b \dot{y}_b \dot{\alpha} + m_b g y_b \cos \beta \end{aligned} \quad (20)$$

The equations (18)-(17) shows the relation between ball's state and plate's state that is plate's inclination. The equations (19)-(20) shows the effect of external torque on the ball-on-plate system. It is obvious that working with such complex equations is too hard. So in order to do Laplace analysis on system's transfer function, we derive the linearised model of system in neighbourhood of working state.

3.1 Plant Values:

In this report we use the values that are shown in following table that are based on [4]:

Parameter	Description	Value
m_b	Mass of the ball	$0.11kg$
r_b	Radius of the ball	$0.02m$
S_p	Dimension of the plate	$1.0 \times 1.0m^2$
v_{max}	Maximum velocity of the ball	$4mm/s$
m_p	Mass of the plate	$0.1kg$
I_p	Mass moment of inertia of the plate	$0.5kg.m^2$
I_b	Mass moment of inertia of the ball	$1.76 \times 10^{-5}kg.m^2$
g	Gravitational acceleration	9.81

3.2 Motor Model

Here we assume (use or consider) an *Armature-Controlled DC motor* which it's block diagram is shown in figure 2. The motor uses armature current i_a as the control variable. The stator field can be established by a field coil current. The detailed description of equations can be found in [8] and [9]. When we have a constant current in the field coil, we have the following postulates:

$$\begin{aligned}
T_m(s) &= K_m I_a(s) \\
V(s) &= (R_a + L_a s) I_a(s) + E_a(s) \\
V(s) &= K_b \omega(s) \\
\Rightarrow I_a(s) &= [V(s) - K_b \omega(s)] / [R_a + L_a s]
\end{aligned}$$

The absolute torque exerted on the load can be found by:

$$T_L(s) = T_m(s) - T_d(s)$$

By letting $T_d(s) = 0$ and considering $G_L(s) = \frac{\omega_L(s)}{T_L(s)}$ to the load torque-velocity characteristic, one can find the input-output transfer function:

$$G(s) = \frac{\theta_L(s)}{V(s)} = \frac{K_m G_L(s)}{s[(R_a + sL_a) + K_m K_b G_L(s)]}$$

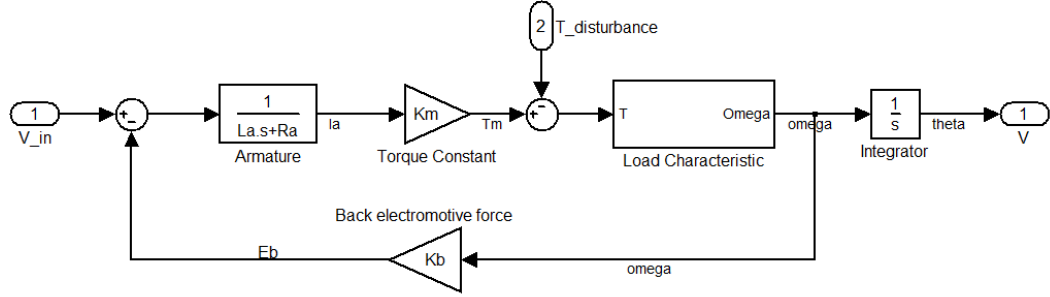


Figure 2: Block diagram for an armature-controlled DC motor.

We can assume an approximate function for load characteristic as following :
 $T_L(s) = \frac{1}{Js+B}$ Where J stands for load inertia and B for load damping. We rewrite the motors transfer function in this way:

$$G(s) = \frac{\theta_L(s)}{V(s)} = \frac{K_m}{s[(R_a + sL_a)(Js+B) + K_m K_b]}$$

We assume following typical values for DC motor[10]:

Parameter	Description	Value
K_m	Toque constant	$6.876 \times 10^{-4} N.m/A$
K_b	Back-emf constant	$7.2 \times 10^{-5} V/rpm$ or $1.9099 \times 10^{-7} V.s/rad$
J_m	Rotor inertia	$1 \times 10^{-3} N.m.s^2/rad$
B	Rotor damping	$10^{-8} N.m/(rad/s)$
τ_f	Field time constant	$1ms$
τ	Rotor time constant	$100ms$
R	Armature resistance	3.9Ω
L	Armature inductance	$12 \times 10^{-6} H$
P_{max}	Maximum output power	$0.25hp$
T_s	Stall torque	$2.4 \times 10^{-4} N \times m$
V_{fl}	Rated DC supply voltage	$1.5V$

3.3 Model Linearisation

In this part we will derive several linearised model of system that we will use them in next sections.

3.3.1 Linearised Model of a Ball-on-Plate

The approximate value for a solid ball's moment of inertia is $I_{ball} = \frac{2}{5}m_b r_b^2$. Therefore equations (17) and (18) can be written as:

$$m_b \left[\frac{5}{7} \ddot{x}_b - (x_b \dot{\alpha}^2 + y_b \dot{\alpha} \dot{\beta}) + g \sin \alpha \right] = 0 \quad (21)$$

$$m_b \left[\frac{5}{7} \ddot{y}_b - (y_b \dot{\beta}^2 + x_b \dot{\alpha} \dot{\beta}) + g \sin \beta \right] = 0 \quad (22)$$

We can linearise above equation by assuming:

- Small angle of inclination for the plate (up to $\pm 5^\circ$): $\alpha \ll 1$ and $\beta \ll 1$
 $\implies \sin \alpha \simeq \alpha$, $\sin \beta \simeq \beta$
- Slow rate of change for the plate: $\dot{\alpha} \ll 0$ and $\dot{\beta} \ll 0 \implies \dot{\alpha} \dot{\beta} \simeq 0$, $\dot{\alpha}^2 \simeq 0$, $\dot{\beta}^2 \simeq 0$.

$$\frac{5}{7} \ddot{x}_b + g \alpha = 0 \quad (23)$$

$$\frac{5}{7} \ddot{y}_b + g \beta = 0 \quad (24)$$

By linearising the above equations, we find two separate differential equations for each of x and y axis. Note that we can use above linear differential equations to estimate the ball-on-plate system's states x_b , y_b , \dot{x}_b , \dot{y}_b . By assuming $\alpha(s)$ and $\beta(s)$ as inputs to ball-on-plate system, we find the transfer functions:

$$P_x(s) = \frac{X_b(s)}{\alpha(s)} = \frac{g}{\frac{5}{7}s^2} \quad (25)$$

$$P_y(s) = \frac{Y_b(s)}{\beta(s)} = \frac{g}{\frac{5}{7}s^2} \quad (26)$$

3.3.2 State-space Model of System

One can write state equations of ball-on-plate system by considering Equations (17)-(20)[3]. As mentioned τ_x and τ_y are torques exerted on the plane (or on the ball from the earth) in y-axis and x-axis direction respectively. By defining state variables $X = [x_1, x_2, x_3, x_4, x_5, x_6]^T = [x_b, \dot{x}_b, \alpha, \dot{\alpha}, y_b, \dot{y}_b]^T$, $U = [u_x, u_y]^T = [\alpha, \beta]^T$ and constant value $B = m / (m + I_b / r_b^2)$ we write the non-linear state equation $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} = \begin{bmatrix} x_2 \\ B(x_1 x_4^2 + x_4 x_5 x_8 - g \sin x_3) \\ x_4 \\ 0 \\ x_6 \\ B(x_5 x_8^2 + x_1 x_4 x_8 - g \sin x_7) \\ x_8 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix}$$

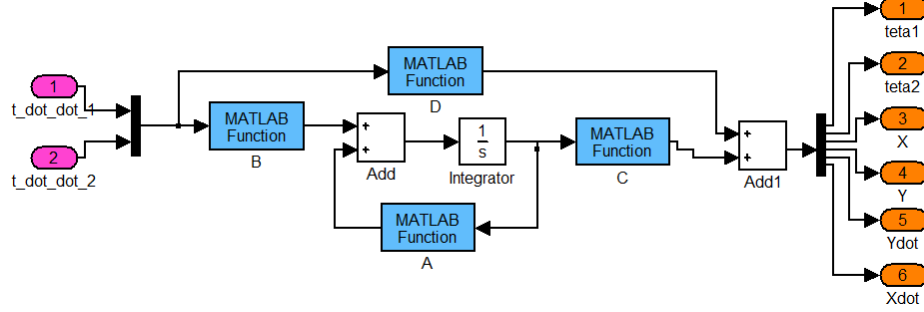


Figure 3: Block diagram of the Plant.

Because of the low velocity and acceleration of the plate rotation ($|\dot{\alpha}| \ll 1$ and $|\dot{\beta}| \ll 1$) we can omit the coupling term in $f(\mathbf{x}, \mathbf{u})$ and divide the system into to sub-systems and control them independently.

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ B(x_1 x_4^2 - g \sin x_3) \\ x_4 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} [u_x]$$

$$\begin{bmatrix} x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} = \begin{bmatrix} x_6 \\ B(x_5 x_8^2 - g \sin x_7) \\ x_8 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} [u_y]$$

The implementation of the plant is shown in figure 3.

3.3.3 Complete Linearisation of Ball-on-Plate and DC Motor

We use MATLAB's internal function to generate a linear model of our non-linear model. First we consider (ezafi) are going to make a linear of model which consists of Plant which is *ball-on-plate* and a DC motor. The goal of this model is to consider the loading effect of ball and plate on the motor. Because if have ball and plates that have considerable weight, they affect the position/velocity of motor. The complete Simulink model of system is show in figure (5).

The linearisation must be done in infinitesimal neighbourhood of *equilibrium point*. The equilibrium point is where speed and its derivatives are zero. By considering zero value for velocity and acceleration terms, we find the following point as equilibrium:

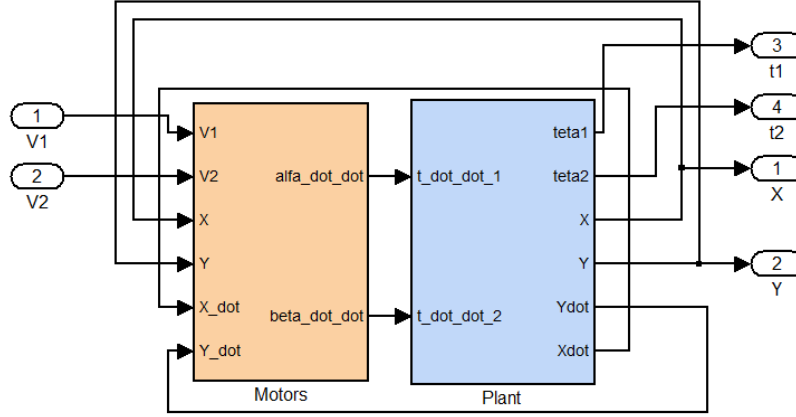


Figure 4: Block diagram of the whole system.

$$X = [x_b, \dot{x}_b, \alpha, \dot{\alpha}, y_b, \dot{y}_b, \beta, \dot{\beta}]^T = [0, 0, 0, 0, 0, 0, 0, 0]^T$$

MATLAB CODE:

```
DefineConstants();
[A,B,C,D]=linmod('PM',[0,0,0,0,0,0,0,0,0,0,0,0,0,0],[0,0]);
sys = ss(A,B,C,D)
sys2=tf(sys)
```

The resulting output transfer functions are as followings:

$$\frac{X(s)}{V_1(s)} = \frac{-802.2}{s^5 + 3.25 \times 10^5 s^4 - 15.09 s - 4.905 \times 10^6}$$

$$\frac{Y(s)}{V_2(s)} = \frac{-802.2}{s^5 + 3.25 \times 10^5 s^4 - 15.09 s - 4.905 \times 10^6}$$

If we plot the *root-locus* plot of the above transfer function, we will get the figure (6).

4 Controller Design

In this section we are going to design compensator for the transfer functions we derived in previous parts. Here we propose several methods of compensation and in

4.1 Single Compensator in Open-Loop

The first compensator consists of only a compensator in *open-loop*. Unfortunately this system is unproper and *DC gain* of this controller is 100 (for *stabilizing*), it's *settling time* (T_s) and *overshoot* is also unacceptable.

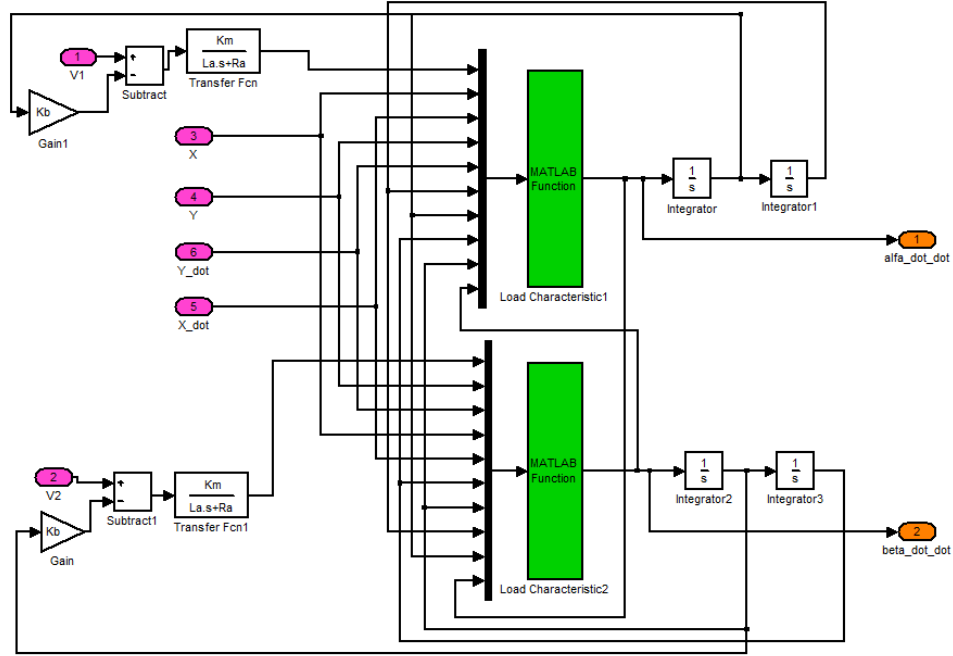


Figure 5: Block diagram of the motors with non-linear load characteristics.

$$C(s) = \frac{-3461538.4615(s+0.01)^2(s+2)(s^2+10s+26)}{(s-1.8)(s+100)}$$

Gain	Poles	Zeros
100	2	5

Root-locus and step response of the first compensator are shown in figure (7).

4.2 Compensator With Prefilter

The second compensator consists of a compensator and a prefilter. Unlike first compensator this system is proper and *DC gain* and *percent overshoot* of this controller is acceptable but this compensator have a $T_s = 15s$ settling time.

$$C(s) = \frac{-0.4294s^5 - 139600s^4 - 323700s^3 - 705500s^2 - 1.202 \times 10^6 s - 798.5}{s^5 + 9.971s^4 + 43.77s^3 + 111.2s^2 + 195.5s}$$

$$F(s) = \frac{798.5}{0.4294s^5 + 139600s^4 + 323700s^3 + 705500s^2 + 1.202 \times 10^6 s + 798.5}$$

Gain	Poles	Zeros	Prefilter poles
4.08	5	5	5

In order to find better compensator, we change the architecture add following add a new controlling loop. The compensator at the rest of report are designed to control plant and previous controller.

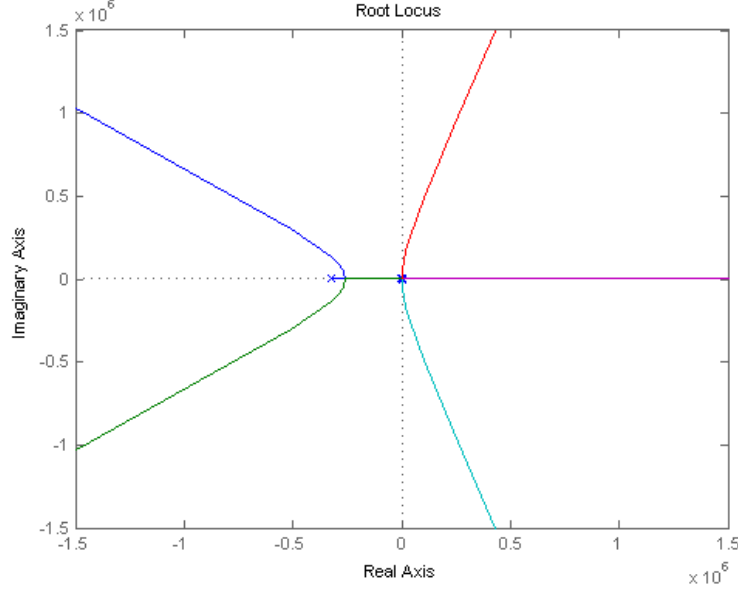


Figure 6: Pole-zero map of system's transfer function.

4.2.1 Multi-loop controller 1

the following system is designed to make the *settling time* of second compensator appropriate. this compensator's aim is to get have *overshoot* but rapid *timerise*.

Gain	Poles	Zeros	Prefilter poles
1.6	8	8	1

$$F(s) = \frac{16.12}{s+10}$$

$$C(s) =$$

$$\frac{7.296 \times 10^{13} s^8 + 5.597 \times 10^{14} s^7 + 1.906 \times 10^{15} s^6 + 3.729 \times 10^{15} s^5 + 4.586 \times 10^{15} s^4 + 3.613 \times 10^{15} s^3 + 1.788 \times 10^{15} s^2 + 4.917 \times 10^{14} s + 6.251 \times 10^{13}}{s^8 + 400 s^7 + 7 \times 10^4 s^6 + 7 \times 10^6 s^5 + 4.375 \times 10^8 s^4 + 1.75 \times 10^{10} s^3 + 4.375 \times 10^{11} s^2 + 6.251 \times 10^{12} s + 3.907 \times 10^{13}}$$

4.2.2 Multi-loop controller 2

the following compensator is desinged to make the *settling time* of second compensator appropriate. this compensator's aim is to *critical damp* mode.

Gain	Poles	Zeros	Prefilter poles
.6	8	8	1

$$F(s) = \frac{26.66}{s+10}$$

$$C(s) =$$

$$\frac{2.736 \times 10^{13} s^8 + 2.099 \times 10^{14} s^7 + 7.148 \times 10^{14} s^6 + 1.398 \times 10^{15} s^5 + 1.72 \times 10^{15} s^4 + 1.355 \times 10^{15} s^3 + 6.706 \times 10^{14} s^2 + 1.844 \times 10^{14} s + 2.344 \times 10^{13}}{s^8 + 400 s^7 + 7 \times 10^4 s^6 + 7 \times 10^6 s^5 + 4.375 \times 10^8 s^4 + 1.75 \times 10^{10} s^3 + 4.375 \times 10^{11} s^2 + 6.251 \times 10^{12} s + 3.907 \times 10^{13}}$$

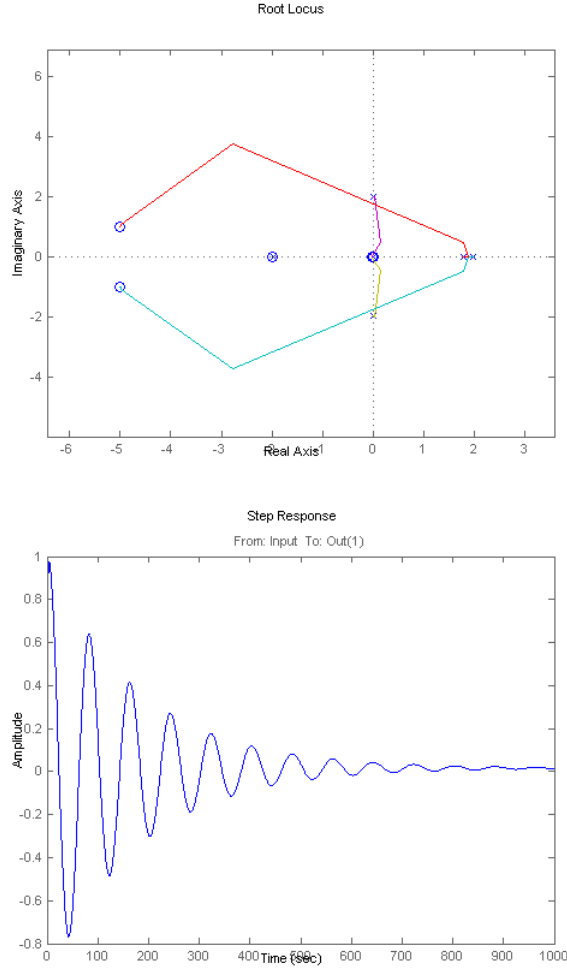


Figure 7: Root-locus and step response of the first compensator.

5 Result Evaluation

In this part we are going to evaluate the results we got from previous parts. The main goal of this part is to check the validity of the previous results and whether they are valid for the case of main non-linear equations. For this goal, we designed the comprehensive non-linear simulator to check the validity of results.

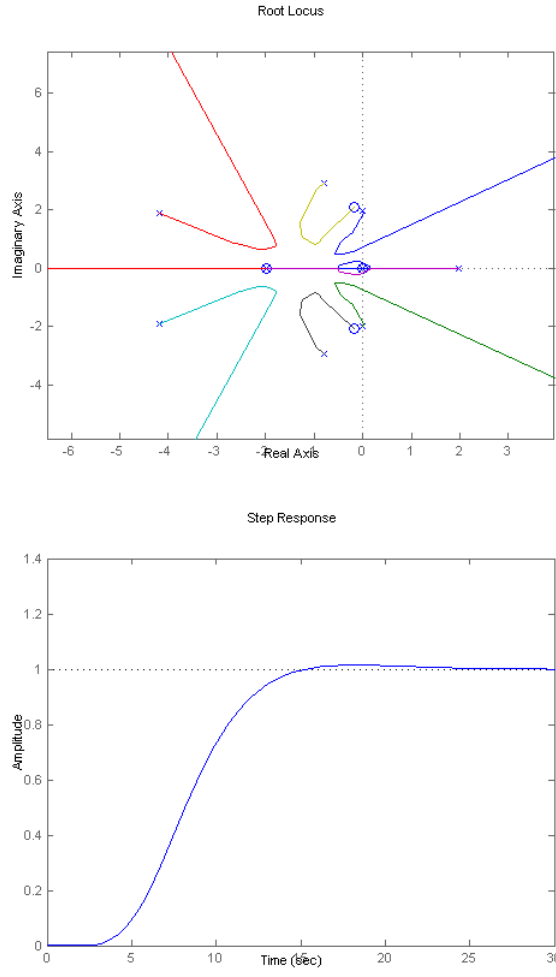


Figure 8: Root-locus and step response of the second compensator.

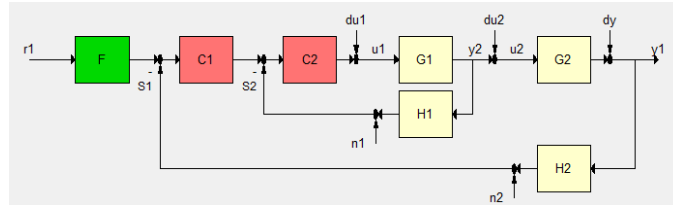


Figure 9: Architecture of added controlling loop.

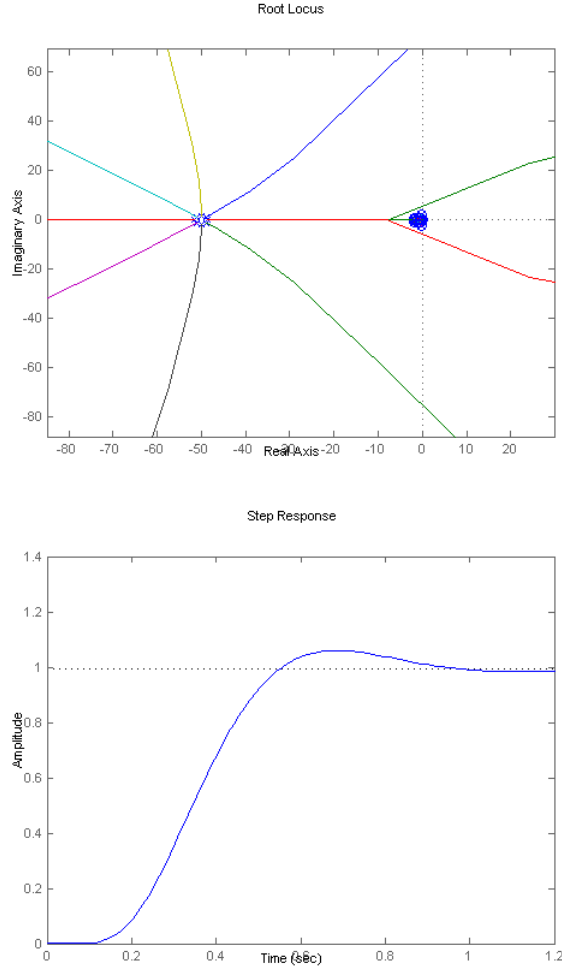


Figure 10: Root-locus and step response of the third compensator.

5.1 Non-linear Simulation

In order to have a realistic evaluation of the designed compensator and its operation we need to design a non-linear simulator by the use of primary non-linear equations (17)-(20).

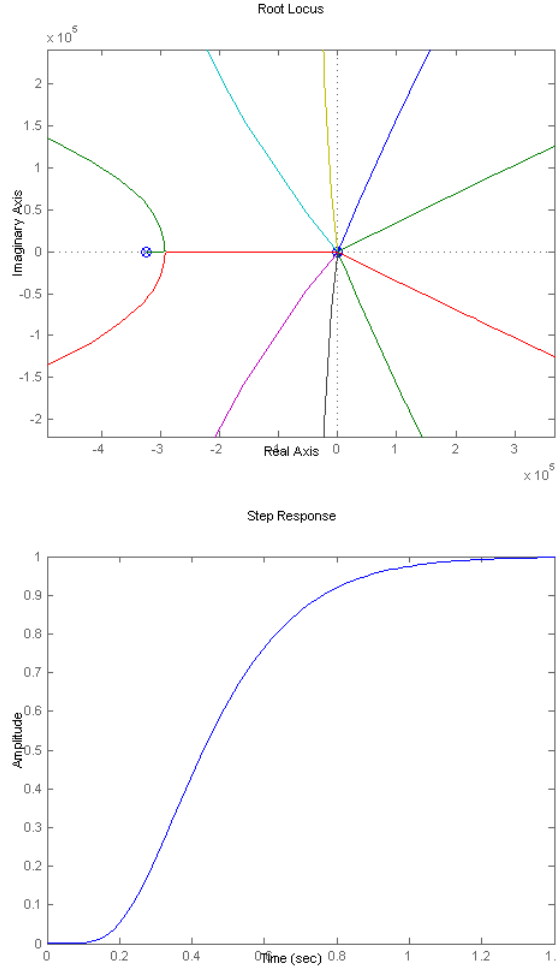


Figure 11: Root-locus and step response of the forth compensator.

5.1.1 Trajectory Tracking Using Non-linear Simulator

The task of tracking a trajectory is one of important tasks that one controller must be able to accomplish. In figure (14) the results of several tracking experiments are shown.

5.1.2 Coulomb Friction

Despite the complex microscopic origins of *Coulomb friction*, we can model it by simple models. One popular model for formulating the Coulomb friction is as following[11] [12] :

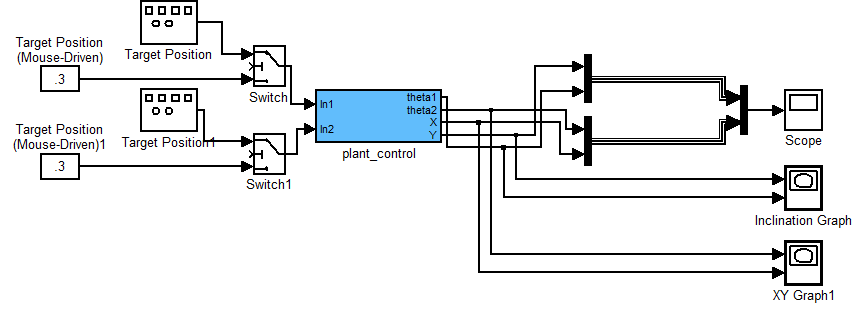


Figure 12: Main block diagram for simulation in presence of designed controller.

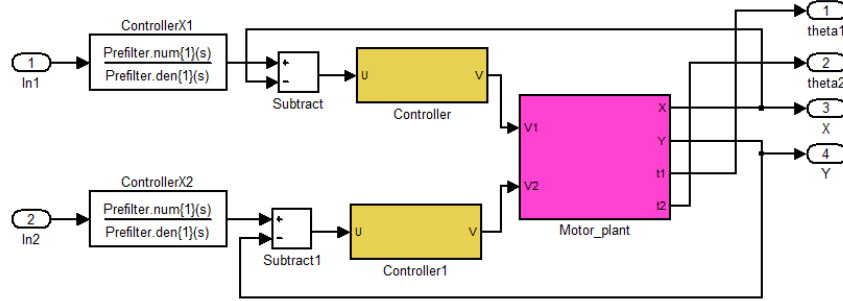


Figure 13: The designed controllers and non-linear implementation of the system.

$$T_f = \rho \tanh(\eta \dot{x})$$

Where ρ and η are constant values. For large values of η the Coulomb friction is similar to a *Sign Function* and for lower values of η we get softer Coulomb friction, as shown in figure (15).

5.2 Two-Dimensional Overshoot

As it is described above system's model is discretely described in each dimension and there is no relation between two subsystem. Consider measuring of overshoot of system, overshoot is the excessive value which systems takes for speed, but what is the relationship between the overshoot measured in each subsystem and overshoot of system. The output signal will be the joint of both systems and because both system's overshoot times are not equal, system will track an unknown trajectory which is for sure in the rectangle of $O_1 \times O_2$. Overshoot is usually considered in building of the physical system, the system's bounds are based on overshoots measured. Considering all above a new measurement of

overshoot can be considered which is maximum distance from the zero point in the 2D path. Figure 16 2d-overshoot point (the point with maximum distance of zero) of a joint system consisting of two standard 2nd order linear subsystems are drawn. Assuming $\omega_{n1} = \omega_{n2}$ this will be just in the corner of rectangle, but when $\omega_{n1} \neq \omega_{n2}$ this point will be on one of the sides.

$$H_1(s) = \frac{\omega_{n1}^2}{s^2 + 2\zeta_1\omega_{n1}s + \omega_{n1}^2}$$

$$H_2(s) = \frac{\omega_{n2}^2}{s^2 + 2\zeta_2\omega_{n2}s + \omega_{n2}^2}$$

In figure 17 the 2d-overshoot of different systems with different $\zeta_1 = \zeta_2$ is shown. It shows that systems which their natural frequencies of their subsystems are equal have the worse 2d-overshoot: $\sqrt{2}(O_{1d})$ (It is also proven with some other experiments). For example when $\omega_{n1} = 1, \omega_{n2} = 1, \zeta_1 = \zeta_2 = .6$, $O_{2d} = 1.5482$, but for $\omega_{n1} = 1, \omega_{n2} = .5, \zeta_1 = \zeta_2 = .6$, $O_{2d} = 1.4767$.

6 Conclusions

In this report we derived complete equations of ball-on-plate system. We did linearisation analysis to find the transfer function of the system. We designed several compensator and at the rest made some analysis on the results. We tried several trajectory tracking tasks. The final results showed that the designed controllers had almost acceptable performances.

7 Acknowledgement

The authors would like to thank all their classmates, friends and masters that helped on this project. They would like to thank Ehsan Omidi and Farokh Atashzar for their illuminating discussion and ideas. They also knowledge Dr.H.A.Talebi for his comprehensive and off course perfect instructions.

References

- [1] N. Mohajerin, M. B. Menhaj, and A. Doustmohammadi, "A reinforcement learning fuzzy controller for the ball and plate system," in *Proc. IEEE Int Fuzzy Systems Conference*, 2010, pp. 1–8.
- [2] S. Awtar, C. Bernard, N. Boklund, A. Master, D. Ueda, and K. Craig, "Mechatronic design of a ball-on-plate balancing system," *Mechatronics*, vol. 12, no. 2, pp. 217–228, 2002.
- [3] X. Dong, Z. Zhang, and C. Chen, "Applying genetic algorithm to on-line updated pid neural network controllers for ball and plate system," in *Proc.*

- Fourth Int Innovative Computing, Information and Control (ICICIC) Conf*, 2009, pp. 751–755.
- [4] X. Fan, N. Zhang, and S. Teng, “Trajectory planning and tracking of ball and plate system using hierarchical fuzzy control scheme,” *Fuzzy Sets and Systems*, vol. 144, no. 2, pp. 297–312, 2004.
 - [5] D. Liu, Y. Tian, and H. Duan, “Ball and plate control system based on sliding mode control with uncertain items observe compensation,” in *IEEE International Conference on Intelligent Computing and Intelligent Systems, ICIS09*, vol. 2. IEEE, 2009, pp. 216–221.
 - [6] K. kyu Lee, G. Btz, and D. Wollherr, “Basketball robot: Ball-on-plate with pure haptic information,” in *IEEE International Conference on Robotics and Automation*, May 2008.
 - [7] D. Yuan and Z. Zhang, “Modelling and control scheme of the ball-plate trajectory-tracking pneumatic system with a touch screen and a rotary cylinder,” *Control Theory & Applications, IET*, vol. 4, no. 4, pp. 573–589, 2010.
 - [8] F. Golnaraghi and B. Kuo, *Automatic control systems*, vol. 2.
 - [9] R. Dorf and R. Bishop, *Modern control systems*, 11th ed. Pearson Printice Hall, 2008.
 - [10] W. Bolton, *Mechatronics: electronic control systems in mechanical and electrical engineering*. Prentice Hall, 2004.
 - [11] D. Haessig and B. Friedland, “On the modeling and simulation of friction,” in *American Control Conference*. IEEE, 2009, pp. 1256–1261.
 - [12] C. S.J. and T. R.W., “On the modeling of coulomb friction,” *UNKNOWN JOURNAL OR CONFERENCE*, 1998.

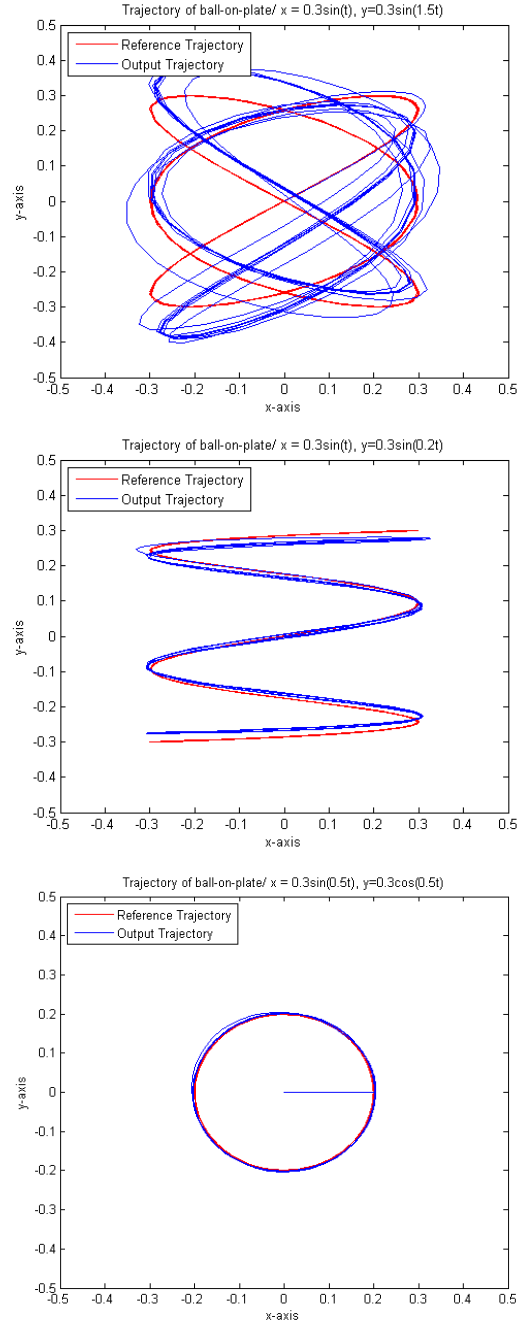


Figure 14: Results of trajectory tracking.

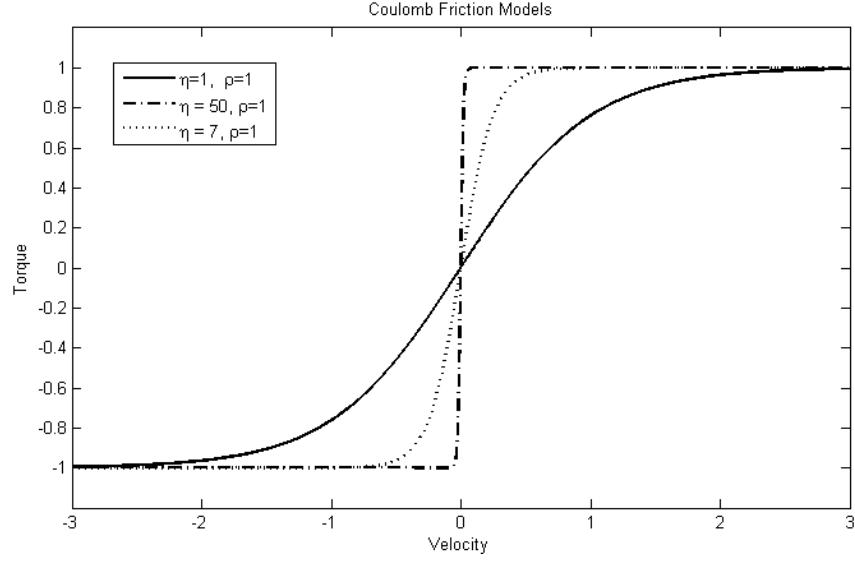


Figure 15: Coulomb Friction Models.

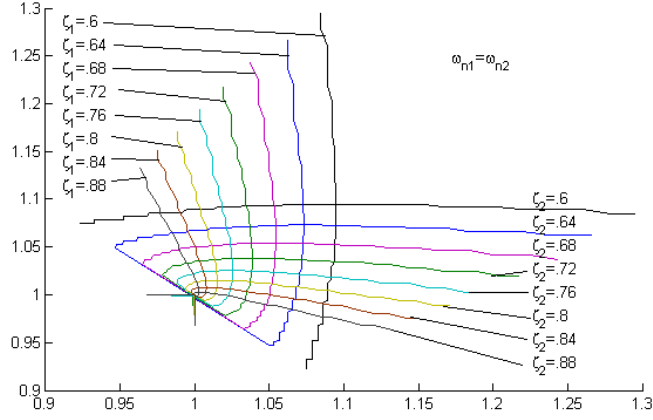


Figure 16: 2-D overshoot place. This figure shows the point which has the maximum distance from zero for different parameters of ζ_1 and ζ_2 . Each curve is a class of systems which have one subsystem and the other subsystem differs.

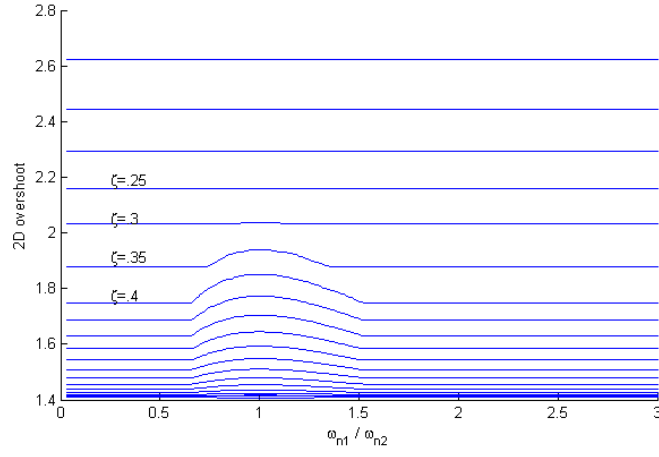


Figure 17: 2-D overshoot. This figure shows the the maximum distance from zero for different systems with $\zeta_1 = \zeta_2$ according to their subsystems natural frequencies ratios.