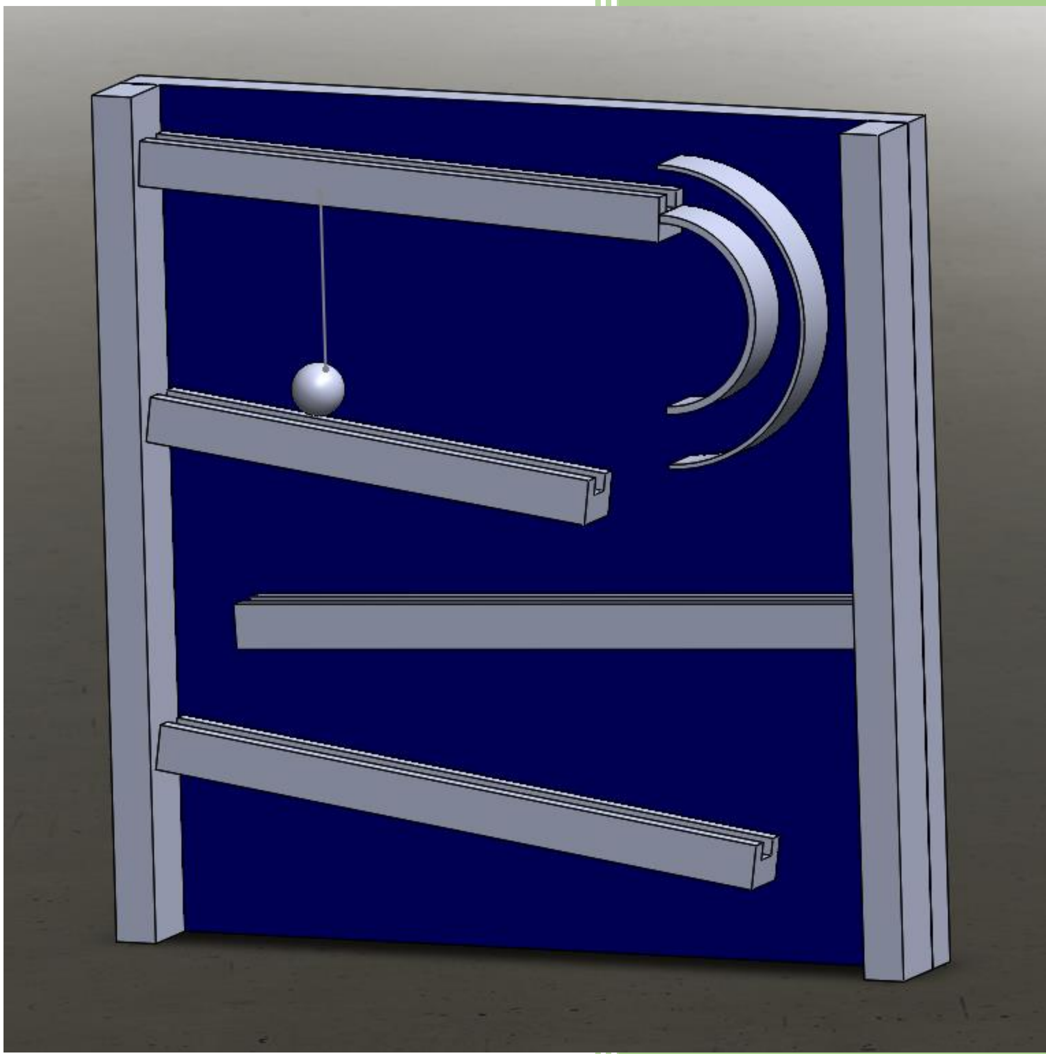
2014

MSE 222 DYNAMICS PROJECT



Sukrut Daga 301091518

Michael Szperkowicz 301180960

Kirill Salita 301209852

Devpreet Bhullar 301217638

Contents

Project Description

The main goal of this project is to design, build, analyze and test a dynamic system that would let a ball travel from the top of the system to the bottom while achieving a time between the two points to be approximately four seconds. The system will be on a board with an area of 12 inch x 12 inch (1 foot x 1 foot). The ball will have to go through a semi-circular tunnel and then hit a pendulum and bounce back (to change direction) and drop off twice from one board to another.

These requirements were achieved by making several diagrams of what the system would look like, calculations were done in Matlab with equations of motion, velocity and acceleration with respect to time and choice of materials for the system were taken into consideration as this would be critical for the system to work as we would want it to.

Part I: Analysis of each component of the system

Initial energy stored in the spring

$$E = \frac{1}{2}kx^2$$

E = spring energy,

k = spring constant,

x = displacement of the spring

Spring energy is converted to kinetic energy and potential energy when the ball is launched (assuming the ball is skipping without rolling):

$$E = \frac{1}{2}mV^2 + mg\Delta h$$

E = spring energy,

m = mass of the ball,

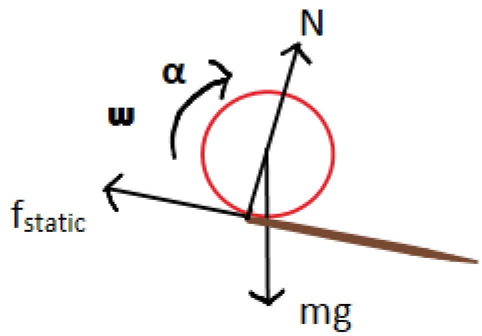
V = launch velocity of the ball,

g = gravity,

Δh = vertical distance the ball travelled.

Ramp

When the ball rolls on a ramp with some kind of slope, it can either roll without slipping or roll with slipping on the ramp. We analyze ramps for each case separately.



Case I: Rolling without slipping

We find Cartesian components of given initial velocity as shown below:

(Note: All symbols and their corresponding terms used from here on in the equations can be found in the appendix at the end of this report.)

$$u_x = u * \cos(\theta_i)$$
$$u_y = u * \sin(\theta_i)$$

Deriving an equation for angular acceleration of the center of gravity of the ball:

$$\sum M_{IC} = \bar{y} * ma_x + \underbrace{\cancel{\bar{x} * ma_y}}_{0} + I_G\alpha$$

$$\underbrace{r * mg * sin\theta}_{M_{IC}} = \underbrace{rm}_{ym} * \underbrace{\alpha r}_{a_x} + I_G\alpha$$

$$r * mg * sin\theta = \alpha(r^2m + I_G)$$

$$|\alpha|^* = \frac{r * mg * \sin(\theta_i)}{mr^2 + I_g}$$

*We find the absolute value first, then empirically improvise each time to find the right sign (+/-). Further details can be found in the Matlab code itself.

Including the effect of rolling friction on velocity of the ball when it is rolling down a ramp as follows:

$$|\alpha| = r_f \; \frac{r * mg * \sin(\theta_i)}{mr^2 + I_g}$$

We introduced this rolling friction parameter to take rolling friction into account when ball rolls down a ramp, for more details please see the corresponding section in Part III of this report.

Acceleration can be calculated as follows:

$$a_x = |\alpha r| * \cos(\theta_i)$$

$$a_y = |\alpha r| * \sin(\theta_i)$$

Angular velocity can be calculated as follows:

$$\omega = \omega_0 + \alpha t$$

Velocity of the center of gravity of the ball can be calculated as follows:

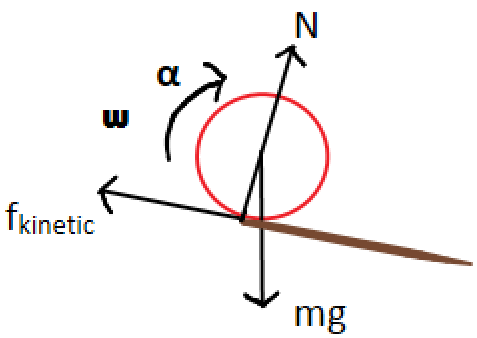
$$v_x = u_x + a_x t$$

$$v_y = u_y + a_y t$$

Position of the center of gravity of the ball can be calculated as follows:

$$x = x_0 + v_x t$$
$$y = y_0 + v_y t$$

Case II: Rolling with slipping



Using initial velocity of the ball as the initial condition, we can find its Cartesian components as:

$$u_x = u * \cos(\theta_i)$$
$$u_y = u * \sin(\theta_i)$$

Frictional force experienced by the ball during slipping:

$$F_f = |\mu_k mg cos(\theta_i)|$$

Acceleration due to frictional force:

$$|a_{f,x}|^* = \left|\frac{F_f \cos(\theta_i)}{m}\right|$$

$$|a_{f,y}|^* = \left|\frac{F_f \sin(\theta_i)}{m}\right|$$

Total acceleration of the ball can be calculated by adding acceleration due to gravity and deceleration due to friction as follows:

$$a_x = g * \sin(\theta_i) * \cos(\theta_i) + a_{F,x}$$
$$a_y = g * sin^2(\theta_i) + a_{F,y}$$

Angular velocity can be calculated as follows:

$$\omega = \omega_0 + \alpha t$$

Velocity of the center of gravity of the ball can be calculated as:

$$v_x = u_x + a_x t$$
$$v_y = u_y + a_y t$$

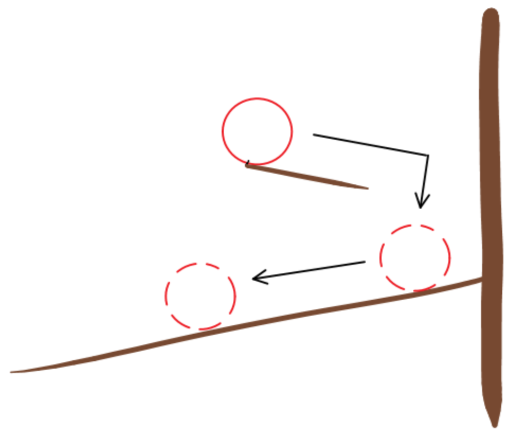
Position of the center of gravity of the ball can be calculated as:

$$x = x_0 + v_x t$$
$$y = y_0 + v_y t$$

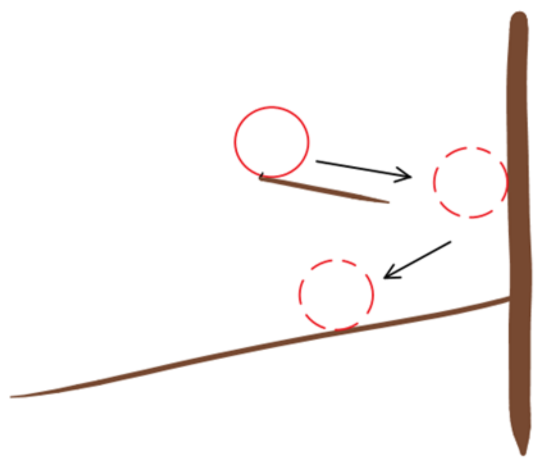
Calculations for free fall:

Two things can happen when the ball falls from the ramp:

1. Ball drops to the ramp below without hitting the vertical wall.



2. Ball hits the vertical wall and then drops.



Case I: ball drops to the ramp below without hitting the vertical wall:

We model the motion of the ball in this case as projectile motion. Since the ball is not in contact with any surface and it doesn't hit the vertical wall, we can make some deductions about the motion of the ball as shown below.

Acceleration of the center of gravity of the ball:

$$a_y = g$$

$$a_x = 0$$

Angular acceleration of the ball:

$$\alpha = 0$$

Therefore, velocity of center of gravity of the ball can be calculated as:

$$v_y = u_y + gt$$

$$v_x = u_x$$

Position of center of gravity of the ball:

$$x = x_0 + v_x t$$

$$y = y_0 + v_y t$$

Case II: The ball hitting the wall and before dropping on the ramp

Calculations for this case are done in a similar manner as the first one but we also consider the role of the coefficient of restitution along with projectile motion to completely analyze the motion of the ball.

Similar to case 1, acceleration of the center of gravity of the ball is:

$$a_y = g$$

$$a_x = 0$$

Angular acceleration of the ball:

$$\alpha = 0$$

We use coefficient of restitution to calculate the effect of collision with vertical wall on horizontal velocity of the ball. While the expression for vertical velocity remains the same as the first case.

$$v_y = u_y + gt$$

$$v_x = -e_w u_x$$

Position of center of gravity of the ball:

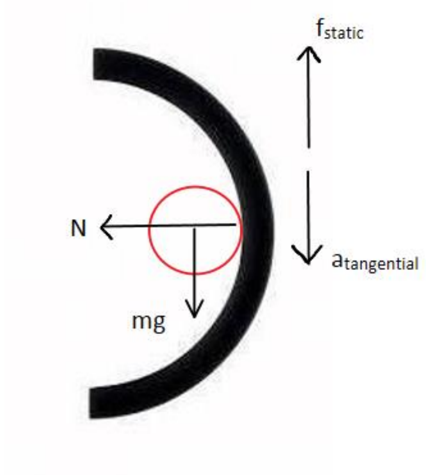
$$x = x_0 + v_x t$$

$$y = y_0 + v_y t$$

Semi-Circular path

The plastic ball travels into a semi-circular path. There can be two possibilities. One is the ball slips on the path and one where it does not slip and just rolls.

Case I: Rolling without slipping on a semi-circular path



Assuming that only x-axis component of velocity exists initially when the ball enters the semi-circular path, we can write the initial Cartesian components of the initial velocity as follows

$$u_x = u * \cos(0)$$

$$u_y = u * \sin(0)$$

Coordinates of position of center of gravity of ball with respect to center of the semi-circular path:

$$x_{bc} = x - x_c$$

$$y_{bc} = y - y_c$$

Polar angle (position of the ball on the semi-circular path represented as an angle between the positive x-axis and position vector of the ball):

$$\theta = tan^{-1}(\frac{y_{bc}}{x_{bc}})$$

We can calculate slope of small angle approximated ramp as follows:

$$\theta_i = \theta - \frac{\pi}{2}$$

Angular acceleration of the ball:

$$|\alpha|^* = \frac{r * mg * \sin(\theta_i)}{mr^2 + I_g}$$

Angular velocity of the ball:

$$\omega = \omega_0 + \alpha t$$

Calculating acceleration from α:

$$a_{x\alpha} = r\alpha * \cos(\theta_i)$$

$$a_{y\alpha} = r\alpha * \sin(\theta_i)$$

Centripetal acceleration:

$$a_c = \frac{u_x^2 - u_y^2}{R}$$

Acceleration of the center of gravity of the ball:

$$a_x = a_{x0} + a_c cos\theta$$

$$a_y = a_{y0} + a_c sin\theta$$

Velocity of the center of gravity of the ball

$$v_x = u_x + a_x t$$

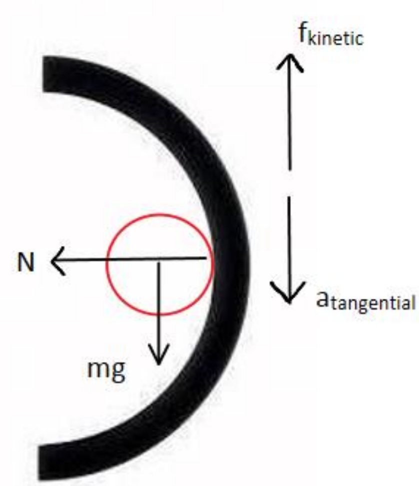
$$v_y = u_y + a_y t$$

Position of the center of gravity of the ball

$$x = x_0 + v_x t$$

$$y = y_0 + v_y t$$

Case II: Rolling with slipping on a semi-circular path



We can use the same steps for slipping as we used for rolling to find slope of each slope approximated using small angle approximation.

Centripetal acceleration can be calculated as:

$$a_c = \frac{u_x^2 - u_y^2}{R}$$

Frictional force during slipping can calculated as:

$$F_f = \mu_k m(a_c + gcos\theta_i)$$

Angular acceleration of the ball:

$$|\alpha|^* = \frac{r * F_f}{I_g}$$

Angular velocity of the ball:

$$\omega = \omega_0 + \alpha t$$

Deceleration due to friction:

$$|a_{f,x}|^* = \left|\frac{F_f \cos(\theta_i)}{m}\right|$$

$$|a_{f,y}|^* = \left|\frac{F_f \sin(\theta_i)}{m}\right|$$

Acceleration of the center of gravity of the ball:

$$a_x = g * \sin(\theta_i) * \cos(\theta_i) + a_{F,x} - a_c cos\theta$$
$$a_y = g * sin^2(\theta_i) + a_{F,y} - a_c sin\theta$$

Velocity of the center of gravity of the ball:

$$v_x = u_x + a_x t$$
$$v_y = u_y + a_y t$$

Position of the center of gravity of the ball:

$$x = x_0 + v_x t$$
$$y = y_0 + v_y t$$

Part II: Design, Simulation and Study of Our Dynamic System

System Design

After mind-numbing planning, we assembled every component of the system on our main board. The system is made out of wood and a PVC semi-circular pipe for the ball to roll down.

The image below will show our final version along with some Solidworks models to follow.

Figure: Final System Design

Solidworks Design and Dimensions (Inches and Degrees)

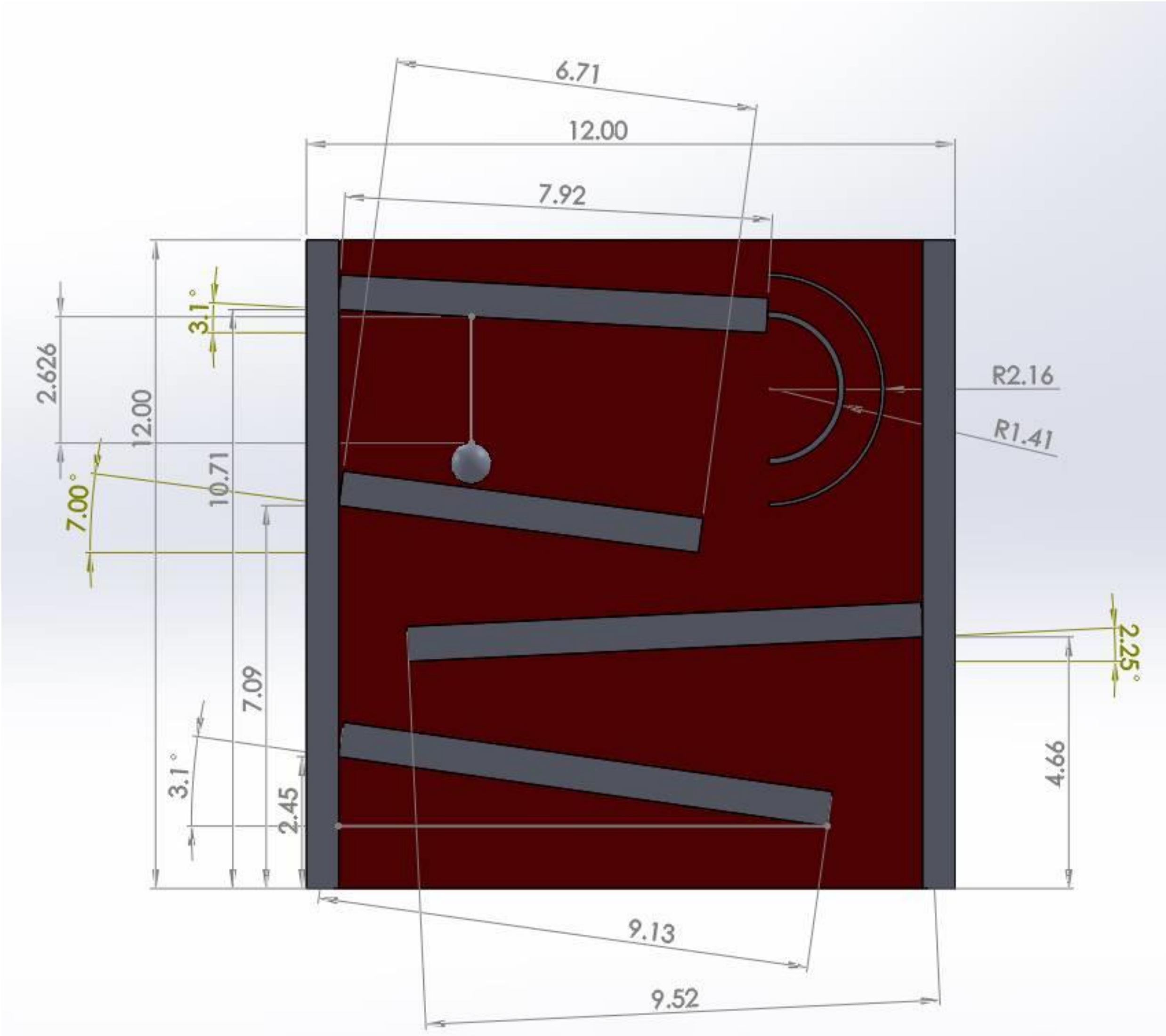
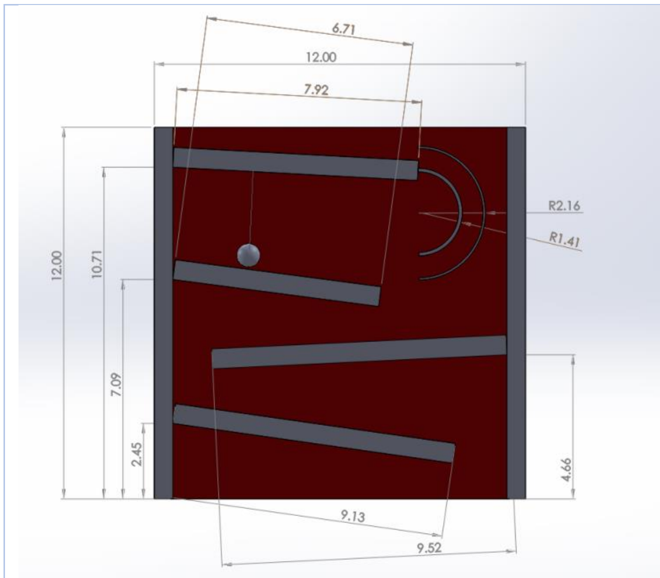
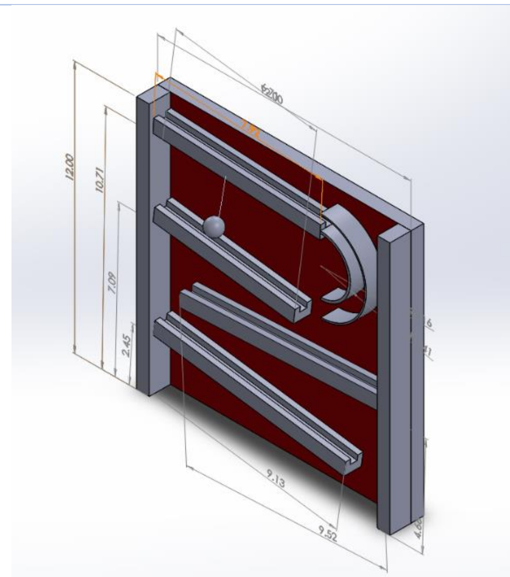


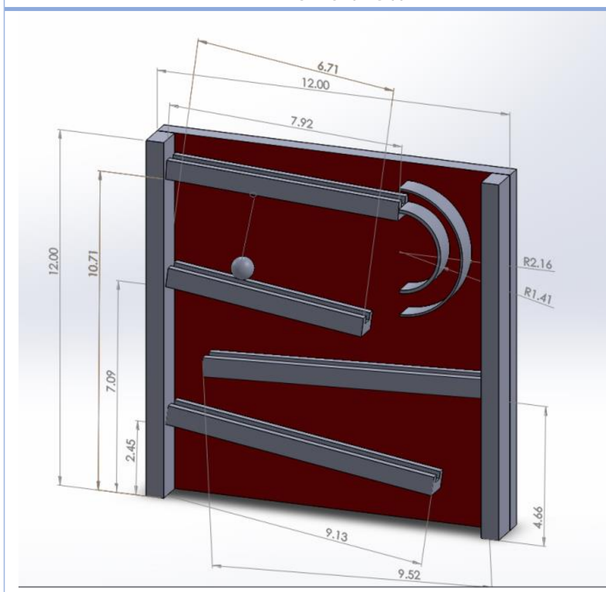
Figure: Solidworks System Design



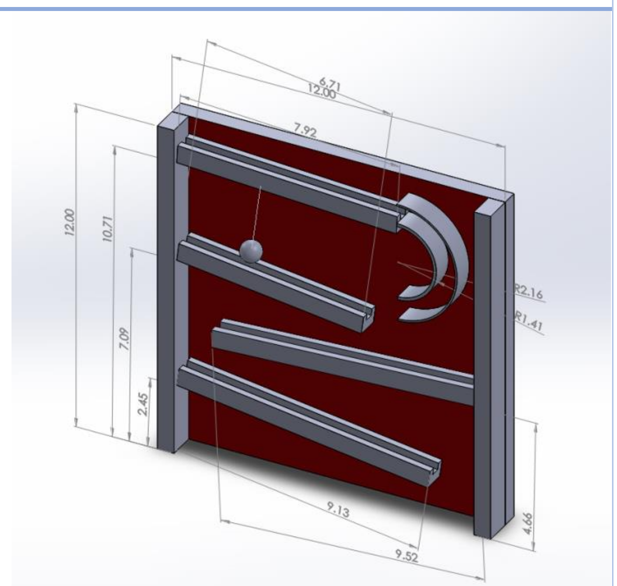
Front view



Isometric view



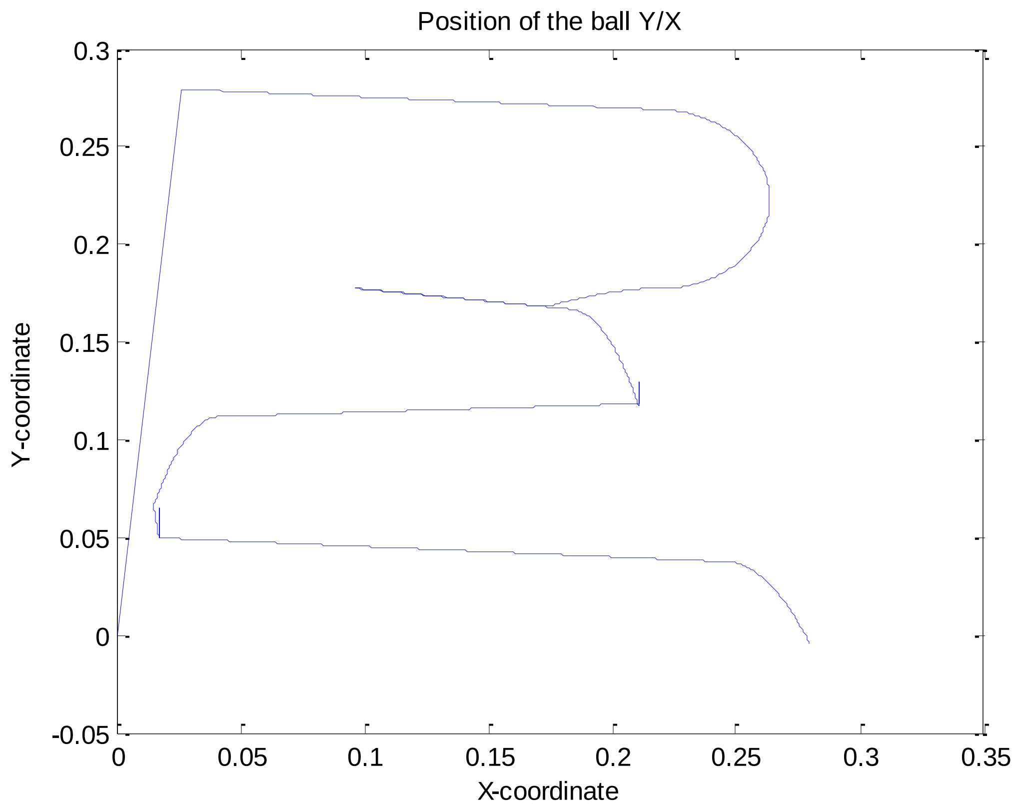
Dimetric view

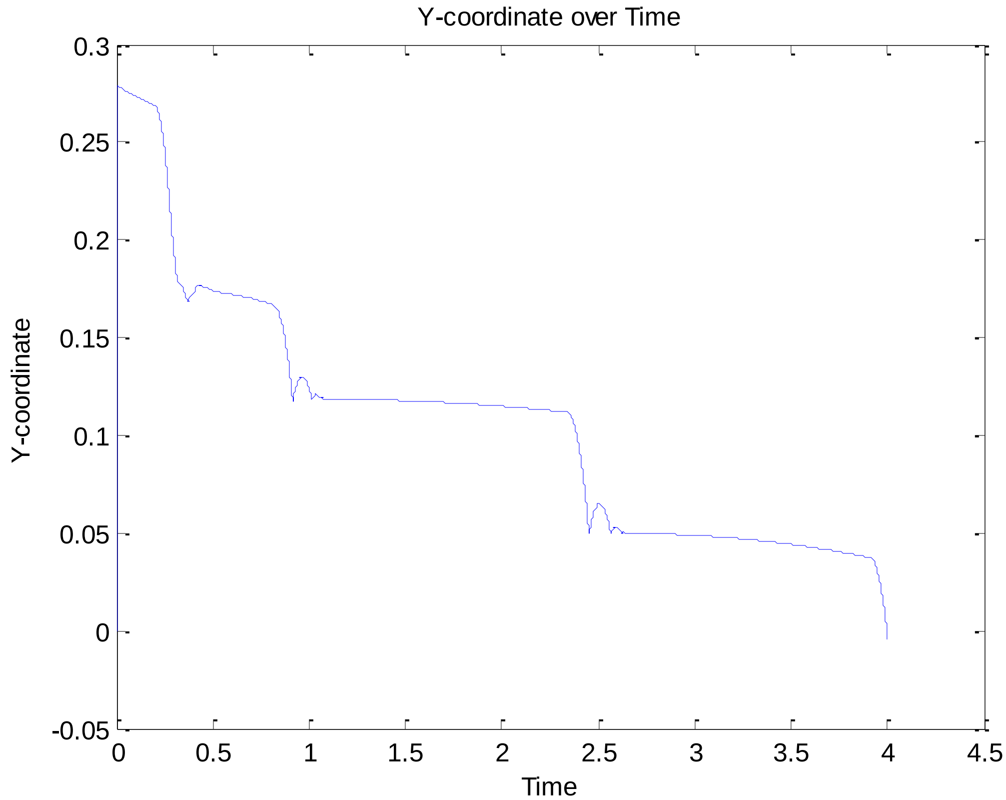


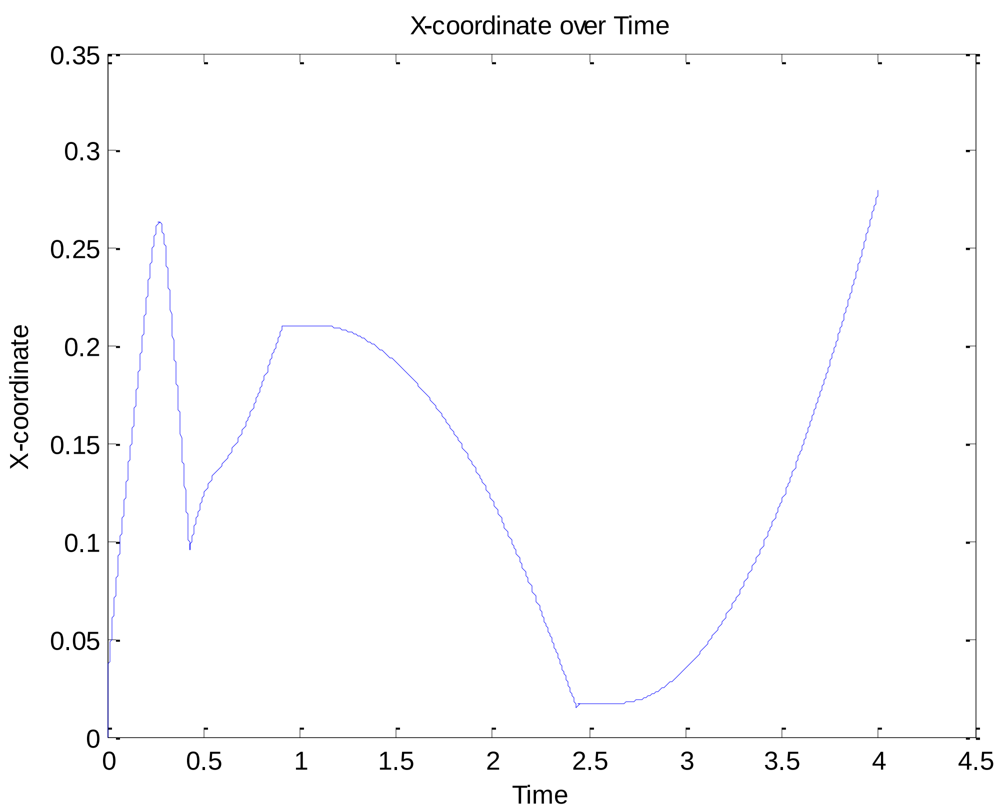
Trimetric view

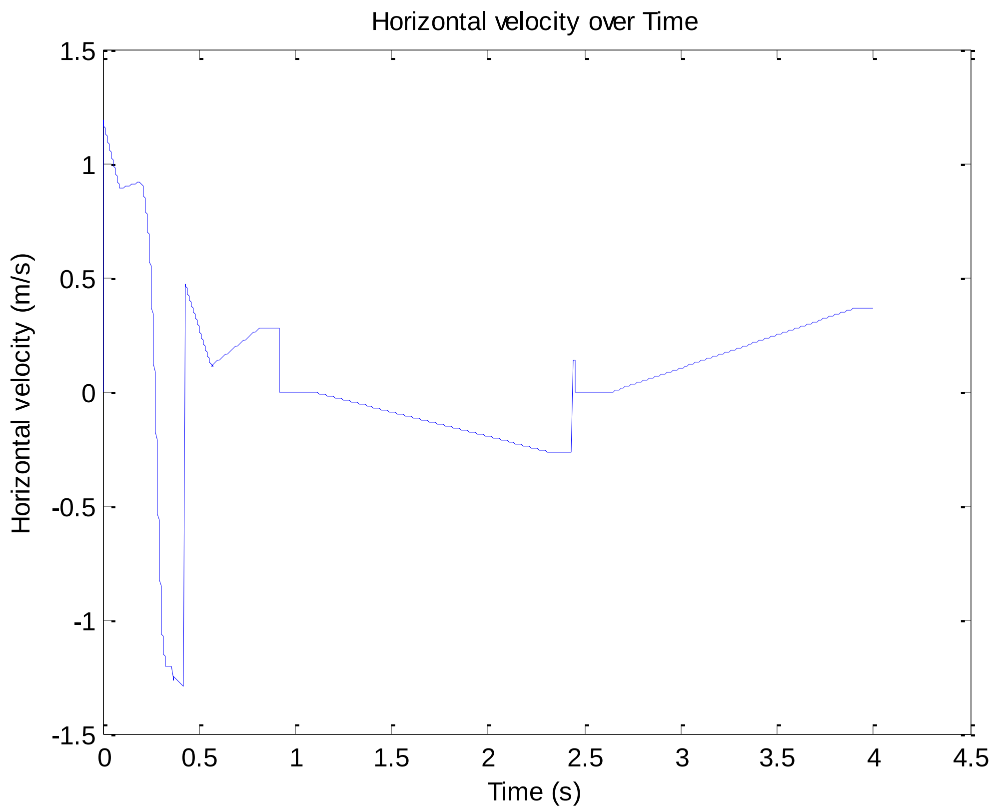
Figure: Multiple Views of the Design

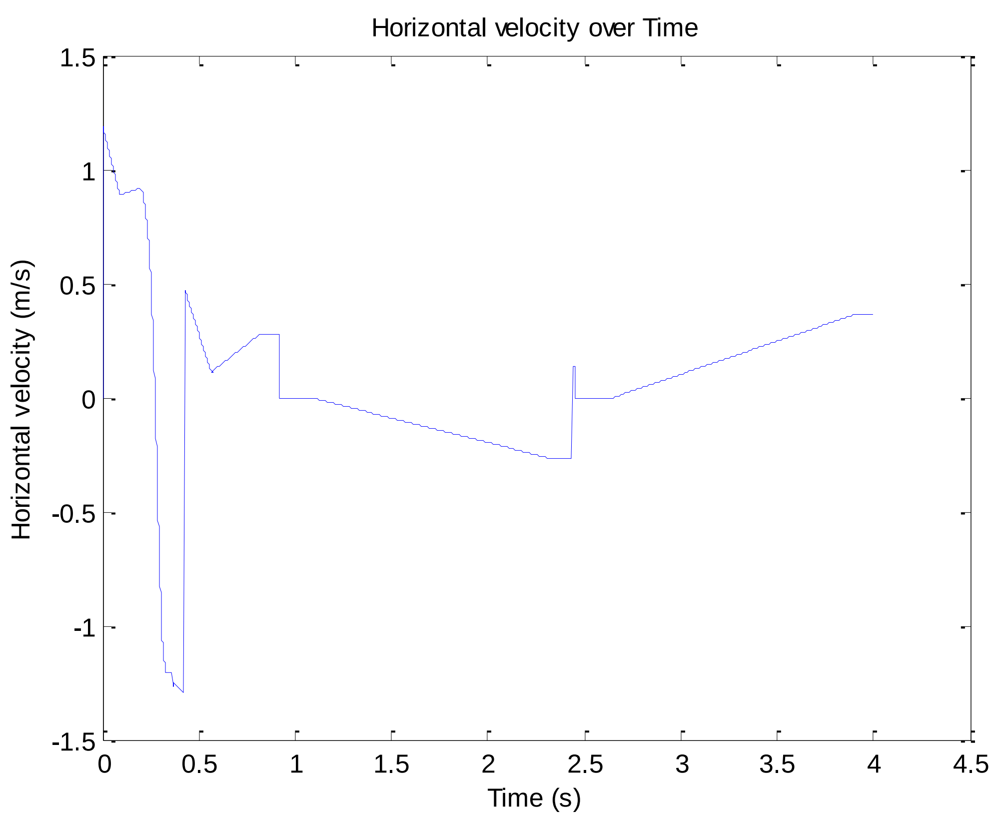
Matlab Plots

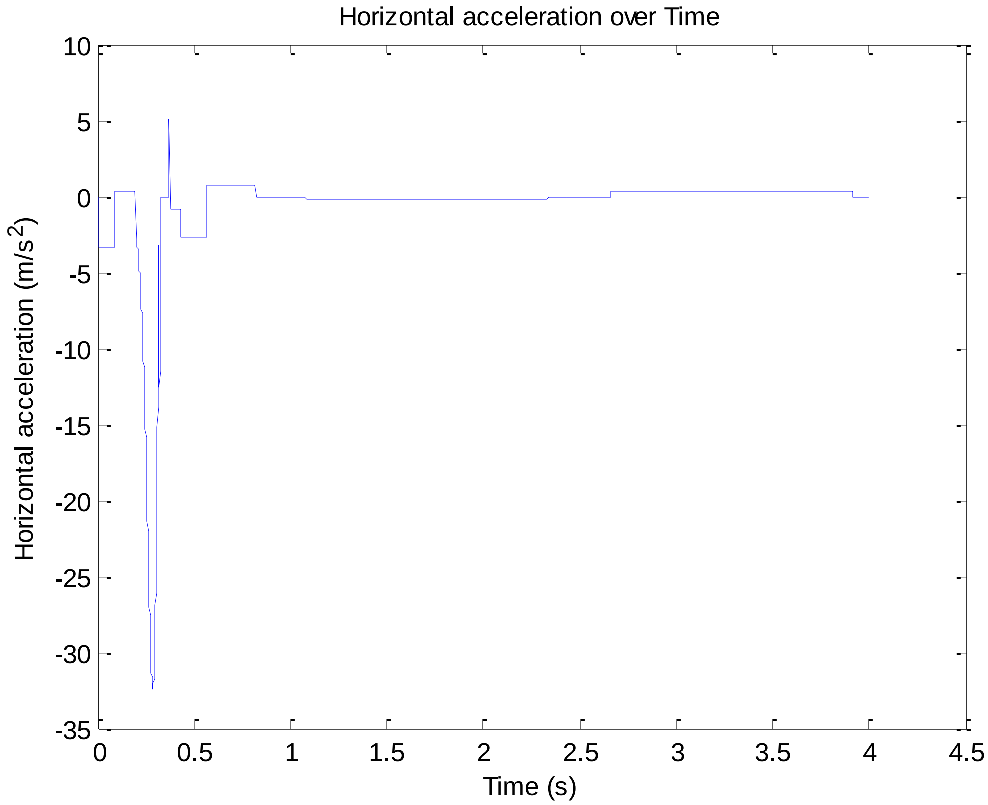


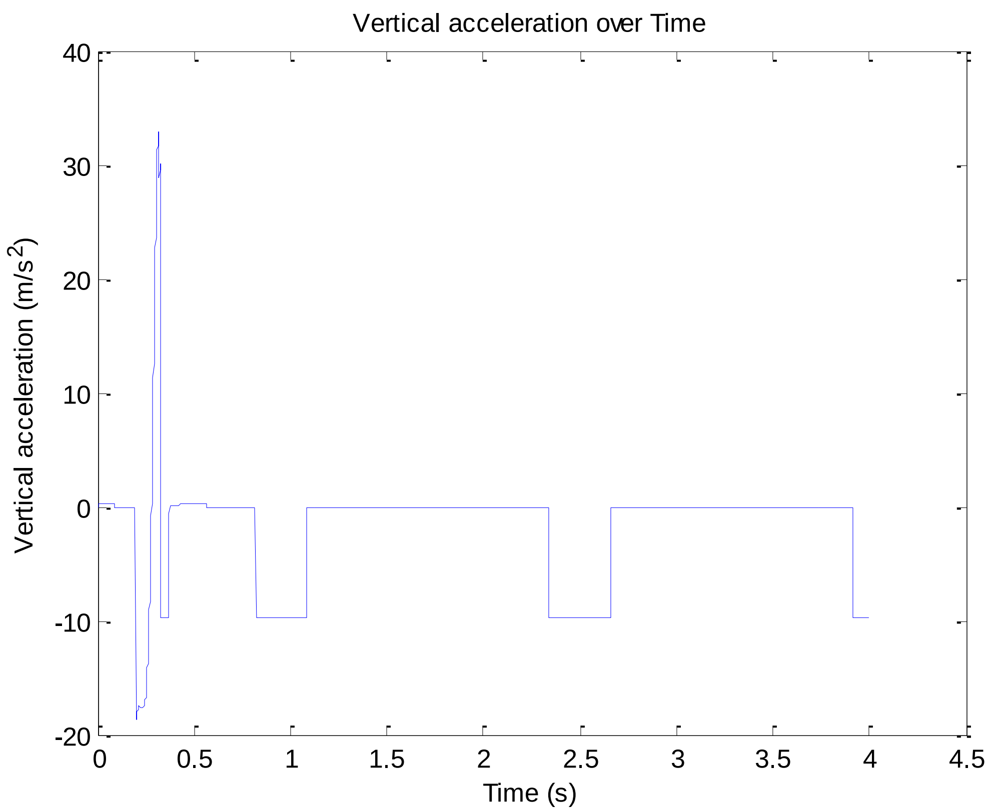

Y-coordinate over Time
Y-coordinate
Time
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.05
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5

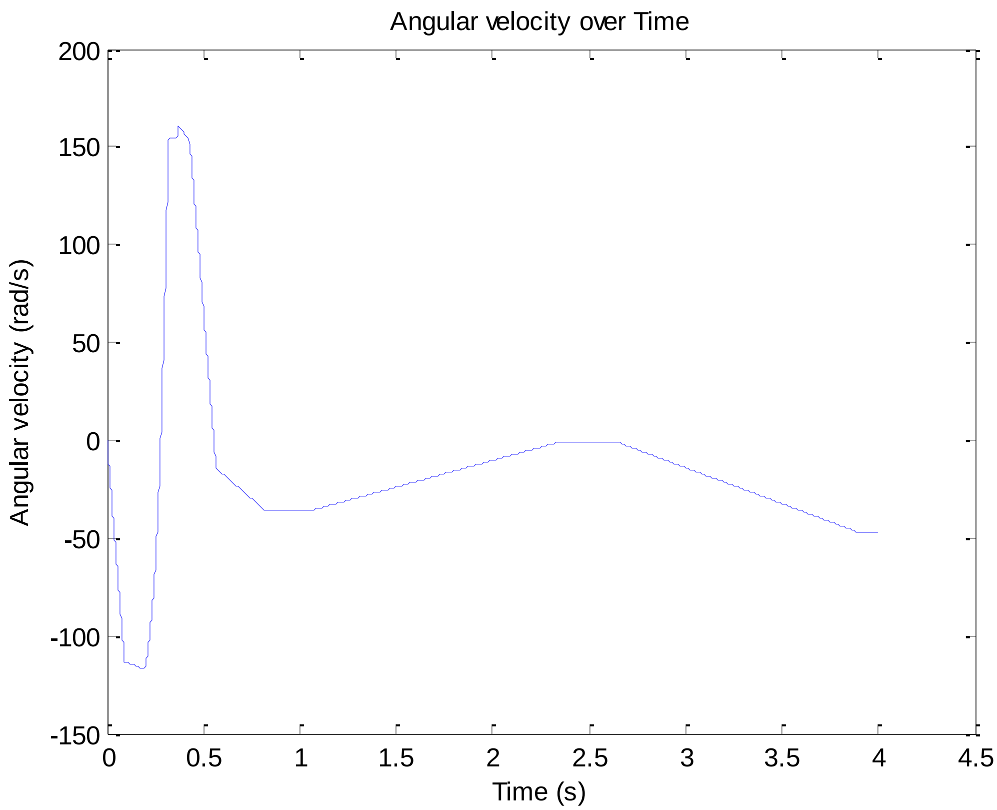

X-coordinate over Time
X-coordinate
Time
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5

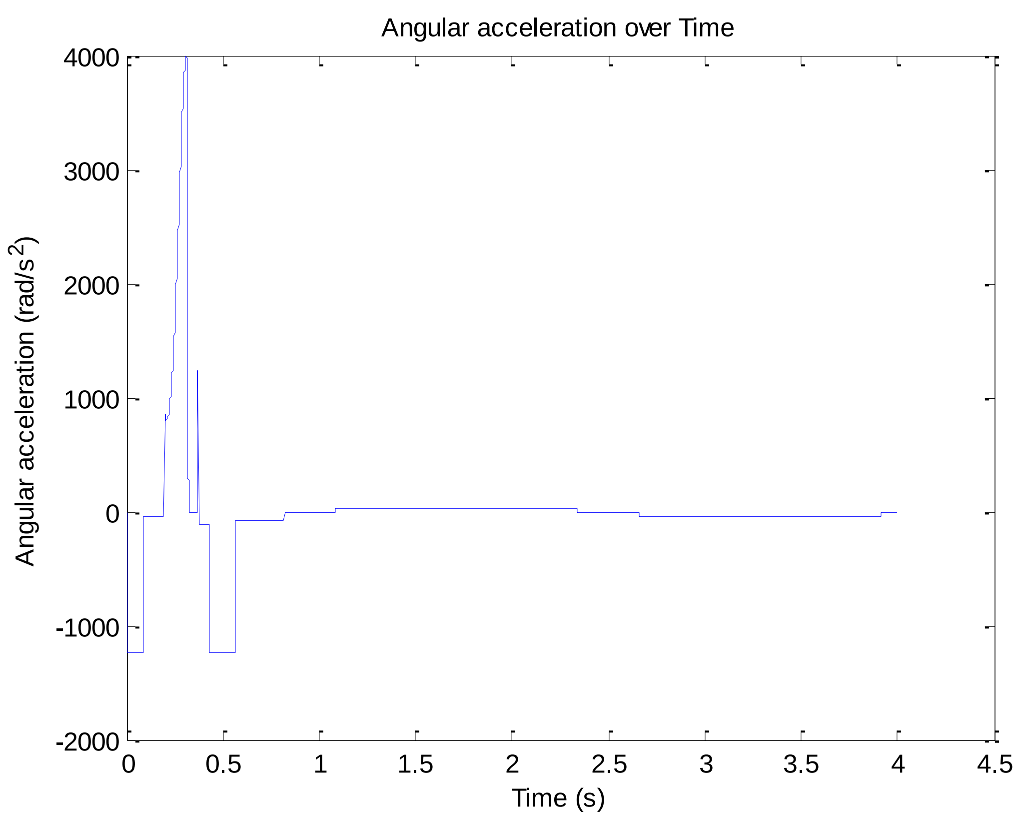

Horizontal velocity over Time
1.5
1
0.5
0
-0.5
-1
-1.5
Horizontal velocity (m/s)
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
Time (s)


Horizontal velocity over Time
1.5
1
0.5
0
-0.5
-1
-1.5
Horizontal velocity (m/s)
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
Time (s)


Horizontal acceleration over Time
Horizontal acceleration (m/s2)
10
5
0
-5
-10
-15
-20
-25
-30
-35
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
Time (s)


Vertical acceleration over Time
Vertical acceleration (m/s2)
40
30
20
10
0
-10
-20
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
Time (s)


Angular velocity over Time
Angular velocity (rad/s)
200
150
100
50
0
-50
-100
-150
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
Time (s)


Angular acceleration over Time
Angular acceleration (rad/s^2)
4000
3000
2000
1000
0
-1000
-2000
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
Time (s)

Sensitivity testing

We changed each parameter by ±10% in the matlab code and recorded the total time to see how sensitive our system is with respect to each parameter. The following table summarizes sensitivity testing results:

Parameter	+10% result	-10%result
Coefficient of kinetic friction with wood	3.992s	4.028s
Coefficient of kinetic friction with PVC	4.013s	4.017s
Rolling friction	4.155s	3.876s
Spring constant	3.994s	4.028s
Spring compression	3.921s	4.039s
Coefficient of restitution with pendulum	3.934s	4.081s
Coefficient of restitution with wood	4.087s	3.937s
Mean time	4.0073s	
Standard Deviation	0.0712s	

Here we can see that our design is most sensitive to rolling friction, for all other parameters our time only changes by less than 100ms.

Part III: Component Sourcing and Characterization

Component Sourcing and Characteristics

1. Ball (no cost) – Provided by Dr. Carolyn Sparrey.
2. Finishing nails (no cost) – The nails were used to secure the ramps and the semi-circular PVC tube to the main board. A member of our group had them in his workshop. We can re-use them after the project is finished.
3. Cider wood (no cost) – The wood was used for most of the structure. The main frame and the ramps. A member of our group had the wood lying around in his workshop. He will re-use the wood after the project has finished.
4. PVC Tube (no cost) – We has half a PVC tube which we decided to use it for the semi-circular ramp. The tube was used instead of being thrown away. We managed to make good use of it.
5. Spring (no cost) – We used the spring to launch the ball into the system. A partner had a broken pen lying around.
6. Metallic Ball for Pendulum (no cost) – This was lying around the house and we struck the idea to use it for the collision of the plastic ball.

Assumptions

1. The ball enters the semi-circular path with zero vertical velocity.
2. The collision with the pendulum ball takes place on a perfectly horizontal plane.
3. Pendulum string is massless.
4. The ball is travelling in two dimensions (for simplifying calculation).
5. The ball is a perfect sphere.
6. Measurements for our co-ordinate system were based on approximation. Values were measured using a standard ruler.
7. Coefficients of friction for the semi-circular pipe (PVC) and cider wood ramps were taken from the internet (http://www.engineeringtoolbox.com).

Calculations for Design Parameters/Coefficients

1. Coefficient of Restitution of plastic ball and wood (e_w):

H (drop height) (m)	h (height at bounce) (m)	V_{Ball1} (m/s)	V_{Ball2} (m/s)	e_w
0.283	0.083	-2.35636585	1.27611128	0.541559063
0.35	0.096	-2.62049614	1.372413932	0.523722937
0.262	0.054	2.26725385	1.029310449	0.453989945

Table 1: Coefficient of Restitution (e_w) of plastic ball and wood.

Coefficient of restitution is used to represent ratio of velocities before and after an impact. The value falls between 0 and 1

Here were have a few values in the table described as follows:

1. H – Drop Height which represents the height from which the ball is dropped.

2. h – This represents the height to which the ball will bounce back up after the impact with the wood.
3. V_{Ball1} – This is the velocity of the ball right before the impact with the wood.

$$V_{Ball1} = \sqrt{2gH}$$

4. V_{Ball2} – This is the velocity of the ball right after the impact with the wood.

$$V_{Ball2} = \sqrt{2gh}$$

5. Coefficient of Restitution (e_w) – This is the ratio of two velocities calculated as follows:

$$e_w = \frac{-V_{Ball1}}{V_{Ball2}}$$

- The average value of coefficient of restitution (e_w) is 0.506423982.
- The standard deviation for the coefficient of restitution (e_w) is 0.046276646.

2. Coefficient of restitution of plastic ball

Run	U_a (m/s)	V_a (m/s)	U_b (m/s)	U_b (m/s)	e_p
1	-0.6416	0.2243	0	-0.1149	0.528678
2	-0.5161	0.2023	0	-0.0983	0.582445
3	-0.7223	0.2636	0	-0.1154	0.524713

Table 2: Coefficient of restitution (e_p) with plastic ball and metallic pendulum.

This is the second table of coefficient of restitution and this time it gives us the ratio of velocities before and after impact with the metallic ball of pendulum.

1. U_a – Velocity of plastic ball before collision
2. U_b – Velocity of metallic ball before collision
3. V_a – Velocity of plastic ball after collision
4. V_b – Velocity of metallic ball after collision
5. e_p – Coefficient of restitution with ratio of velocities before and after impact with the metallic ball of pendulum.

$$e_p = \frac{V_b - V_a}{U_b - U_a}$$

- The average value of coefficient of restitution is 0.545279.
- The standard deviation of coefficient of restitution is 0.032248.

3. Spring constant (k):

Length after compression (m)	Uncompressed Length (m)	Compression (Δ x) (m)	Load (N)	Average of Δx (m)
0.00745	0.01877	0.01132	0.55	0.01181
0.00663		0.01214		
0.0068		0.01197		
0.0102	0.01877	0.00857	0.39	0.00850667
0.0104		0.00837		
0.01019		0.00858		
0.01097	0.01877	0.0078	0.20	0.0078
0.01107		0.0077		
0.01087		0.0079		
0.01158	0.01877	0.00719	0.14	0.0072
0.01157		0.0072		
0.01167		0.0071		

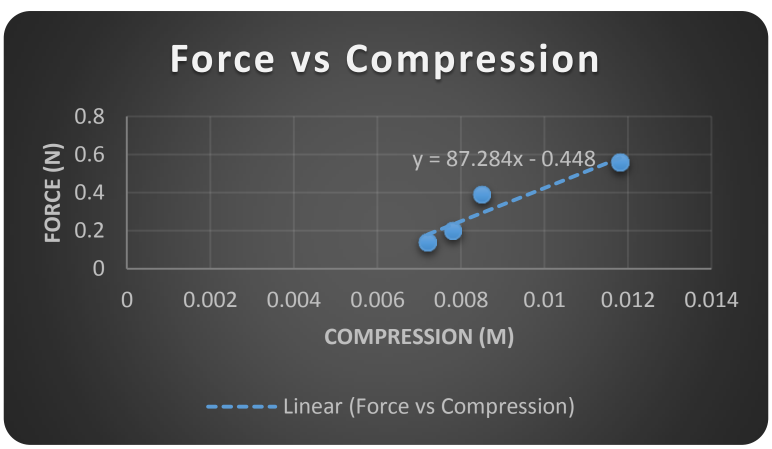
Table 3: Deformation

- Length of uncompressed spring is 18.77mm.

The above table contains the following values:

1. Length after compression – The length of spring after it was suspended with load attached.
2. Uncompressed Length – The length of the spring with no load, compression or extension.
3. Compression – The difference between uncompressed length and compressed length of the spring.
4. Load – The weight of the load attached to the spring
5. Average of Δx – Average of values obtained by computing the difference of uncompressed length and compressed length.

Graph: Force vs Compression



- The slope of the line is the value of spring constant (k in N/m). This is calculated using the following formula.

$$k = \frac{F}{x}$$

4. Rolling friction parameter (r_f)

At first out matlab code didn't account for any rolling friction, but after we noticed that the time it takes the ball takes to roll down a ramp in simulation is significantly less than the actual time, we decided to experimentally find a parameter which could make our simulation better by accounting for rolling friction. The ball rolls on a horizontal wooden ramp followed by an inclined wooden ramp. We recorded the motion of the ball and analyzed it using LoggerPro to find the linear acceleration of the ball on the inclined plane which was then used to calculate the angular acceleration.

Figure: A picture of the setup we used to find the rolling friction parameter

Run	α(rad/s^2)	α_c (rad/s^2)	r_f
1	55.6	48.8	0.88
2	51.6	39.2	0.76
3	48.8	40	0.82
4	62.8	52	0.83
5	46.4	40	0.86
6	56.4	38.4	0.68
7	81.2	56	0.69
8	49.6	36.8	0.74
9	49.6	37.6	0.76
10	56.4	44	0.78
Mean r_f			0.78
Standard deviation			0.06

Table 4: Rolling friction (r_f)

We calculate angular acceleration from linear acceleration as:

$$\alpha = \frac{a}{r}$$

1. α = angular acceleration
2. a = linear acceleration
3. r = radius of the ball

We define the rolling friction parameter as follows:

$$r_f = \frac{\alpha_c}{\alpha} \; OR \; \alpha_c = r_f \alpha$$

1. α – angular acceleration of the ball without rolling friction (generated from previous version of matlab code)
2. V_c – Real angular acceleration of the ball (found using LoggerPro)
3. r_f – Rolling friction parameter

- The average value of the rolling friction parameter is 0.78.
- The standard deviation of the rolling friction parameter is 0.06.

Observations, Recommendations and Limitations

1. The ball didn't roll on the ramps as predicted in the matlab code because we didn't account for any rolling friction.
- We solved this problem by experimentally finding a rolling friction parameter which is multiplied with initial angular acceleration to get a final angular acceleration which takes into account the effects of rolling friction. While testing the sensitivity, we also found that our system was most sensitive to this parameter.
2. The semicircular covered path in the system makes it hard to tell whether the ball is rolling or slipping
- We can use a slow motion camera to precisely tell what it is doing by observing the ball movements.
3. The measurement of the tilt angle of the ramp is not accurate.
- A clinometer can be used for precise measurements of the tilting angles.
4. It is hard to tell whether the spring struck the ball at its center of gravity.
- The problem can be solved using a laser guided launch system and a perfectly spherical ball.
5. The ball is not a perfect sphere and it makes it challenging to simulate the system.
- A higher quality perfectly spherical ball can be used.
6. It is hard to measure the time when the ball reaches the end point.
- We can use a laser beam towards the end of the system to stop the time as the ball cuts it. This along with a laser guide launch system will give us very accurate times.
7. The spring doesn't always provide the same amount of energy to the ball, this affects the total time significantly and the class testing results confirm this limitation of the design. Our test times in the class were 4.2s, 4.5s, and 3.9s, we believe that these discrepancies could be caused by inconsistency of the spring.
- A better spring system can be used which ensures spring compression is always constant for each run.

Conclusion

The project gave us a thorough understanding on how we can apply the knowledge from dynamics class into real world situations. We simulated and crafted a real world model of our system.
We used advanced programs like Matlab, LoggerPro and Solidworks to visualize, simulate and analyze the system.
However as with every system, it is impossible to get precise values practically from what we calculated but we can get pretty close to them. We have assumed that the ball was a perfect sphere even though it is not. We can get very accurate reading of friction as the ball rolls or slips along the ramps and semicircular path. This changes the ball's acceleration, angular acceleration, angular velocity and the velocity of the ball. This in the end will also have an effect on the time the ball took to travel through the system.

Appendix

- E = spring energy,
- k = spring constant,
- x = displacement of the spring
- m = mass of the ball
- g = gravity
- t = time
- Δh = vertical distance the ball travelled
- u = initial velocity
- u_x = horizontal initial velocity
- u_y = vertical initial velocity
- θ_i = incline angle of ramp with respect to positive x-axis
- r = radius of the plastic ball
- I_g = moment of inertia of the plastic ball
- a_x = horizontal acceleration of the ball
- a_y = vertical acceleration of the ball
- ω = angular velocity
- ω_0 = initial angular velocity
- α = angular acceleration
- α_0 = initial angular acceleration
- v_x = horizontal final velocity
- v_y = vertical final velocity
- v = velocity of the ball
- r_f = rolling friction parameter
- x = x-coordinate of the final position of the ball
- y = y-coordinate of the final position of the ball
- x_0 = x-coordinate of the initial position of the ball
- y_0 = y-coordinate of the initial position of the ball
- F_f = force due to friction
- μ_k = coefficient of friction
- $a_{f,x}$ = acceleration due to frictional force in horizontal direction
- $a_{f,y}$ = acceleration due to frictional force in vertical direction
- e_w = coefficient of restitution of plastic ball and wood
- e_p = coefficient of restitution of plastic ball and metallic pendulum ball.
- x_{bc} = x-coordinate of position of the ball with respect to center of the semi-circular path
- y_{bc} = x-coordinate of position of the ball with respect to center of the semi-circular path
- θ = polar angle (position of the ball on the semi-circular path represented as an angle between the positive x-axis and position vector of the ball)
- a_c = centripetal acceleration
- R = radius of the semi-circular path