METR7203 Problem Based Assignment 2

Callum Rohweder s4357594

August 2017

1 Question 1

The inverted pendulum on a cart system is illustrated in figure 1. It can be seen that the coordinates of interest for this are p (linear position), and theta (the pendulum angle with respect to the upwards vertical axis). To control this system, an input F is applied on the cart, allowing it to compensate for the movement of the pendulum; altering the cart's position.
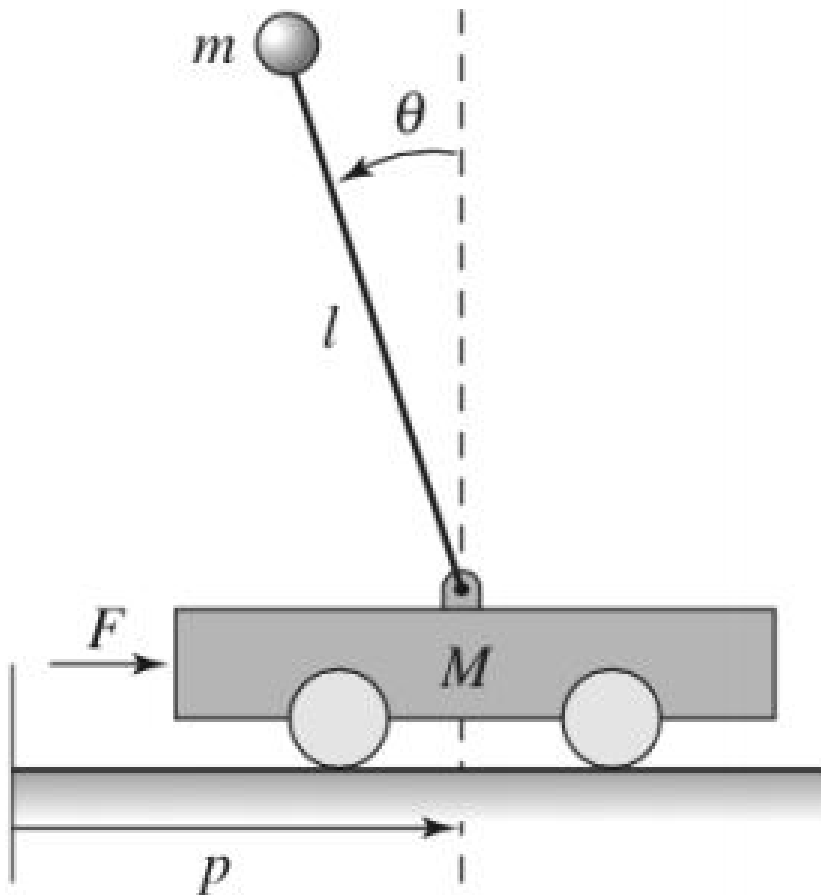

Figure 1: Inverted Pendulum on a cart diagram

The parameters of this system are:

→ the damping of the pendulum (ρ)
→ the damping of the cart (c)
→ the mass of the cart (M)
→ the mass of the pendulum as a point load (m) at distance (
→ the acceleration sure to gravity (g)
→ the position of the cart (p) and derivatives $(\dot{p}, \ddot{p})$
→ the angle of the pendulum (th) and derivatives $(\dot{\theta}, \ddot{\theta})$

Using Lagrange's equations, the equations of motion for this system can be expressed. The lagrangian (L) is kinetic energy taken away from potential energy, as such:

$$L = T - V$$

for this system, the kinetic energy of the point mass (m) is made up of vertical and horizontal components, where the velocity in the horizontal will be decreased due to the linking cart's velocity. As such, the kinetic energy of the system is;

$$K = \tfrac{1}{2}m(\dot{p} - l \times \dot{\theta}cos(\theta))^2 + \tfrac{1}{2}m(l \times \dot{\theta}sin(\theta))^2 + \tfrac{1}{2}M\dot{p}^2$$

and the potential energy is contributed from the pendulum falling with increasing theta, such that;

$$V = mglcos(\theta)$$

Langrange's equation states that:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial q_d}\right) - \frac{\partial L}{\partial q} = Q$$

where Q is the non-conservative forces acting on the system, and q is a coordinate system.

Thus, for position, p;

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{p}}\right) - \frac{\partial L}{\partial p} = F - c\dot{p}$$

thus, where;

$$\frac{\partial L}{\partial \dot{p}} = \frac{\partial}{\partial \dot{p}}\left(\tfrac{1}{2}m(\dot{p}^2 - 2\dot{p}l \times \dot{\theta}cos(\theta) + l^2 \times \dot{\theta}^2cos(\theta))\right) + M\dot{p}$$

$$= m(\dot{p} + l \times \dot{\theta}cos(\theta) + M\dot{p}$$

thus,

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{p}}\right) = m\ddot{p} - ml \times \ddot{\theta}cos(\theta) + ml \times \dot{\theta}^2sin(\theta) + M\ddot{p}$$

and

$$\frac{\partial L}{\partial p} = 0$$

thus the lagrangian about p is;

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

similarly for theta;

$$\frac{\partial L}{\partial \dot{\theta}} = \frac{1}{2}m\frac{\partial}{\partial \dot{\theta}}(\dot{p}^2 - 2\dot{p}l \times \dot{\theta}cos(\theta) + l^2 \times \dot{\theta}^2 cos(\theta)^2 + l^2 \times \dot{\theta}^2 sin(\theta)^2)$$

$$= \frac{1}{2}m(-2\dot{p}lcos(\theta) + 2l^2 \times \dot{\theta}cos(\theta)^2 + 2l^2 \times \dot{\theta}sin(\theta)^2)$$

$$= -m\dot{p}lcos(\theta) + l^2 m\dot{\theta}$$

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{\theta}}) = -m\ddot{p}lcos(\theta) + m\dot{p} \times \dot{\theta}lsin(\theta) + l^2 m\ddot{\theta}$$

and

$$\frac{\partial L}{\partial th} = \frac{1}{2}m\frac{\partial}{\partial th}(\dot{p}^2 - 2\dot{p}l \times \dot{\theta}cos(\theta) + l^2 \times \dot{\theta}^2 cos(\theta)^2 + l^2 \times \dot{\theta}^2 sin(\theta)^2 + 2glcos(\theta))$$

$$= \frac{1}{2}m(2\dot{p}l \times \dot{\theta}sin(\theta) + 2glsin(\theta))$$

$$= mlsin(\theta) \times \dot{\theta}p + gmlsin(\theta)$$

putting these together gives

$$\rightarrow -\rho \times \dot{\theta} = ml^2\ddot{\theta} - ml\ddot{p}cos(\theta) + ml\dot{p} \times \dot{\theta}sin(\theta) - mlsin(\theta) \times \dot{\theta}p - gmlsin(\theta)$$

thus the two equations of motion are:

$$-\rho \times \dot{\theta} = ml^2 \times \ddot{\theta} - mlcos(\theta)\ddot{p} - mglsin(\theta)$$

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

From here the state-space representation of the system can be obtained by small angle approximation, and rewriting the equations of motion in terms of $\frac{dx}{dt}$, where x is;

$$x_1 = p,\ x_2 = th,\ x_3 = \dot{p},\ \text{and}\ x_4 = \dot{\theta}$$

which gives:

$$\frac{dx_1}{dt} = x_3$$

$$\frac{dx_2}{dt} = x_4$$

$$\frac{dx_3}{dt} = \frac{F}{M} - \frac{C}{M}x_3 + \frac{mg}{M}x_2 - \frac{\rho}{l \times M}x_4$$

$$\frac{dx_4}{dt} = \frac{g}{l}x_2 + \frac{M}{M \times l} - \frac{c}{M \times l}x_3 + \frac{mg}{M \times l}x_2 - \frac{\rho(M+m)}{M \times l^2 \times m}x_4$$

These can now be written in state-space form. This result can also be computed using Matlab to compute the linearisation from the aforementioned equations of motion; by solving the Jacobian.

```matlab
%% METR7203 Assignment 3 Part 1
%% mass and cart system derivation script

%% declare symbolic variables;
% masses, length, dampening constants, gravity, linear position, angle
% theta, cart force
syms m M l c rho g p_dd p_d p th_dd th_d th F real

%% Lagrange
% Lagrangian = Kinetic Energy - Potential Energy
T = (1/2)*m*(p_d - l*th_d*cos(th))^2 + (1/2)*m*(l*th_d*sin(th))^2 +
    (1/2)*M*p_d^2;
V = m*g*l*cos(th);

L = T - V;

% Equations of motion: d/dt (pdL/ pdq_d) - (pdL/pdq) = Q
%(Q makes up the non-conservative forces acting on the system),
% q is a state-variable,
% pd denotes partial derivative,
% the right-hand side of the above equation will be denoted L_q for state
% variable q; this implies that Q - L_q = 0

% find L_q's given p's first and second derivative is p_d and p_dd
% and theta's derviative is th_d and th_dd
% the lagrange function assumes V contains up to the second derivative of
% each variable
V = [p, p_d, p_dd, th, th_d, th_dd];
L_ = lagrange(L, V);

% for linear position
Q_p = F - c*p_d;
L_p = L_(1);

% for pendulum angle
Q_th = -rho*th_d;
L_th = L_(2);

%% Equations of Motion
% writing the equations of motion, whilst collecting  the variables of
    interest, V
eom1 = collect(simplify(Q_p - L_p), V);
eom2 = collect(simplify(Q_th - L_th), V);

%% State Space Representation
% define state variables and rewrite eom
syms x_1 x_2 x_3 x_4 real
```

```matlab
x = [x_1; x_2; x_3; x_4];
x_T = transpose(x);

% where x_1 = p, x_2 = th, x_3 = p_d, x_4 = th_d

eom_s1 = subs(eom1, [p th p_d th_d], x_T);
eom_s2 = subs(eom2, [p th p_d th_d], x_T);

% contruct statespace imput function x_d = F(x)
% prepare state variables
x_d = solve(eom_s1,eom_s2, p_dd, th_dd);      %solve for p_dd and th_dd

x_1d = x_3;                                    % x_1d = p_d
x_2d = x_4;                                    % x_2d = th_d
x_3d = x_d.p_dd;                               % x_3d = p_dd
x_4d = x_d.th_dd;                              % x_4d = th_dd

% write the state-space representation
F_x = [ x_1d; x_2d; x_3d; x_4d ];
F_x = collect(F_x, x);

% solve the jacobian about an equilibrium position
equ = [0 0 0 0];                   %define an equilibrium position

% rewrite input function in the form
% x_d = Ax + Bu
A = simplify(subs(jacobian(F_x, x), x_T, equ));
B = simplify(subs(jacobian(F_x, F), x_T, equ));

A_ = simplify(subs(A, [M m c l rho g], [10 80 0.1 1 0.01 9.81]));
B_ = simplify(subs(B, [M m c l rho g], [10 80 0.1 1 0.01 9.81]));

EIG = simplify(eig(A_));
```

Script 1: PBA3_1.m

The expression $F(x)$, such that $\dot{x} = F(x)$, including non-linearities is representing by F_x in the Matlab script above; calculated using the same method as above but without small angle approximation. This gives:

$$F(x) = \begin{pmatrix} x_3 \\ x_4 \\ -\frac{\text{rho } x_4 \, \cos(x_2) - F\, l + c\, l\, x_3 + l^2\, m\, x_4^2\, \sin(x_2) - g\, l\, m\, \cos(x_2)\, \sin(x_2)}{l\left(-1\, m\cos(x_2)^2 + M + m\right)} \\ -\frac{M\, \rho\, x_4 + m\, \text{rho } x_4 - F\, l\, m\, \cos(x_2) - g\, l\, m^2\, \sin(x_2) + l^2\, m^2\, x_4^2\, \cos(x_2)\, \sin(x_2) - M\, g\, l\, m\, \sin(x_2) + c\, l\, m\, x_3\, \cos(x_2)}{l^2\, m\left(-1\, m\cos(x_2)^2 + M + m\right)} \end{pmatrix}$$

From Taylor's series expansion, it is known that this function can be approximated about a point; using the Jacobian. This is outlined in the code, where the equilibrium point for this system is $x = [0000]$, as this defines the position where the pendulum is held still in the vertical position, without movement of the cart. The Jacobian is computed with respect to x for the state's matrix A, and with respect to F for the input matrix B. This gives the following state-space equation:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{g\,m}{M} & -\frac{c}{M} & -\frac{\text{rho}}{M\,l} \\ 0 & \frac{g\,(M+m)}{M\,l} & -\frac{c}{M\,l} & -\frac{\text{rho}\,(M+m)}{M\,l^2\,m} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \frac{1}{M} \\ \frac{1}{M\,l} \end{pmatrix} * F$$

It can be seen that this system of equations was the same as that derived from small angle approximation; shown above.

Substituting in parameter values: M = 10 kg, m = 80 kg, c = 0.1N/m/s, l = 1m, rho = 0.01Nms, g = $9.81\text{m}/s^2$

thus the state-space equation becomes:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 1.0 \\ 0 & 78.48 & -0.01 & -0.001 \\ 0 & 88.29 & -0.01 & -0.001125 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0.1 \\ 0.1 \end{pmatrix} * F$$

All of the mathmatics for these steps is completed using Matlab, as outlined in the above snippet of code.

2 Question 2

To construct a Simulink model of the system, the equations of motion had to be written in such a way that $\dot{p}$ can be integrated to p, and $\dot{\theta}$ integrated to give th. This can be done by writing one equation in terms of $\ddot{\theta}$, and the other in terms of $\ddot{p}$, allowing integration to achieve the state variables $(p, th, \dot{p}, \dot{\theta})$. In doing this, the outputs end at the right hand side of the simulink model and the input force at the left.

With the first equation of motion:

$$-\rho \times \dot{\theta} = ml^2\ddot{\theta} - mlcos(\theta)\ddot{p} - mglsin(\theta)$$

$$ml^2 \times \ddot{\theta} = mglsin(\theta) + ml\ddot{p}cos(\theta) - \rho\dot{\theta}$$

$$\therefore \ddot{\theta} = \frac{g}{l}sin(\theta) + \frac{\ddot{p}}{l}cos(\theta) - \frac{\rho}{ml^2} \times \dot{\theta}$$

similarly:

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

substituting in the equation above (solve for $\ddot{\theta}$)

$$= (M + m)\ddot{p} + ml\dot{\theta}^2sin(\theta) - mlcos(\theta)\frac{g}{l}sin(\theta) - mlcos(\theta)\frac{\ddot{p}}{l}cos(\theta) + mlcos(\theta)\frac{\rho}{ml^2} \times \dot{\theta}$$

$$= (M + m - mcos(\theta)^2)\ddot{p} + ml \times \dot{\theta}^2sin(\theta) - mgcos(\theta)sin(\theta) + \frac{\rho}{l}\dot{\theta}cos(\theta)$$

$$\therefore \ddot{p} = \frac{1}{M+m-mcos(\theta)^2}(F - c\dot{p} - ml \times \dot{\theta}^2sin(\theta) + mgcos(\theta)sin(\theta) - \frac{\rho}{l}\dot{\theta}cos(\theta))$$

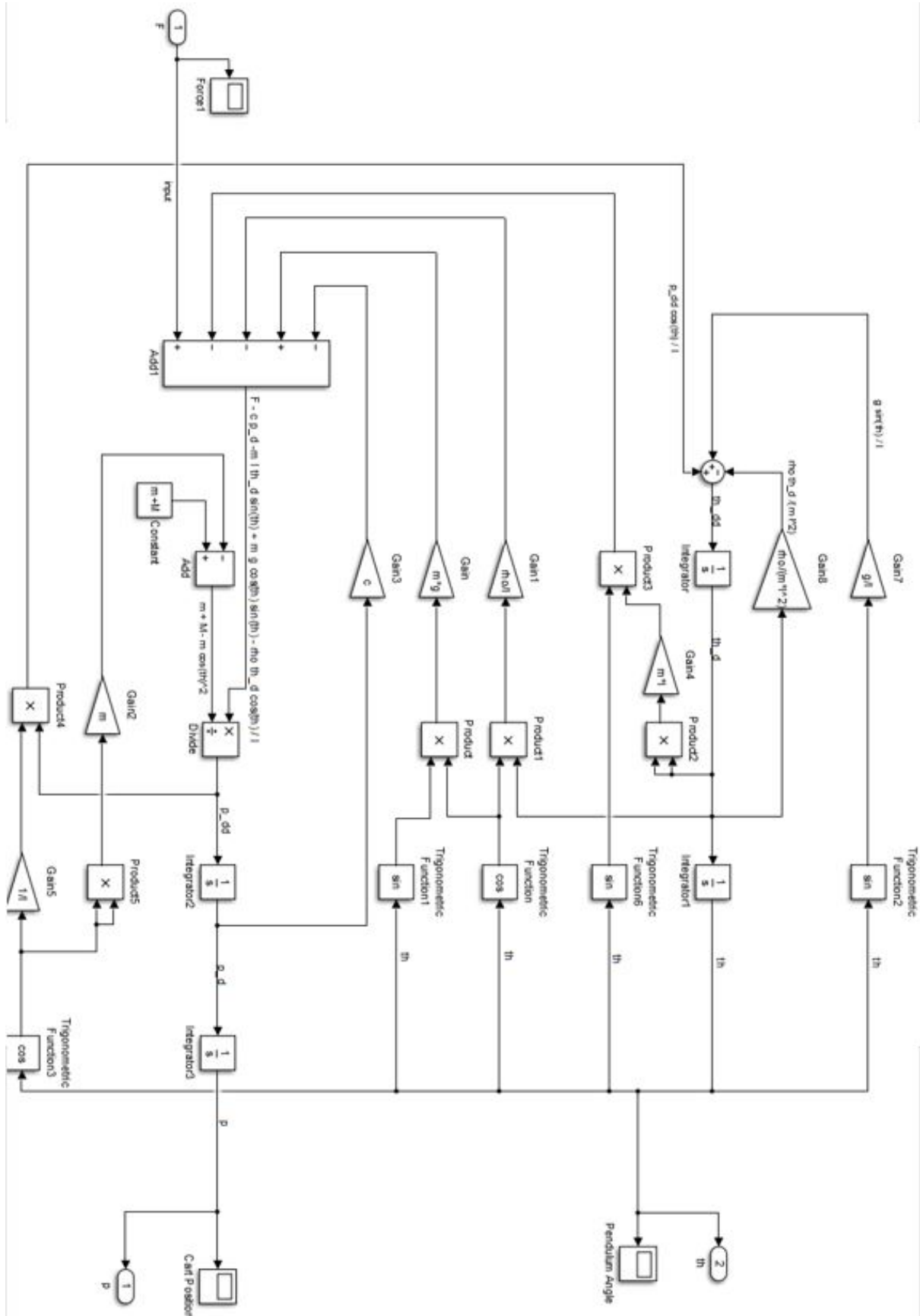From this, the following simulink model was generated.

Figure 2: Simulink Model of Pendulum and Cart System

3   Question 3

The state-space representation of the simulink model made in question two can be computed using Matlab's linmod command; which requires an equilibrium point to be able to linearise the system. The script below displays the code required to do this task, and computes the eigenvalues of A to show that the it is the same model as the analysis in question 1.

```matlab
%close all;
% Define parameters
M = 10;
m = 80;
c = 0.1;
l = 1;
rho = 0.01;
g = 9.81;

%% Question 2, calculate the eigenvalues of the non-linear (simulink) model
sys = linmod('PBA3_2_2', [0, 0, 0, 0], 0);

EIG_sim = eig(sys.a)

% EIG_sim =
%
%         0
%   -9.4013
%    9.3913
%   -0.0011
```

Script 2: PBA3_2_script.m

it can be seen that the equilibrium point was the same as that used in question 1, and the input to the system is 0.

In the script of question 1, the eigenvalues of A (represented by the variable EIG) were computed to be:

$$\lambda = \begin{pmatrix} 0 \\ 9.3913 \\ -9.4013 \\ -0.0011 \end{pmatrix}$$

this is transparent to the simulation as (the variable EIG_sim):

$$\lambda = \begin{pmatrix} 0 \\ -9.4013 \\ 9.3913 \\ -0.0011 \end{pmatrix}$$

and thus they are the same.

Similarly it can be seen that the state-space representation of the simulation is equivalent to that of question 1 (where $A = sys.a$, $B = sys.b$ in the previous code):

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 1.0 \\ 0 & 78.48 & -0.01 & -0.001 \\ 0 & 88.29 & -0.01 & -0.0011 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0.1 \\ 0.1 \end{pmatrix} * F$$

4   Question 4

Given the system is controllable, the poles of the system can be repositioned using state-feedback. The controllability of the system is determined by ensuring the controllability matrix has a determinant not equal to 0; implying the linear independence and invertibility.

The controllability matrix is;

$$W = [(B)(AB)(A^2B)(A^3B)]$$

Computing the determinant of this matrix can be done in Matlab; using the symbolic A and B from part 1. This gives;

$$det(W) = -\frac{g^2}{M^4 l^4}$$

This is not equal to 0 so long as there is gravity, and doesn't exist if there is no cart mass or pendulum length. Thus, state-feedback can be used with the control law

$$u = -Kx$$

state feedback has the form:

$$\dot{x} = (A - KB)x$$

where is a 4x4 matrix which will change the poles of the system to those desired; stabilising the system.

The feedback matrix K can be computed using Matlab's acker function, which takes in the state-space matrices of the system, and a matrix defining the desired pole locations.

To stabilize the system, the poles must be placed in the left hand plane (the general expression is $y = exp(\lambda t)$, thus if λ is negative the response of the system will decay to a steady state). The position of the poles will drastically affect the system's response, and thus must be chosen more wisely than just having 4 random negative poles. The closer the poles to the imaginary axis, the slower the response time, however the further the poles the more susceptible and responsive the system becomes to noise (high frequency). If the poles are positioned too far away from the imaginary axis, the desired response could be too quick for the actuator to perform (saturation). Therefore, it is important to decide poles on application, where in this case the system replicates that of a segway. A segway will be driven at the base, carrying a adult load (pendulum mass) and have a pendulum / handle length of a meter. In this application it would be important to have a fast response time such that hitting a small bum in the road wouldn't create a long

duration, noticeable sway of. Additionally, it would be undesirable to have poles with a low damping ratio (perhaps anything smaller than 0.707) such that oscillation occurs; as the pendulum will sway back and forth with disturbances. In the s-place, this refers to complex conjugate poles with a higher imaginary component than that of it's negative real component.

There are multiple different ways to simulate different disturbances to acknowledge how the chosen poles respond, and whether they need to be rethought; one can change the mass of the pendulum (to replicate the experience for a child instead of a 80kg person), add a disturbance acceleration to mimic a bump on the ground, or similarly add an impulse or step force to the base of the pendulum.

The poles chosen for this section were

$$P = \begin{pmatrix} (-4+2i) & (-4-2i) & (-20+10i) & (-20-10i) \end{pmatrix}$$

this will produce a response with $\zeta = 0.8944$ and a dominant natural frequency of $4.472 rad/sec$; the poles were moved into this position by considering required gain of the system, where one maximum gain (of K) is 19477, which is assumed to be high in itself. It is believe that the second conjugate pair are fair enough from the dominant pair that they will have a minimal effect on the response, without drastically increasing the system's sensitivity to noise.

Using the acker function, as seen in the below code, results in the following K matrix for the aforementioned pole placement.

$$K = \begin{pmatrix} -10193.68 & 19477.19 & -4893.19 & 5373.08 \end{pmatrix}$$

```
%close all;
% Define parameters
M = 10;
m = 80;
c = 0.1;
l = 1;
rho = 0.01;
g = 9.81;

%% Question 2, calculate the eigenvalues of the non−linear (simulink) model
sys = linmod('PBA3_2_2', [0, 0, 0, 0], 0);
%
% EIG_sim = eig(sys.a)
%
%% Generate a K matrix for the selected pole placement
% P = [−2 −2 −8 −8];                    %stable, th_max = 2.8 degrees for
    68.4N force applied for 1sec
%P = [−1+0.5i −1−0.5i −2+1i −2−1i];         %unstable for 68.4N force
    applied for 1sec
P = [−4+2i −4−2i −20+10i −20−10i];          % th_max = 1.6e−3 rad for
    68.4N force applied for 1sec
% % % %P = [−400 −500 −505 −600];
%P = [−1 −1 −2 −2];                          %very unstable
%
K = acker(sys.a, sys.b, P);
```

```
%% Pole Placement Notes

% having large poles −> response time too quick for motors, more
% subseptible to disturabances / high frequency forcing

% having poles too low −> slower response time, larger displacement /
% overshoot, resulting in larger drifting on p due to mass on larger angle
% before compensation

% including complex conjugate poles increases rise time −> decreases
% drift on position p, at the expense of (very) small amplitude oscillation
% / overshoot before settling if 1 > zeta > 0.707.

%% Plot the step response of the controlled system
con_sys = linmod('PBA3_4', [0, 0, 0, 0], 0);
[Num, Den] = ss2tf(con_sys.a, con_sys.b, con_sys.c, con_sys.d);
T_pos = tf(Num(1,:), Den);
T_theta = tf(Num(2,:), Den);
figure(1); hold on; step(T_pos); step(T_theta); hold off;
```

Script 3: PBA3_4_script.m

Using the code towards the bottom of the above script, the step response of the system can be illustrated (however it uses the simulink model made in question 5 of this assignment, but it's response is useful to see how changing the poles can change the response of this system):
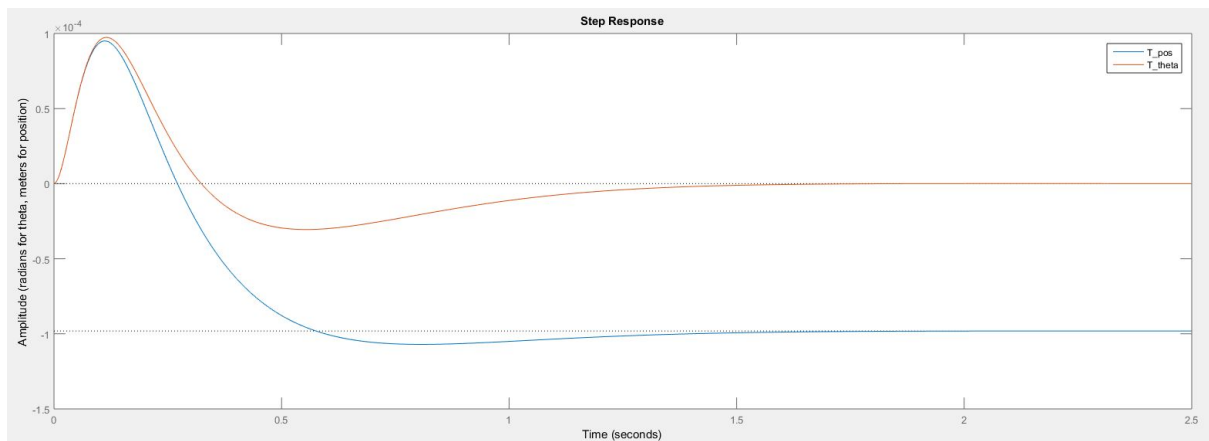

Figure 3: Step response of pendulum angle and cart position

5 Question 5

The model made in question 2 is illustrated below with state feedback; each state is multiplied by it's corresponding gain from the matrix K and fed in as input to the system such that $u = -Kx$. This is illustrated on the following page; a bit cramped to fit the width on the page, but the key difference is the four gain stages at the bottom of the model.
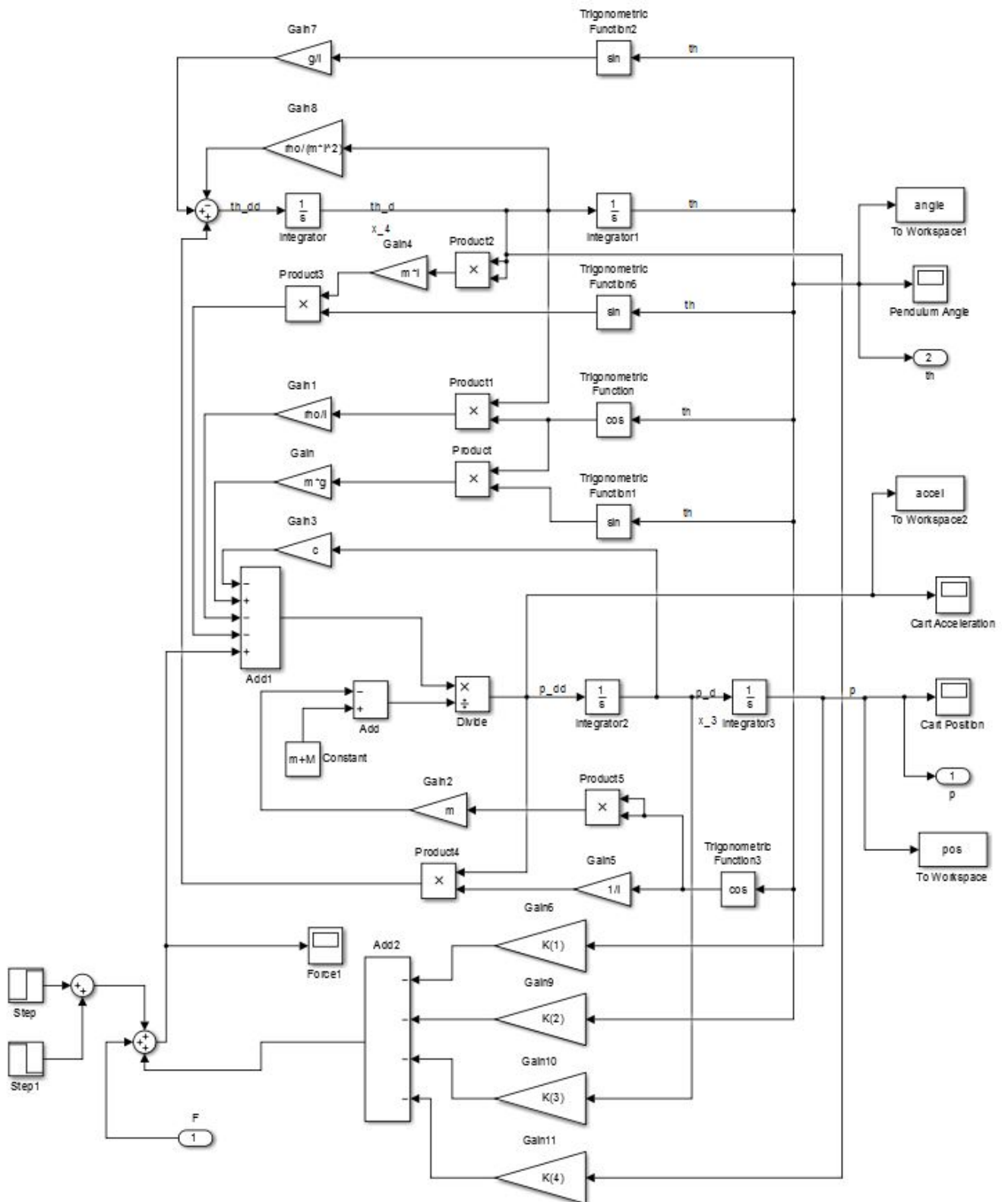
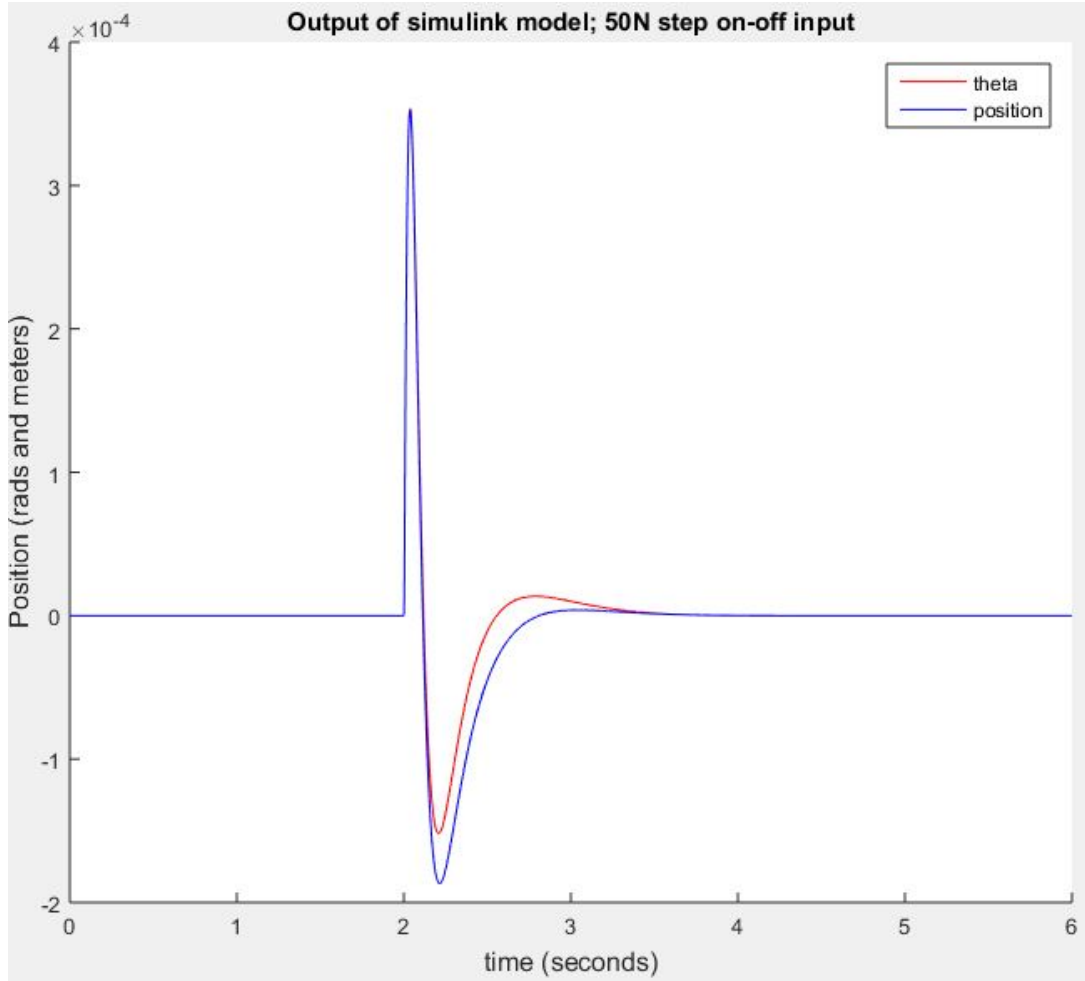Figure 4: Simulink Model of Pendulum and Cart System with state feedback

Figure 5: Response of 50N step on-off force applied to state feedback system at input

It was found that this system is very robust to disturbances(having started at equilibrium), where a 200N impulse and a ramp disturbance on theta with a slope of 2, and changing the length and mass of the pendulum wasn't able to considerably move the pendulum angle, or make unstable; mainly due to the high frequency poles. If however, the initial state of theta was $\pi/2$ it was found that the system became unstable; thus the system is locally asymptotically stable, as a change in initial conditions has rendered the system unstable. In addition to this, if the poles of this system were moved closer to the imaginary axis, they responded slower to system disturbances, thus it was found that poles with real components -1, -1, -2, -2, led to an unstable system when a 70N force was applied at the input for 1 second. These are both illustrated below.
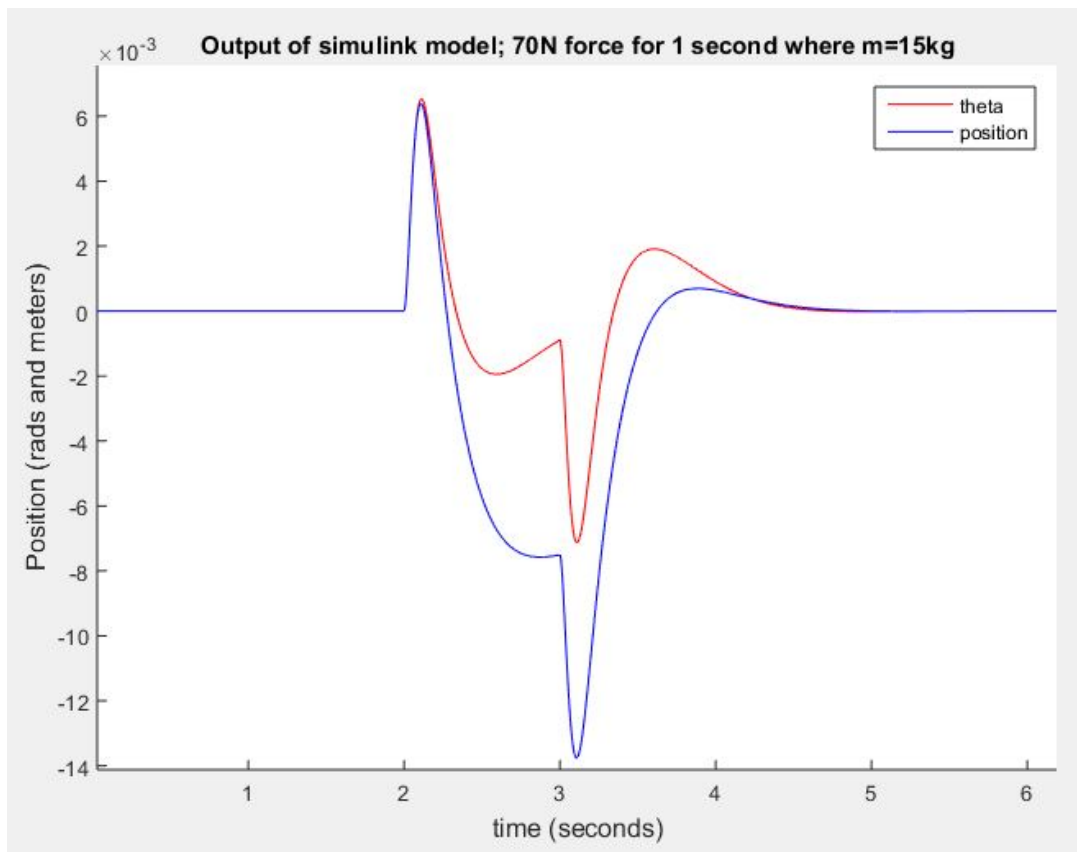
15

Figure 6: Response of 70N step on-off 1 sec force, where m=15kg, it can be seen that a greater amplitude of oscillation occurred, however it is unrecognisable for the rider

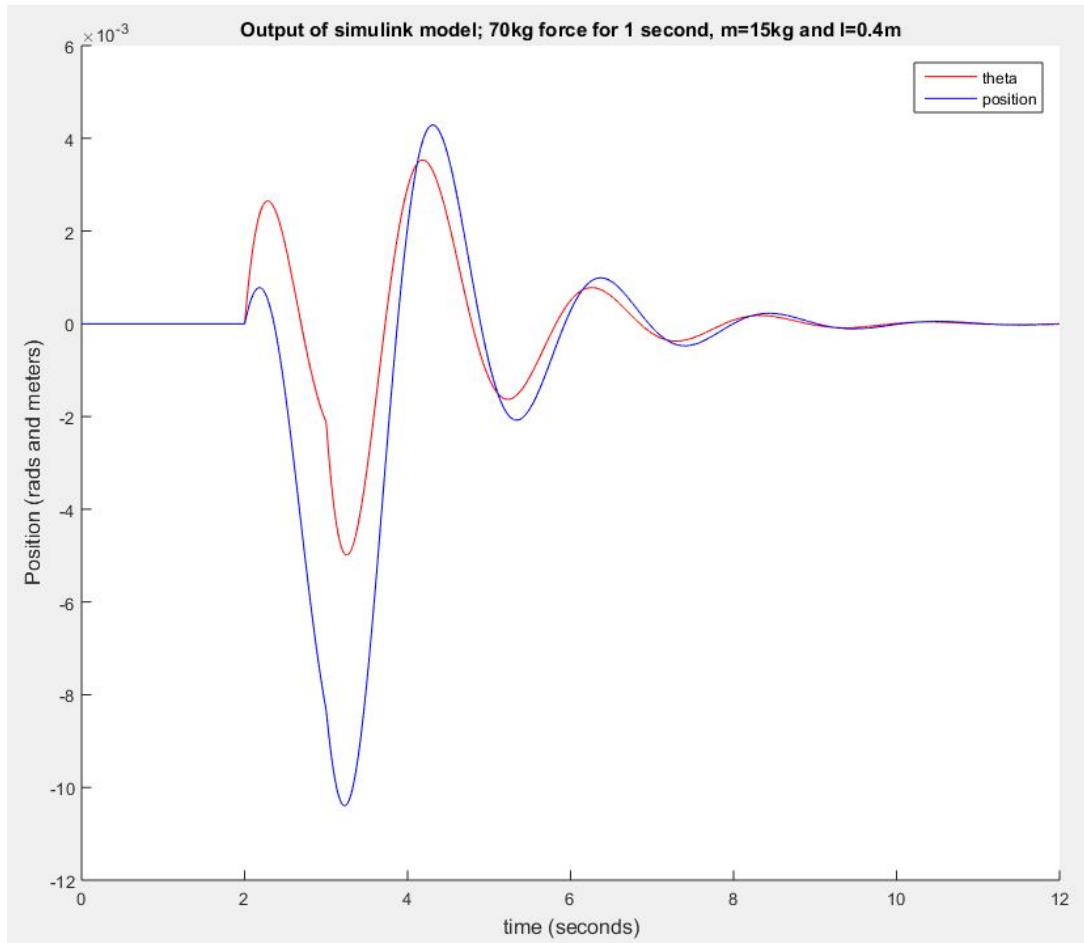Figure 7: Response of 70N step on-off 1 sec force, where m=15kg and l=0.4m, it can be seen that a greater amplitude of and more oscillation occurred, however it is unrecognisable for the rider
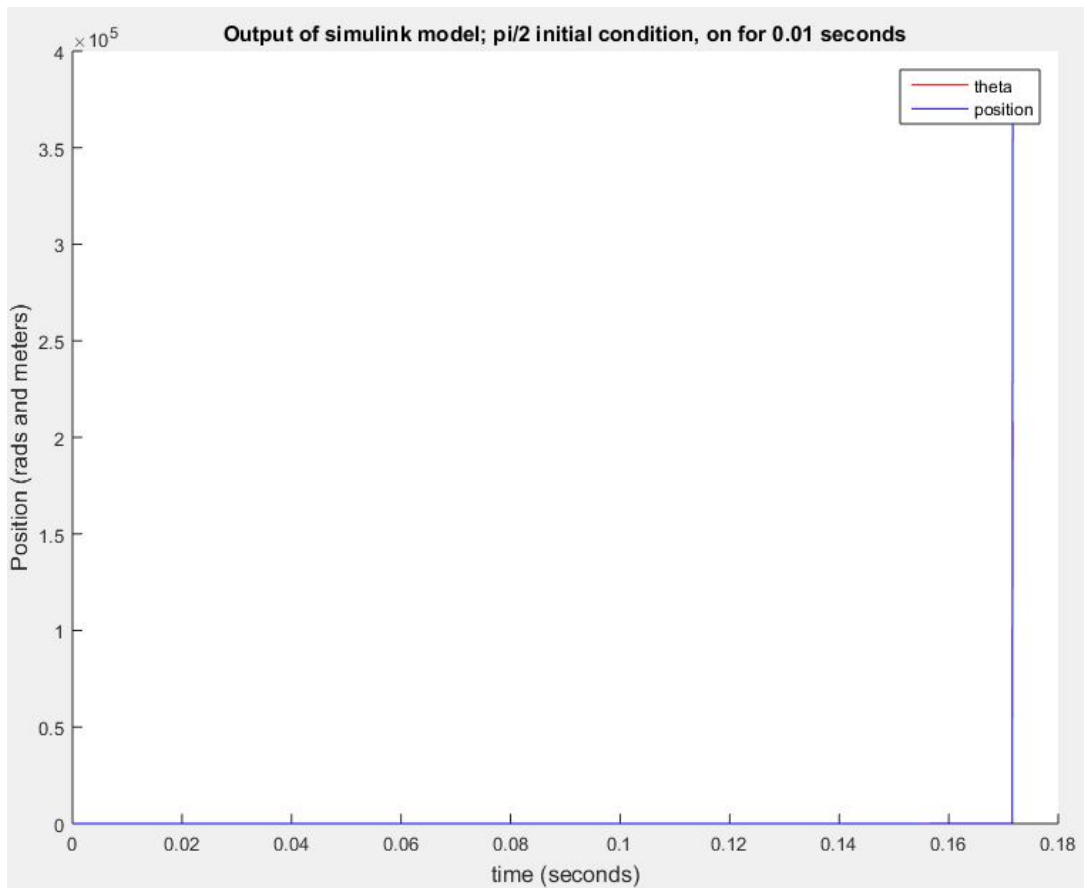
Figure 8: Response of 70N step on-off 1 sec force, with 0.01 second initial step added to theta
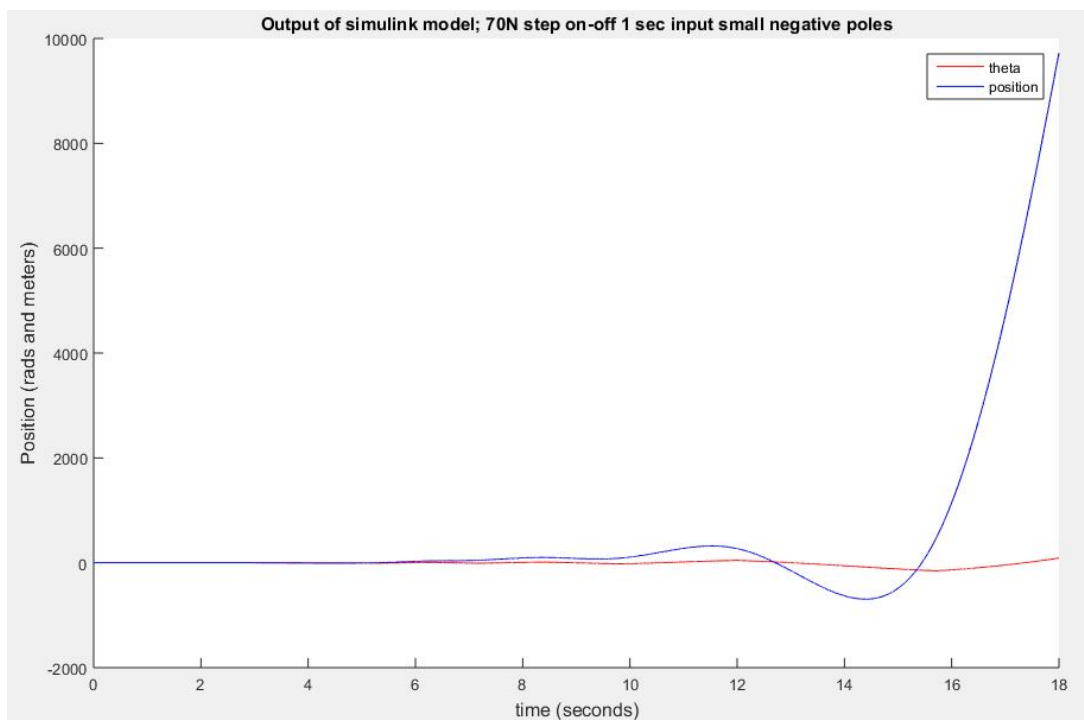

Figure 9: Response of 70N step on-off 1 sec force applied to state feedback system at input for poles at -1, -1, -2, -2