EE498 Special Topics : Control System Design and Simulation Project

**Student Name:** Onurcan YILMAZ

**Student ID:** 2544754

**Student email:** [e254475@metu.edu.tr](mailto:e209318@metu.edu.tr)

İçindekiler

[Part 1 3](#_Toc76849644)

[Introduction 3](#_Toc76849645)

[Model 3](#_Toc76849646)

[Linearized State Space Model 3](#_Toc76849647)

[Control Method Selection 4](#_Toc76849648)

[Model Predictive Control (MPC) 4](#_Toc76849649)

[Part 2 6](#_Toc76849650)

[a) Simulation in Simulink 7](#_Toc76849651)

[b) Runge-Kutta Method 8](#_Toc76849652)

[References 11](#_Toc76849655)

[APPENDICES 12](#_Toc76849656)

[APPENDIX A 13](#_Toc76849657)

# Part 1

## Introduction

For this part of project, a double inverted pendulum on a cart system with a rotary actuator between rods is used. At the first part, system’s equations of motion are obtained using Lagrangian Mechanics. After that, these equations linearized.

At the second part, control methods studied in the lecture are discussed and Model Predictive Controller (MPC) is chosen for implementation.

At the third part, MPC is constructed using linearized system model and validated on it.

In the final part, this controller applied on non-linear system model and results discussed.

## Model

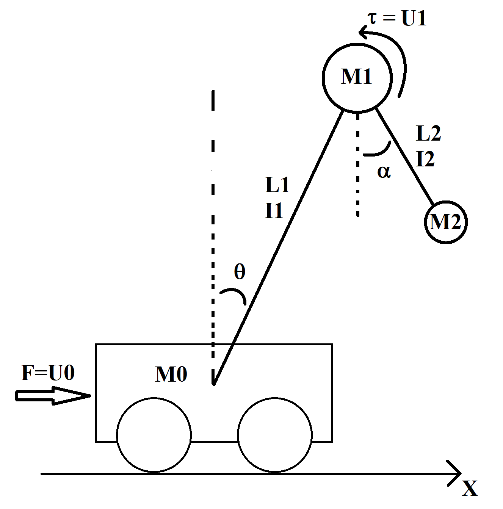


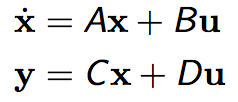
Figure 1. System Diagram

In figure 1, system diagram is shown. For the sake of simplicity, all masses are considered as point masses located in given positions. Also, Internal parameters set to; M0=1kg, M1=0.4kg, M2=0.2kg, L1=0.5m, L2=0.25m, I1=1/12\*M1\*L1^2 and I2=1/12\*M2\*L2^2

Since the resulting non-linear equations too long, they are given in APPENDIX A.

### Linearized State Space Model

Linearized state pace model of the system given below.

(1)

Where:

A = [0, 0, 0, 1, 0, 0]

[0, 0, 0, 0, 1, 0]

[0, 0, 0, 0, 0, 1]

[0, -26487/4900, 2943/12250, 0, 0, 0]

[0, 70632/1225, 35316/1225, 0, 0, 0]

[0, -105948/1225, -553284/6125, 0, 0, 0] (2)

B = [ 0, 0]

[ 0, 0]

[ 0, 0]

[193/196, 97/392]

[ -90/49, -1485/49]

[ -12/49, 4506/49] (3)

C = [1, 0, 0, 0, 0, 0] (4)

D= [0, 0] (5)

## Control Method Selection

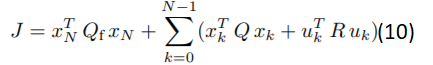
Through the course several control methods are studied however most of them are not applicable for this system. For example:

* Symmetric optimum needs a Transfer function with large time constants and one small time constant.
* Smith predictor is applied to delayed systems.
* Pole placement, PID controllers and LQR is better suited for linear system which has same transfer function for all state values.

Thus MPC is chosen as controller.

## Model Predictive Control (MPC)

For MPC, cost function in Equation 6 is used.

(6)

Horizon N is chosen as 8 and dual mode prediction is used to avoid any instability. As a result,  is equal to P, where P is calculated using Discrete Time Lyapunov Equation in Equation 7 with discritized A and B matrices at 0.01s sampling time.

(7)

Aim of the system is to move from a point to another without falling. So, cost variables chosen as

(8)

R = [1, 0]

[0, 1] (9)

X0 = [1; 0; 0; 0; 0; 0] (10)

Application of designed MPC controller on linearized system can be seen in figure 2 and 3.



Figure 2. System inputs and output for linearized system



Figure 3. Linearized system states

## Simulation of Nonlinear Model

Codes of The Function Blocks derived from previous codes.

## 

Figure 4. Simulink Block Diagram

# Part 2

## Simulation in Simulink

Trajectory of Apollo spacecraft realized in Simulink for T=10. Contents of the function block can be seen in Appendix C.

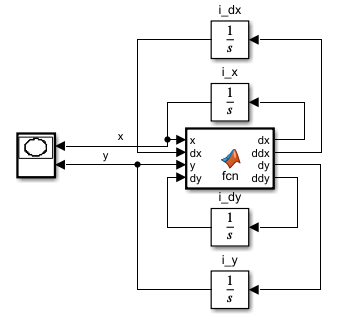


Figure 5. Simulink model of the system.

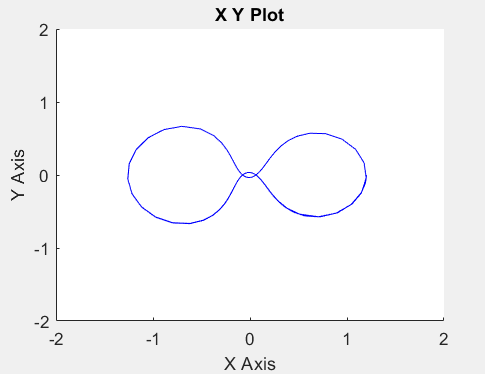


Figure 6. Simulation result of the system

## b) Runge-Kutta Method

a) 4’th order method with constant step size

For this part, 4’th order Runge-Kutta Method with the Butcher Tableau given in Figure 7 is used.

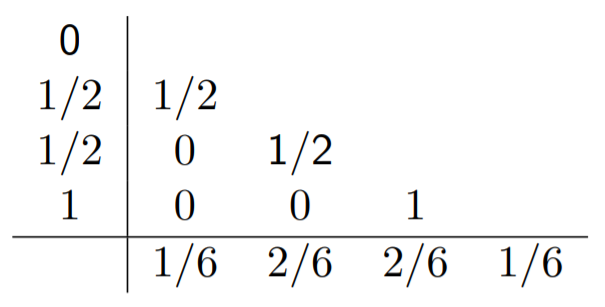


Figure 7. Butcher Tableau of 4’t order Runge-Kutta Method

# 

Figure 8. Trajectory of Apollo spacecraft simulated with for 5\*e-7 step size for T=10. Took 113.3s to compute.

# 

Figure 9. Trajectory of Apollo spacecraft simulated with for 5\*e-6s step size for T=10s. Took 10.7s to compute

1. For variable step size a Heun’s 3’rd order Method with Butcher Tableau given in Figure 10 is used.

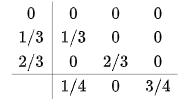


Figure 10. Butcher Tableau of Heun’s 3’rd order Method.

1. For variable step size, previous 3’rd and 4’th order methods are used. Since, simulation is unstable for 5e-5 relative tolerance, 5e-8 relative tolerance is used since it has similar output to Simulink result and can be compared. It took 0.035438s. The MATLAB code can be found in Appendix D.



Figure 11. Simulation result with variable step size using 3’rd and 4’th order Runge-Kutta Methods.

1. As we can see from the figure below, step size is inversely correlated with magnitude of the derivative terms. If the derivative terms are close to 0, it increases and if their magnitude is large, it decreases. Even in some point, one of the derivative terms becomes 0 while the other is large. At these points, h becomes discontinuous.



Figure 12. Step Sizes and State Magnitudes

# References

[1] Crowe-Wright, I., 2018. *Control Theory: The Double Pendulum Inverted on a Cart*. [online] Digitalrepository.unm.edu. Available at: <https://digitalrepository.unm.edu/cgi/viewcontent.cgi?article=1131&context=math\_etds> [Accessed 10 July 2021].

[2] En.wikipedia.org. 2021. *List of Runge–Kutta methods - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/List\_of\_Runge%E2%80%93Kutta\_methods> [Accessed 10 July 2021].

[3] van Biezen, M., 2021. *Physics - Adv. Mechanics: Lagrangian Mech. (1 of 25) What is Lagrangian Mechanics?*. [online] Youtube.com. Available at: <https://www.youtube.com/watch?v=4uJaKJASKnY> [Accessed 10 July 2021].

[4] Schmidt, K., 2021. *EE498 Lecture Notes*.

# APPENDICES

## APPENDIX A

syms theta0(t) theta1(t) theta2(t) m0 m1 m2 L1 L2 I1 I2 g ...

%Define the symbolic variables.

dtheta0 = diff(theta0,t); %Linear velocity of cart

dtheta1 = diff(theta1,t); %Angular velocity of first rod

dtheta2 = diff(theta2,t); %Angular velocity of second rod

g = 9.81; %Gravity Constant

L1 = 0.5; %Lenght of first rod

L2 = 0.25; %Lenght of first rod

m0 = 1; %Mass of cart

m1 = 0.2; %Mass of first rod

m2 = 0.4; %Mass of second rod

I1 = 1/12\*m1\*L1^2; %Inertia of first rod

I2 = 1/12\*m2\*L2^2; %Inertia of second rod

%% Calculating Equations of Motion using Lagrangian Mechanics

%Kinetic Energy

K0 = 1/2\*m0\*dtheta0^2;

K1 = 1/2\*m1\*dtheta0^2 + 1/2\*(m1\*L1^2+I1)\*dtheta1^2 + m1\*L1\*dtheta0\*dtheta1\*cos(theta1);

K2 = 1/2\*m2\*dtheta0^2 + 1/2\*m2\*L1^2\*dtheta1^2 + 1/2\*(m2\*L2^2+I2)\*dtheta2^2 + ...

m2\*L1\*dtheta0\*dtheta1\*cos(theta1) + m2\*L2\*dtheta0\*dtheta2\*cos(theta2) + ...

m2\*L1\*L2\*dtheta1\*dtheta2\*cos(theta1+theta2);

%Potential Energy

V1 = g\*m1\*L1\*cos(theta1);

V2 = g\*m2\*(L1\*cos(theta1) - L2\*cos(theta2));

%Lagrangian Equation

L = simplify(expand(K0+K1+K2 - V1-V2));

%Take Lagrangian derivatives for all 3 variables

ftheta0 = simplify(expand(diff(diff(L,dtheta0),t) - diff(L,theta0)));

ftheta1 = simplify(expand(diff(diff(L,dtheta1),t) - diff(L,theta1)));

ftheta2 = simplify(expand(diff(diff(L,dtheta2),t) - diff(L,theta2)));

%% Calculate Non-Linear A matrice (Only for display purposes)

syms ddtheta0 ddtheta1 ddtheta2

%Change diff(theta0(t),t,t) variables with ddtheta0 or corresponding variables

Ftheta0 = subs(subs(subs(ftheta0,diff(theta0(t), t, t),ddtheta0),diff(theta1(t), t, t),ddtheta1),diff(theta2(t), t, t),ddtheta2);

Ftheta1 = subs(subs(subs(ftheta1,diff(theta0(t), t, t),ddtheta0),diff(theta1(t), t, t),ddtheta1),diff(theta2(t), t, t),ddtheta2);

Ftheta2 = subs(subs(subs(ftheta2,diff(theta0(t), t, t),ddtheta0),diff(theta1(t), t, t),ddtheta1),diff(theta2(t), t, t),ddtheta2);

Ftheta0 = subs(subs(subs(Ftheta0,diff(theta0(t), t),dtheta0),diff(theta1(t), t),dtheta1),diff(theta2(t), t),dtheta2);

Ftheta1 = subs(subs(subs(Ftheta1,diff(theta0(t), t),dtheta0),diff(theta1(t), t),dtheta1),diff(theta2(t), t),dtheta2);

Ftheta2 = subs(subs(subs(Ftheta2,diff(theta0(t), t),dtheta0),diff(theta1(t), t),dtheta1),diff(theta2(t), t),dtheta2);

%Get solutions of ddtheta0, ddtheta1 and ddtheta2 in terms of position and velocity

eqns = [Ftheta0, Ftheta1, Ftheta2];

vars = [ddtheta0, ddtheta1, ddtheta2];

S = solve(eqns,vars);

soltheta0 = simplify(S.ddtheta0);

soltheta1 = simplify(S.ddtheta1);

soltheta2 = simplify(S.ddtheta2);

%% Calculate Linearized System Model

% This equations taken from the master thesis (Control Theory: The Doubl Pendulum Inverted on a Cart)

% of Ian J P Crowe-Wright from University of New Mexico

% https://digitalrepository.unm.edu/cgi/viewcontent.cgi?article=1131&context=math\_etds

% Seperate function as D(?)dd? + C(?, d?)d? + G(?) = Hu

D = [subs(Ftheta0,{ddtheta0 ddtheta1 ddtheta2 dtheta0 dtheta1 dtheta2},{1 0 0 0 0 0}) subs(Ftheta0,{ddtheta0 ddtheta1 ddtheta2 dtheta0 dtheta1 dtheta2},{0 1 0 0 0 0}) subs(Ftheta0,{ddtheta0 ddtheta1 ddtheta2 dtheta0 dtheta1 dtheta2},{0 0 1 0 0 0});

subs(Ftheta1,{ddtheta0 ddtheta1 ddtheta2 dtheta0 dtheta1 dtheta2},{1 0 0 0 0 0}) subs(Ftheta1,{ddtheta0 ddtheta1 ddtheta2 dtheta0 dtheta1 dtheta2},{0 1 0 0 0 0}) subs(Ftheta1,{ddtheta0 ddtheta1 ddtheta2 dtheta0 dtheta1 dtheta2},{0 0 1 0 0 0});

subs(Ftheta2,{ddtheta0 ddtheta1 ddtheta2 dtheta0 dtheta1 dtheta2},{1 0 0 0 0 0}) subs(Ftheta2,{ddtheta0 ddtheta1 ddtheta2 dtheta0 dtheta1 dtheta2},{0 1 0 0 0 0}) subs(Ftheta2,{ddtheta0 ddtheta1 ddtheta2 dtheta0 dtheta1 dtheta2},{0 0 1 0 0 0})];

% C becomes 0 after linearization so it is not calculated

% C = [0 -(m1+m2)\*L1\*sin(theta1)\*dtheta1 -m2\*L2\*sin(theta2)\*dtheta2;

% 0 0 m2\*L1\*L2\*sin(theta1-theta2)\*dtheta2;

% 0 -m2\*L1\*L2\*sin(theta1-theta2)\*dtheta1 0 ];

G = [subs(Ftheta0,{dtheta0 dtheta1 dtheta2 ddtheta0 ddtheta1 ddtheta2},{0 0 0 0 0 0});

subs(Ftheta1,{dtheta0 dtheta1 dtheta2 ddtheta0 ddtheta1 ddtheta2},{0 0 0 0 0 0});

subs(Ftheta2,{dtheta0 dtheta1 dtheta2 ddtheta0 ddtheta1 ddtheta2},{0 0 0 0 0 0})];

% Firs input is the force on the cart.

% Second input is the torque generated by servo between two rods

H = [1 L1;

0 0;

0 1];

LD = subs(D,{theta0, theta1,theta2},{0,0,0});

LG = subs([diff(G,theta0) diff(G,theta1) diff(G,theta2)],{theta0, theta1,theta2},{0 0 0});

%[dtheta0] = [theta0]

%[dtheta1] = [theta1]

%[dtheta2] = A\*[theta2] + B\*[u1(t)]

%[ddtheta0] = [dtheta0] [u2(t)]

%[ddtheta1] = [dtheta1]

%[ddtheta2] = [dtheta2]

%Linearized A matrice

LA = [zeros(3,3) eye(3);

-LD\LG zeros(3,3)];

LA = simplify(LA);

%Linearized B matrice

LB = [zeros(3,2);

LD\H];

LB = simplify(LB);

## APPENDIX B

clc, clear all, close all;

%% Vehicle Model

fi = 0.03;

tao = 0.05;

Ts = 0.01;

tend = 250;

N = 8;

% x = [x; theta; alpha; dx; dtheta; dalpha]

LAc = [0, 0, 0, 1, 0, 0;

0, 0, 0, 0, 1, 0;

0, 0, 0, 0, 0, 1;

0, -26487/4900, 2943/12250, 0, 0, 0;

0, 70632/1225, 35316/1225, 0, 0, 0;

0, -105948/1225, -553284/6125, 0, 0, 0];

LBc = [ 0, 0;

0, 0;

0, 0;

193/196, 97/392;

-90/49, -1485/49;

-12/49, 4506/49];

Cc = [1 0 0 0 0 0];

Dc = 0;

sys = ss(LAc,LBc,Cc,Dc);

sys\_disc = c2d(sys,Ts);

[A,B,C,D] = ssdata(sys\_disc);

x0 = [1;0;0;0;0;0];

% Optimal control solution for N = 8

G = [zeros(6,2) zeros(6,2) zeros(6,2) zeros(6,2) zeros(6,2) zeros(6,2) zeros(6,2) zeros(6,2);

B zeros(6,2) zeros(6,2) zeros(6,2) zeros(6,2) zeros(6,2) zeros(6,2) zeros(6,2);

A\*B B zeros(6,2) zeros(6,2) zeros(6,2) zeros(6,2) zeros(6,2) zeros(6,2);

A^2\*B A\*B B zeros(6,2) zeros(6,2) zeros(6,2) zeros(6,2) zeros(6,2);

A^3\*B A^2\*B A\*B B zeros(6,2) zeros(6,2) zeros(6,2) zeros(6,2);

A^4\*B A^3\*B A^2\*B A\*B B zeros(6,2) zeros(6,2) zeros(6,2);

A^5\*B A^4\*B A^3\*B A^2\*B A\*B B zeros(6,2) zeros(6,2);

A^6\*B A^5\*B A^4\*B A^3\*B A^2\*B A\*B B zeros(6,2);

A^7\*B A^6\*B A^5\*B A^4\*B A^3\*B A^2\*B A\*B B];

H = [eye(6); A; A^2; A^3; A^4; A^5; A^6; A^7; A^8];

Q = C'\*C;

R = 0.01\*eye(2);

%Q = eye(3);

Pinf = idare(A,B,Q,R,zeros(6,2),eye(6) );

Kinf = inv(R+B'\*Pinf\*B)\*B'\*Pinf\*A;

% A\*X\*A' - X + Q = 0; X = dlyap(A,Q)

P = dlyap( (A-B\*Kinf)',Q+Kinf'\*R\*Kinf);

Qf = P;

Qbar = blkdiag(Q,Q,Q,Q,Q,Q,Q,Q,Qf);

Rbar = blkdiag(R,R,R,R,R,R,R,R);

M = G'\*Qbar\*G + Rbar;

% input bound: umin <= u <= umax

umin = -Inf;

umax = Inf;

%umin = -3;

%umax = 3;

lb = [umin;umin;umin;umin];

ub = [umax;umax;umax;umax];

% Apply MPC steps

xVec(:,1) = x0;

yVec(1) = C\*x0;

uVec = [0;0];

for kk = 1:250

alpha = G'\*Qbar'\*H\*xVec(:,kk);

Usol = quadprog(M,alpha',[],[],[],[],lb,ub);

uVec(:,kk) = [Usol(1);Usol(2)];

xVec(:,kk+1) = A\*xVec(:,kk) + B\*uVec(:,kk);

yVec(kk+1) = C\*xVec(:,kk+1);

Xsol(:,1) = xVec(:,kk);

Xsol(:,2) = A\*Xsol(:,1) + B\*[Usol(1);Usol(2)];

Xsol(:,3) = A\*Xsol(:,2) + B\*[Usol(1);Usol(2)];

Xsol(:,4) = A\*Xsol(:,3) + B\*[Usol(1);Usol(2)];

Ysol(1) = C\*Xsol(:,1);

Ysol(2) = C\*Xsol(:,2);

Ysol(3) = C\*Xsol(:,3);

Ysol(4) = C\*Xsol(:,4);

end

uVec = [uVec uVec(:,end)];

tVec = [0:Ts:250\*Ts];

% figure;

figure, subplot(3,1,1)

stairs(tVec,uVec(1,:),'LineWidth',2);

hold all;

xlabel('time [sec]')

grid

ylabel('u0')

title('Input u0')

subplot(3,1,2)

stairs(tVec,uVec(2,:),'LineWidth',2)

hold all;

grid

xlabel('time [sec]')

ylabel('u1')

title('Input u1')

subplot(3,1,3)

stairs(tVec,C\*xVec,'LineWidth',2)

hold all;

grid

xlabel('time [sec]')

ylabel('y')

title('Output y')

figure, subplot(3,1,1)

stairs(tVec,[1 0 0 0 0 0]\*xVec,'LineWidth',2)

hold all;

grid

xlabel('time [sec]')

ylabel('x')

title('State x')

subplot(3,1,2)

stairs(tVec,[0 1 0 0 0 0]\*xVec,'LineWidth',2)

hold all;

grid

xlabel('time [sec]')

ylabel('theta')

title('State theta')

subplot(3,1,3)

stairs(tVec,[0 0 1 0 0 0]\*xVec,'LineWidth',2)

hold all;

grid

xlabel('time [sec]')

ylabel('alpha')

title('State alpha')

set(findall(gcf,'Type','line'),'LineWidth',2)

set(findall(gcf,'-property','FontSize'),'FontSize',14);

% legend('$u\_{max} = 1.5$','$u\_{max} = 2.5$','$u\_{max} = 4$')

## APPENDIX C

function [dx, ddx, dy, ddy] = fcn(x, dx, y, dy)

m = 1/82.45;

ms = 1 - m;

ddx = 2\*dy + x - ms\*(x+m)/(sqrt((x+m)^2+(y)^2)^3) - m\*(x-ms)/(sqrt((x-ms)^2+(y)^2)^3);

ddy = -2\*dx + y - ms\*(y)/(sqrt((x+m)^2+(y)^2)^3) - m\*(y)/(sqrt((x-ms)^2+(y)^2)^3);

## APPENDIX D

%initial satate & order of simulation

p = 4;

x0 = [1.2;0;0;-1.0494];

%calculate h

rtol = 1e-8;

fac0 = 0.2;

fac1 = 5;

B = 0.5;

calc\_sigma = @(r4,r3,rtol,p) (sqrt(sum((abs(r4-r3)./(abs(r4)\*rtol)).^2)/length(r4)))^(-1/p);

h\_next = @(h,fac0,fac1,B,sigma) (h\*min(fac1,max(fac0,B\*sigma)));

% calculate h0

Fx1 = x0(2);

Fx2 = @(x1,x3,x4) (2\*x4 + x1 - 81.45/82.45\*(x1+1/82.45)/sqrt((x1+1/82.45)^2+x3^2)^3 - 1/82.45\*(x1-81.45/82.45)/sqrt((x1-81.45/82.45)^2+x3^2)^3);

Fx3 = x0(4);

Fx4 = @(x1,x2,x3) -2\*x2 + x3 - 81.45/82.45\*x3/sqrt((x1+1/82.45)^2+x3^2)^3 - 1/82.45\*x3/sqrt((x1-81.45/82.45)^2+x3^2)^3;

h0 = rtol^(1/p) / (sqrt(sum(RK1(x0).^2))/length(x0));

tic

MAX\_ITER = 2000000;

T\_final = 10;

x = [x0 zeros(4,MAX\_ITER)];

h = [h0 zeros(1,MAX\_ITER)];

T = 0; %time passed

toc

for i = 1:(MAX\_ITER)

x(:,i+1) = RK4(x(:,i)',h(i))';

T = T + h(i);

if T >= T\_final

break

end

eta = RK3(x(:,i)',h(i))';

sigma = calc\_sigma(x(:,i+1),eta,rtol,p);

h(i+1) = h\_next(h(i),fac0,fac1,B,sigma);

if T+h(i+1) > T\_final

h(i+1) = T\_final-T;

end

end

figure(1)

plot(x(1,1:i),x(3,1:i))

figure(2)

subplot(2,1,1)

plot(h(1,5:i))

xlabel('step')

ylabel('step size')

title('Step size')

subplot(2,1,2)

plot(x(1,5:i))

xlabel('step')

title('SState Magnitudes')

hold on

plot(x(2,5:i))

plot(x(3,5:i))

plot(x(4,5:i))

legend("x","dx","y","dy")