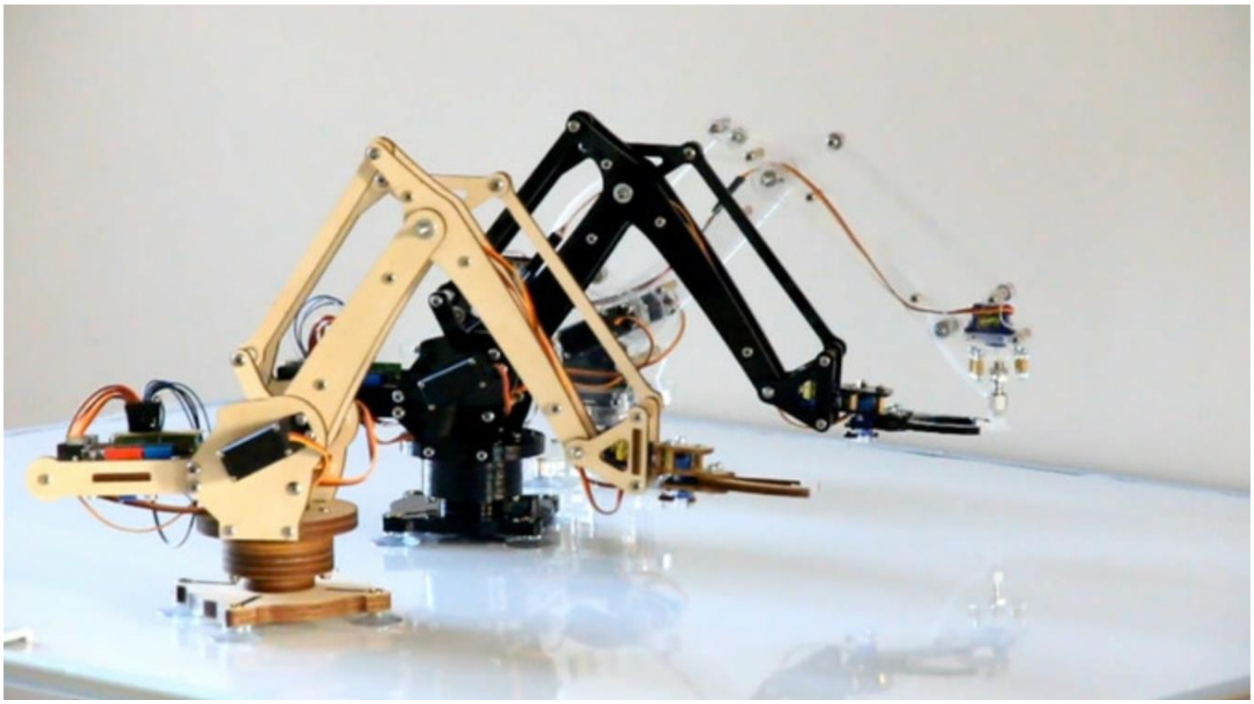
Advance dynamics final project

3-DOFs Industrial Robot



Presented to Assoc. Prof. Thitima Jintanawan

By

Saeth Wannasuphoprasit 5930504121

Thitiwat Hannarong 5930141021

Preface

This report is a part of 2103496 ADV TOP MECH 2.It’s objective is to apply knowledge that we learnt from this subject to analyze the 3-DOFs robot including rotation at joint 1 (along x axis), rotation at joint 2 (along x axis) and rotation at joint 3 (along z axis) which is commonly used in industries. Note that the reference coordinate frame is xyz which is attached to the robot and it also rotates with the robot as well. The global coordinate is XYZ which is fixed to the ground without moving with the robot. You can see details in an Appendix section. Analysis of this robot includes forward kinematics relationship, inverse kinematics relationship and equations of motion. Examples of principles that are mostly used in this project are Dynamics of a Rigid Body, Multi-Body Mechanical System and Lagrange Mechanics. In addition, Matlab was used in order to calculate equations that relate to the robot’s analysis and to plot parameter’s graph that we want to find it’s relation.

Contents

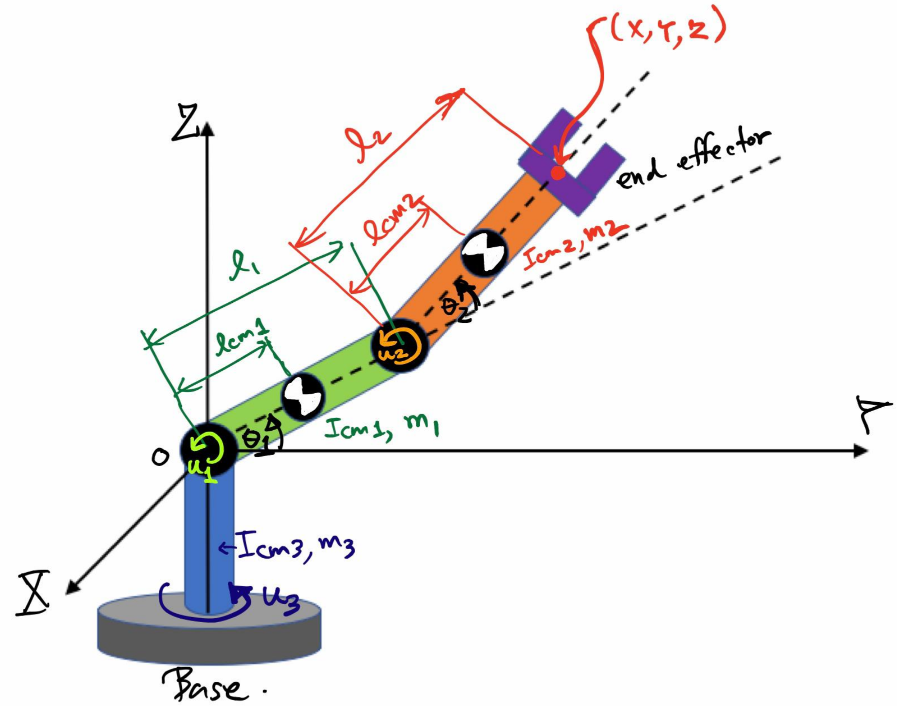
Forward kinematics relationship's analysis

According to a coordinate system and robot's parameters (see Appendix), we can derive an end effector's location (X,Y,Z) in term of θ_1, θ_2, θ_3 and robot's parameters as follows

$X = -[l_1\cos(\theta_1) + l_2\cos(\theta_1+\theta_2)]\sin(\theta_3)$ ----(1) ***

$Y = [l_1\cos(\theta_1) + l_2\cos(\theta_1+\theta_2)]\cos(\theta_3)$ ----(2) ***

$Z = l_1\sin(\theta_1) + l_2\sin(\theta_1+\theta_2)$ ----(3) ***



A coordinating system and a robot's diagram

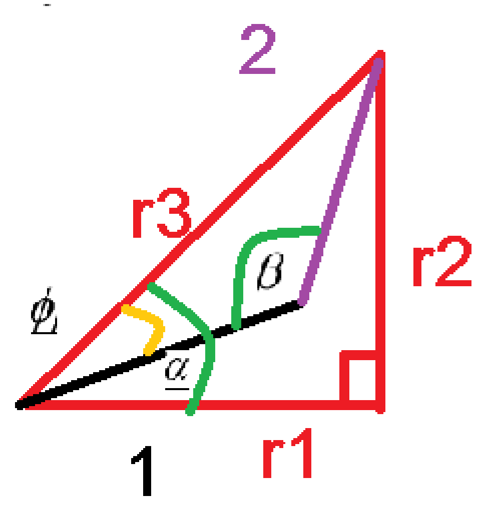
Inverse kinematics relationship's analysis

According to Forward kinematics relationship's analysis, we can recalculate to find θ_1, θ_2 and θ_3 in term of X, Y, Z and robot's parameters as follows

(1)/(2)

$-\frac{X}{Y} = \tan(\theta_3)$ ----(4)

So $\theta_3 = \tan^{-1}\left(\frac{-X}{Y}\right)$ ***



Link 1 and link 2

From link 1(Black) and link 2(Purple) : $\theta_1 = \alpha - \phi$ ----(5), $\theta_2 = 180^\circ - \beta$ ----(6)

, $r_3 = \sqrt{r_1^2 + r_2^2}$, $r_1 = \sqrt{X^2 + Y^2}$, $r_2 = Z$, $\alpha = \tan^{-1}\left(\frac{r_2}{r_1}\right)$, $l_2^2 = l_1^2 + r_3^2 - 2l_1 r_3 \cos(\phi)$,

$\phi = \cos^{-1}\left(\frac{l_2^2 - l_1^2 - r_3^2}{-2l_1 r_3}\right)$, $r_3^2 = l_1^2 + l_2^2 - 2l_1 l_2 \cos(\beta)$, $\beta = \cos^{-1}\left(\frac{r_3^2 - l_1^2 - l_2^2}{-2l_1 l_2}\right)$

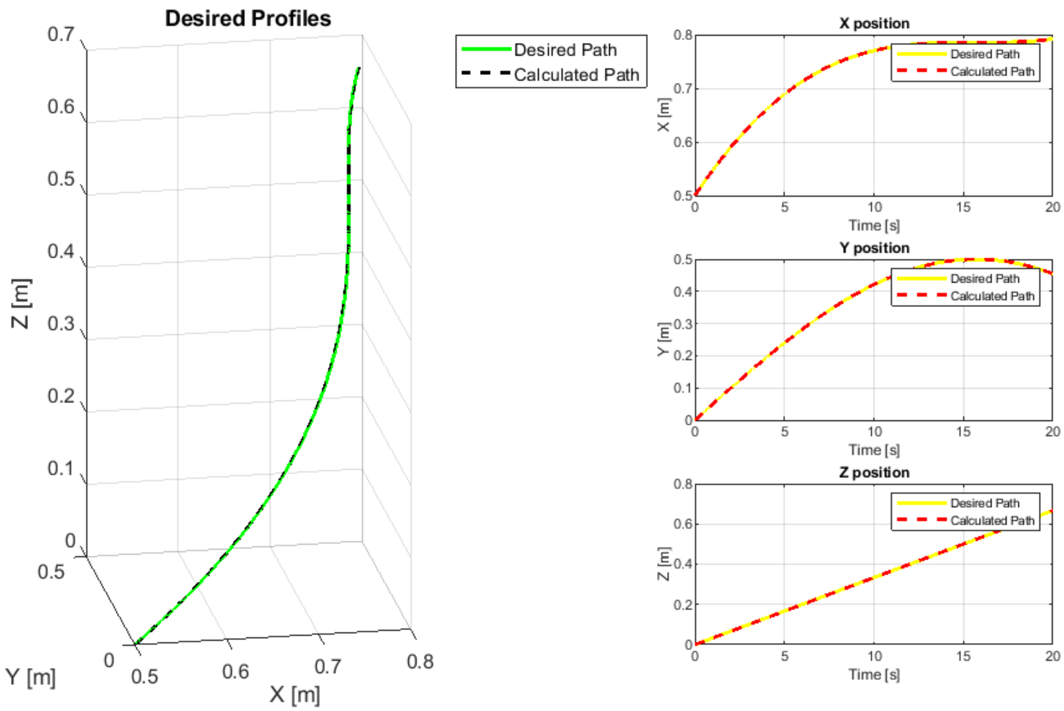
From (5) and (6)

$$\theta_1 = \tan^{-1}\left(\frac{Z}{\sqrt{X^2+Y^2}}\right) + \cos^{-1}\left(\frac{l_2^2 - l_1^2 - (\mathrm{X}^2+\mathrm{Y}^2+\mathrm{Z}^2)}{-2l_1\sqrt{X^2+Y^2+Z^2}}\right) \text{----(7)} \; ***$$

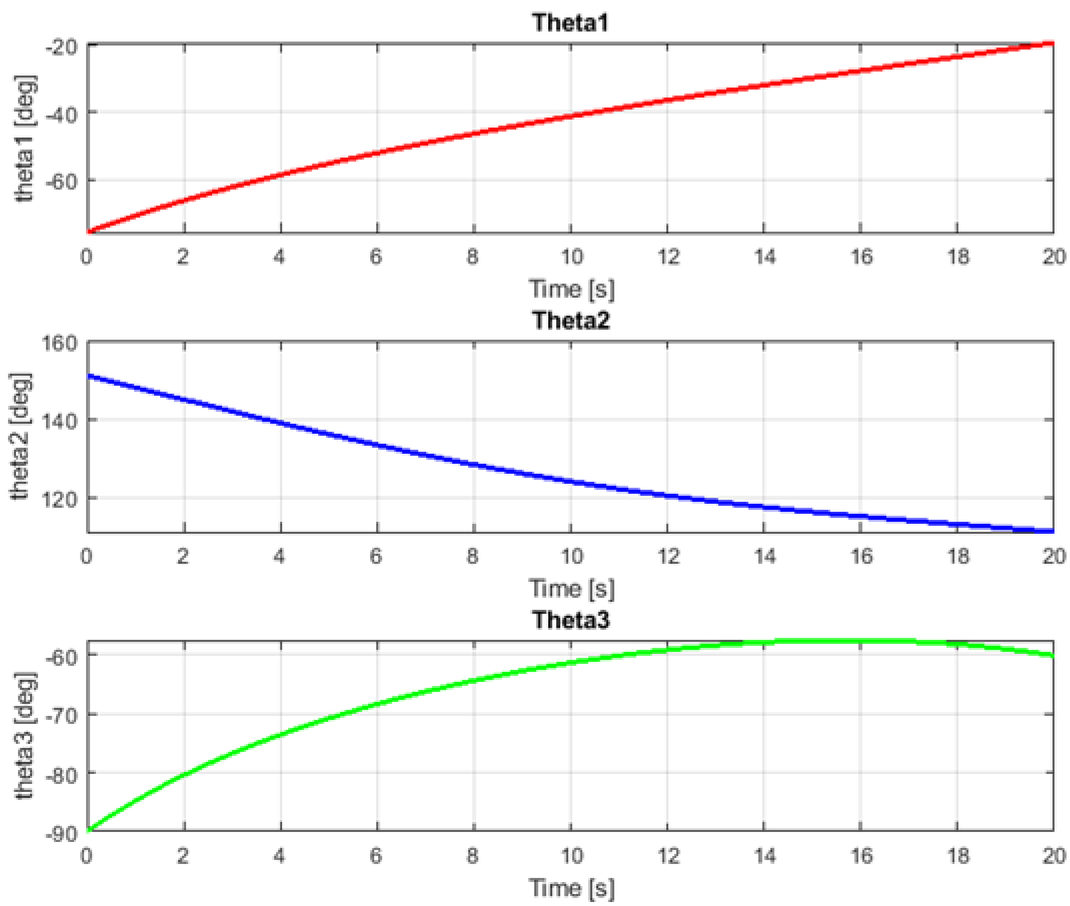
$$\theta_2 = 180^\circ - \cos^{-1}\left(\frac{(\mathrm{X}^2+\mathrm{Y}^2+\mathrm{Z}^2) - l_1^2 - l_2^2}{-2l_1l_2}\right) \text{----(8)} \; ***$$

Matlab: calculating kinematics relationship

First we define desired path (X,Y,Z). Then use inverse kinematics to calculate theta 1 ,2 and 3. After that, we use forward kinematics to recalculate (X,Y,Z) with theta 1,2,3 from the inverse kinematics. Desired path and calculated path are almost the same so in this case, the forward and inverse kinematics are acceptable.

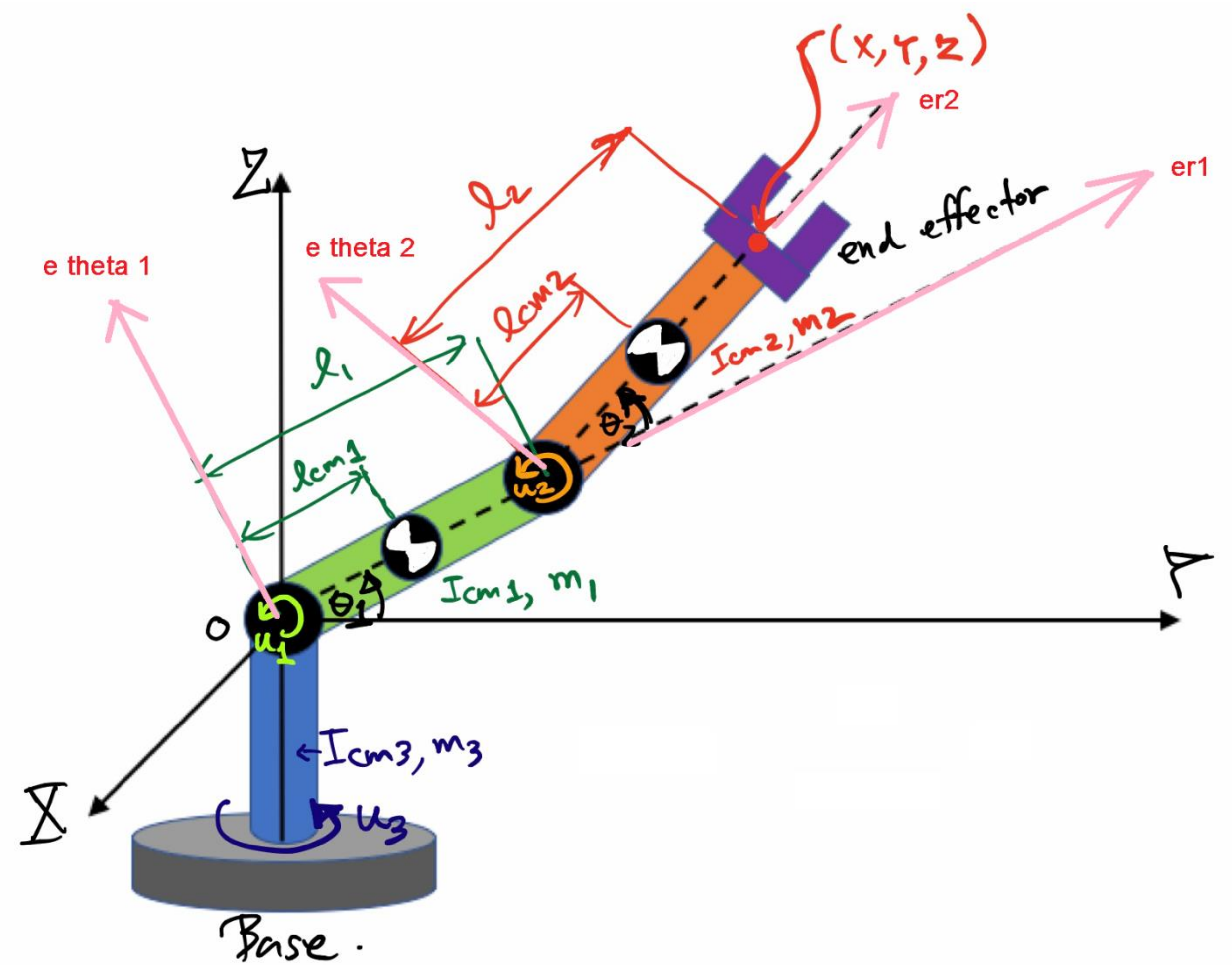


Desired path vs calculated path



Theta 1,2 and 3

Equations of motion's analysis



coordinating systems and a robot's diagram

First, we define generalize coordinate as θ_1, θ_2 and θ_3

The angular velocity of link 1 can be calculated as $\overrightarrow{\omega_1} = \dot{\theta}_3 \overrightarrow{e_z} + \dot{\theta}_1 \overrightarrow{e_x}$

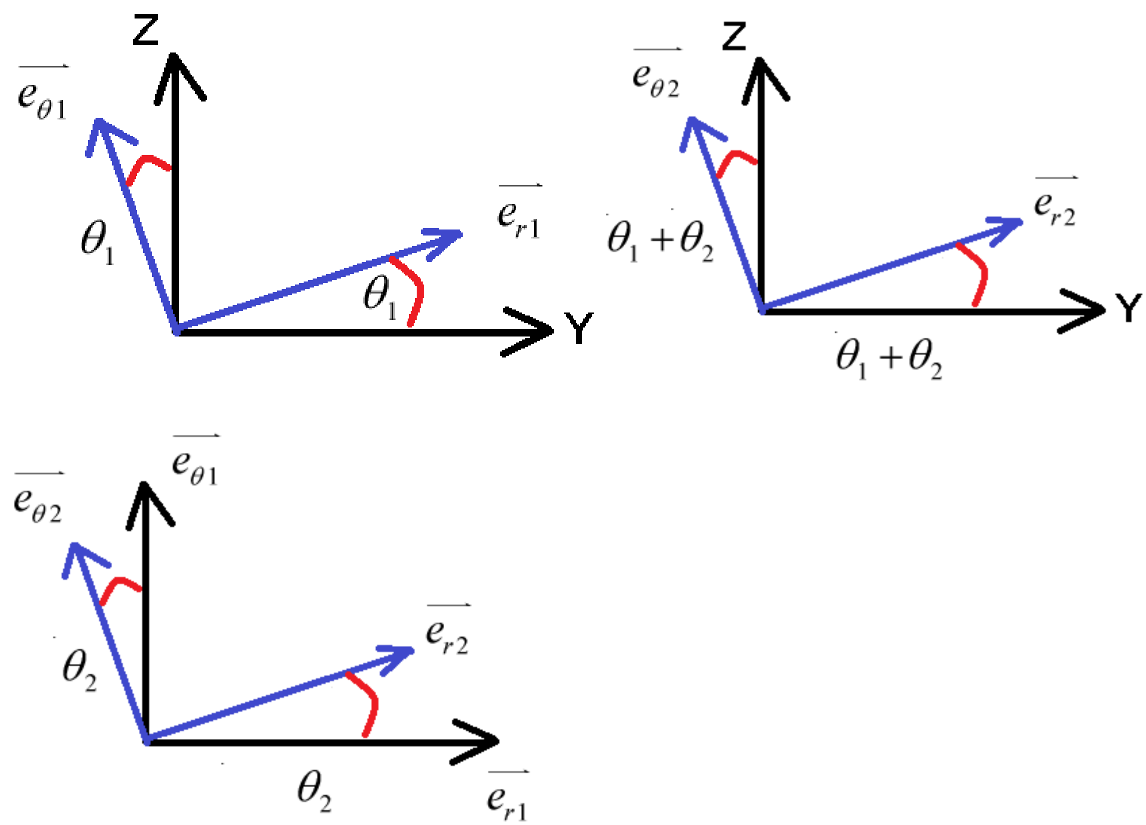
By using coordinate transformation we can rewrite $\overrightarrow{\omega_1}$ as

$\overrightarrow{\omega_1} = \dot{\theta}_3 \cos(\theta_1) \overrightarrow{e_{\theta 1}} + \dot{\theta}_3 \sin(\theta_1) \overrightarrow{e_{r1}} + \dot{\theta}_1 \overrightarrow{e_x}$ ----(9)

Then we can find $\overrightarrow{\omega_2}$ in the same way like $\overrightarrow{\omega_1}$ below

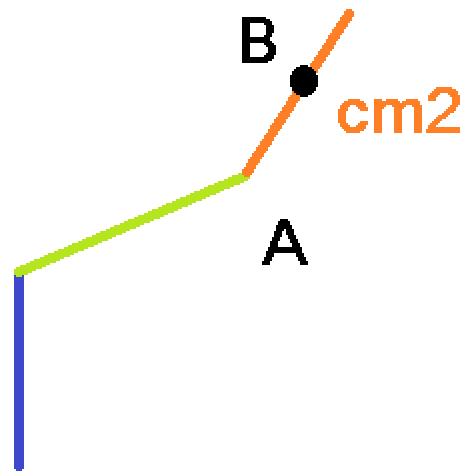
$$\overrightarrow{\omega_2} = \dot{\theta}_3 \overrightarrow{e_z} + (\dot{\theta}_1 + \dot{\theta}_2)\overrightarrow{e_x}$$

$$\overrightarrow{\omega_2} = \dot{\theta}_3 \sin(\theta_1 + \theta_2)\overrightarrow{e_{r2}} + \dot{\theta}_3 \cos(\theta_1 + \theta_2)\overrightarrow{e_{\theta 2}} + (\dot{\theta}_1 + \dot{\theta}_2)\overrightarrow{e_x} \text{ ----(10)}$$



Coordinate transformation

The velocity of the center of mass of the link 2 (cm2) can be calculated by relative velocity between B and A as follows



Two reference points

$$\vec{V}_B = \vec{V}_A + \dot{\theta}_2 l_{cm2} \vec{e}_{\theta 2}$$

; $\vec{V}_A = -\dot{\theta}_3 l_1 \cos(\theta_1)\vec{e}_x + \dot{\theta}_1 l_1 \vec{e}_{\theta 1}$

So $\vec{V}_{cm2} = -\dot{\theta}_3 l_1 \cos(\theta_1)\vec{e}_x + \dot{\theta}_1 l_1 \vec{e}_{\theta 1} + \dot{\theta}_2 l_{cm2} \vec{e}_{\theta 2}$

With coordinate transformation we can rewrite $\vec{V}_{cm2}$ as

$$\vec{V}_{cm2} = -\dot{\theta}_3 l_1 \cos(\theta_1)\vec{e}_x + \dot{\theta}_1 l_1 \sin(\theta_2)\vec{e}_{r2} + [\dot{\theta}_2 l_{cm2} + \dot{\theta}_1 l_1 \cos(\theta_2)]\vec{e}_{\theta 2}$$

Then we can calculate a dot product of $\vec{V}_{cm2}$ as follows

$$\vec{V}_{cm2} \bullet \vec{V}_{cm2} = [-\dot{\theta}_3 l_1 \cos(\theta_1)]^2 + [\dot{\theta}_1 l_1 \sin(\theta_2)]^2 + [\dot{\theta}_2 l_{cm2} + \dot{\theta}_1 l_1 \cos(\theta_2)]^2$$

$$\vec{V}_{cm2} \bullet \vec{V}_{cm2} = \dot{\theta}_3^{\,2} l_1^{\,2} \cos^2(\theta_1) + \dot{\theta}_1^{\,2} l_1^{\,2} + \dot{\theta}_2^{\,2} l_{cm2}^{\,2} + 2\dot{\theta}_1 \dot{\theta}_2 l_1 l_{cm2} \cos(\theta_2) \text{ ----(11)}$$

Kinetic energy of the link 1 can be calculated as follows

$$T_1 = \frac{1}{2} \vec{\omega}_1^{\,T} I_{1o} \vec{\omega}_1$$

; $\overrightarrow{\omega_1} = \begin{bmatrix} \dot{\theta}_3 \sin(\theta_1) \\ \dot{\theta}_3 \cos(\theta_1) \\ \dot{\theta}_1 \end{bmatrix}$, $I_{1o} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & I_\theta & 0 \\ 0 & 0 & I_x \end{bmatrix}$ with r_1, θ_1, x coordinate

So $T_1 = \frac{1}{2}\begin{bmatrix} \dot{\theta}_3 \sin(\theta_1) & \dot{\theta}_3 \cos(\theta_1) & \dot{\theta}_1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & I_\theta & 0 \\ 0 & 0 & I_x \end{bmatrix} \begin{bmatrix} \dot{\theta}_3 \sin(\theta_1) \\ \dot{\theta}_3 \cos(\theta_1) \\ \dot{\theta}_1 \end{bmatrix}$

; $I_\theta = I_x = I_{cm} + m_1 l_{cm1}^2 = 10 + 50(0.5)^2 kg.m^2 = 22.5kg.m^2$

So $T_1 = 11.25[\dot{\theta}_3^2 \cos^2(\theta_1) + \dot{\theta}_1^2]$

Kinetic energy of the link 2 can be calculated as follows

$$T_2 = \frac{1}{2}\overrightarrow{\omega_2}^T I_{2cm} \overrightarrow{\omega_2} + \frac{1}{2} m_2 \overrightarrow{V}_{cm2} \bullet \overrightarrow{V}_{cm2}$$

; $\overrightarrow{\omega_2} = \begin{bmatrix} \dot{\theta}_3 \sin(\theta_1 + \theta_2) \\ \dot{\theta}_3 \cos(\theta_1 + \theta_2) \\ \dot{\theta}_1 + \dot{\theta}_2 \end{bmatrix}$, $I_{2cm} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix}$ with r_2, θ_2, x coordinate and with

$\overrightarrow{V}_{cm2} \bullet \overrightarrow{V}_{cm2}$ from (11)

We will get T_2 as follows

$$T_2 = \frac{1}{2} m_2 [\dot{\theta}_3^2 l_1^2 \cos^2(\theta_1) + \dot{\theta}_1^2 l_1^2 + \dot{\theta}_2^2 l_{cm2}^2 + 2\dot{\theta}_1 \dot{\theta}_2 l_1 \mathrm{l}_{cm2} \cos(\theta_2)] + 5[\dot{\theta}_3^2 \cos^2(\theta_1 + \theta_2) + (\dot{\theta}_1 + \dot{\theta}_2)^2]$$

Kinetic energy of the link 3 can be calculated as follows

$$T_3 = \frac{1}{2}\overrightarrow{\omega_3}^T I_{3cm} \overrightarrow{\omega_3}$$

; $\overrightarrow{\omega_3} = \dot{\theta}_3^2 \overrightarrow{e_z}$ and $I_{3cm} = 10kg.m^2$ in x,y,z coordinate

So $T_3 = 5\dot{\theta}_3^2$

Kinetic energy of the system = $T_1+T_2+T_3$=T

$$T=11.25[\dot{\theta}_3^2\cos^2(\theta_1)+\dot{\theta}_1^2]+\frac{1}{2}m_2[\dot{\theta}_3{}^2l_1^2\cos^2(\theta_1)+\dot{\theta}_1{}^2l_1^2+\dot{\theta}_2{}^2l_{cm2}{}^2+2\dot{\theta}_1\dot{\theta}_2l_1\,\mathrm{l}_{cm2}\cos(\theta_2)]+5[\dot{\theta}_3{}^2\cos^2(\theta_1+\theta_2)+(\dot{\theta}_1+\dot{\theta}_2)^2]+5\dot{\theta}_3^2$$

----(12) **

Potential energy of the system can be calculated by sum of the potential energy of each link $V=V_1+V_2+V_3$

; $V_1=m_1gl_{cm1}\sin(\theta_1)$, $V_2=m_2g[\mathrm{l}_1\sin(\theta_1)+\mathrm{l}_{cm2}\sin(\theta_1+\theta_2)]$ and $V_3=-m_3gl_{cm3}$

So $V=m_1gl_{cm1}\sin(\theta_1)+m_2g[\mathrm{l}_1\sin(\theta_1)+\mathrm{l}_{cm2}\sin(\theta_1+\theta_2)]-m_3gl_{cm3}$----(13) **

Generalize forces can be calculated as follows

$$\delta w=u_1\delta\theta_1-u_2\delta\theta_1+u_2(\delta\theta_1+\delta\theta_3)+u_3\delta\theta_3$$

$$\delta w=u_1\delta\theta_1+u_2\delta\theta_2+u_3\delta\theta_3=Q_{\theta 1}\delta\theta_1+Q_{\theta 2}\delta\theta_2+Q_{\theta 3}\delta\theta_3$$

So $Q_{\theta 1}=u_1$, $Q_{\theta 2}=u_2$ and $Q_{\theta 3}=u_3$ **

Lagrange's equations

$$\frac{d}{dt}\left(\frac{\partial T}{\partial\dot{\theta}_1}\right)-\frac{\partial T}{\partial\theta_1}+\frac{\partial V}{\partial\theta_1}=Q_{\theta 1}$$ ----(14)

$$\frac{d}{dt}\left(\frac{\partial T}{\partial\dot{\theta}_2}\right)-\frac{\partial T}{\partial\theta_2}+\frac{\partial V}{\partial\theta_2}=Q_{\theta 2}$$ ----(15)

$$\frac{d}{dt}\left(\frac{\partial T}{\partial\dot{\theta}_3}\right)-\frac{\partial T}{\partial\theta_3}+\frac{\partial V}{\partial\theta_3}=Q_{\theta 3}$$ ----(16)

Then find each term in Lagrange's equations as follows

$$\frac{\partial T}{\partial \dot{\theta}_1} = 22.5\dot{\theta}_1 + m_2\dot{\theta}_1 l_1^2 + m_2\dot{\theta}_2 l_2 l_{cm2}\cos(\theta_2) + 10\dot{\theta}_1 + 10\dot{\theta}_2$$

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\theta}_1}\right) = (32.5 + \mathrm{m}_2\mathrm{l}_1^2)\ddot{\theta}_1 - [\mathrm{m}_2\mathrm{l}_2\mathrm{l}_{cm2}\sin(\theta_2)]\dot{\theta}_2^2 + [m_2 l_2 l_{cm2}\cos(\theta_2) + 10]\ddot{\theta}_2 \text{----(17) **}$$

$$\frac{\partial T}{\partial \theta_1} = \sin(\theta_1)\cos(\theta_1)[-22.5\dot{\theta}_3^2 - \mathrm{m}_2\mathrm{l}_1^2\,\dot{\theta}_3^2] - 10\sin(\theta_1+\theta_2)\cos(\theta_1+\theta_2)\dot{\theta}_3^2 \text{----(18) **}$$

$$\frac{\partial V}{\partial \theta_1} = m_1 g l_{cm1}\cos(\theta_1) + m_1 g l_1\cos(\theta_1) + m_2 g l_{cm2}\cos(\theta_1+\theta_2) \text{----(19) **}$$

$$\frac{\partial T}{\partial \dot{\theta}_2} = m_2\dot{\theta}_2 l_{cm2}^2 + m_2\dot{\theta}_1 l_1 l_{cm2}\cos(\theta_2) + 10\dot{\theta}_1 + 10\dot{\theta}_2$$

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\theta}_2}\right) = [m_2 l_1\,\mathrm{l}_{cm2}\cos(\theta_2) + 10]\ddot{\theta}_1 + [\mathrm{m}_2\mathrm{l}_{cm2}^2 + 10]\ddot{\theta}_2 - m_2 l_1 l_{cm2}\sin(\theta_2)\dot{\theta}_1\dot{\theta}_2 \text{----(20) **}$$

$$\frac{\partial T}{\partial \theta_2} = -10\sin(\theta_1+\theta_2)\cos(\theta_1+\theta_2)\dot{\theta}_3^2 - \mathrm{m}_2\, l_1 l_{cm2}\sin(\theta_2)\dot{\theta}_1\dot{\theta}_2 \text{----(21) **}$$

$$\frac{\partial V}{\partial \theta_2} = m_2 g l_{cm2}\cos(\theta_1+\theta_2) \text{----(22) **}$$

$$\frac{\partial T}{\partial \dot{\theta}_3} = [22.5\cos^2(\theta_1) + \mathrm{m}_2\mathrm{l}_1^2\cos^2(\theta_1) + 10\cos^2(\theta_1+\theta_2) + 10]\dot{\theta}_3$$

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\theta}_3}\right) = \left(\cos^2(\theta_1)\ddot{\theta}_3 - 2\cos(\theta_1)\sin(\theta_1)\dot{\theta}_1\dot{\theta}_3\right)\left[22.5 + m_2 l_1^2\right] + 10\left\{\cos^2(\theta_1+\theta_2)\ddot{\theta}_3 - 2\sin(\theta_1+\theta_2)\cos(\theta_1+\theta_2)[\dot{\theta}_1+\dot{\theta}_2]\dot{\theta}_3\right\} + 10\ddot{\theta}_3$$

----(23) **

$\frac{\partial T}{\partial \theta_3} = 0$----(24) **

$\frac{\partial V}{\partial \theta_3} = 0$----(25) **

Then find equations of motion by placing (17),(18),(19) into (14) , placing (20),(21),(22) into (15) and placing (23),(24),(25) into (16)

From (14)

$[(32.5 + \mathrm{m}_2 \mathrm{l}_1^2)\ddot{\theta}_1 - [\mathrm{m}_2 \mathrm{l}_2 \mathrm{l}_{cm2} \sin(\theta_2)]\dot{\theta}_2^2 + [m_2 l_2 l_{cm2} \cos(\theta_2) + 10]\ddot{\theta}_2]$ -

$[\sin(\theta_1)\cos(\theta_1)[-22.5\dot{\theta}_3^2 - \mathrm{m}_2 \mathrm{l}_1^2\, \dot{\theta}_3^2] - 10\sin(\theta_1 + \theta_2)\cos(\theta_1 + \theta_2)\dot{\theta}_3^2]$ +

$[m_1 g l_{cm1} \cos(\theta_1) + m_1 g l_1 \cos(\theta_1) + m_2 g l_{cm2} \cos(\theta_1 + \theta_2)]$ = u_1 ----(26) ***EOM1

From (15)

$[[m_2 l_1\, \mathrm{l}_{cm2} \cos(\theta_2) + 10]\ddot{\theta}_1 + [\mathrm{m}_2 \mathrm{l}_{cm2}^2 + 10]\ddot{\theta}_2 - m_2 l_1 l_{cm2} \sin(\theta_2)\dot{\theta}_1\dot{\theta}_2]$ –

$[-10\sin(\theta_1 + \theta_2)\cos(\theta_1 + \theta_2)\dot{\theta}_3^2 - \mathrm{m}_2\, l_1 l_{cm2} \sin(\theta_2)\dot{\theta}_1\dot{\theta}_2]$

+ $[m_2 g l_{cm2} \cos(\theta_1 + \theta_2)]$ = u_2----(27) ***EOM2

From (16)

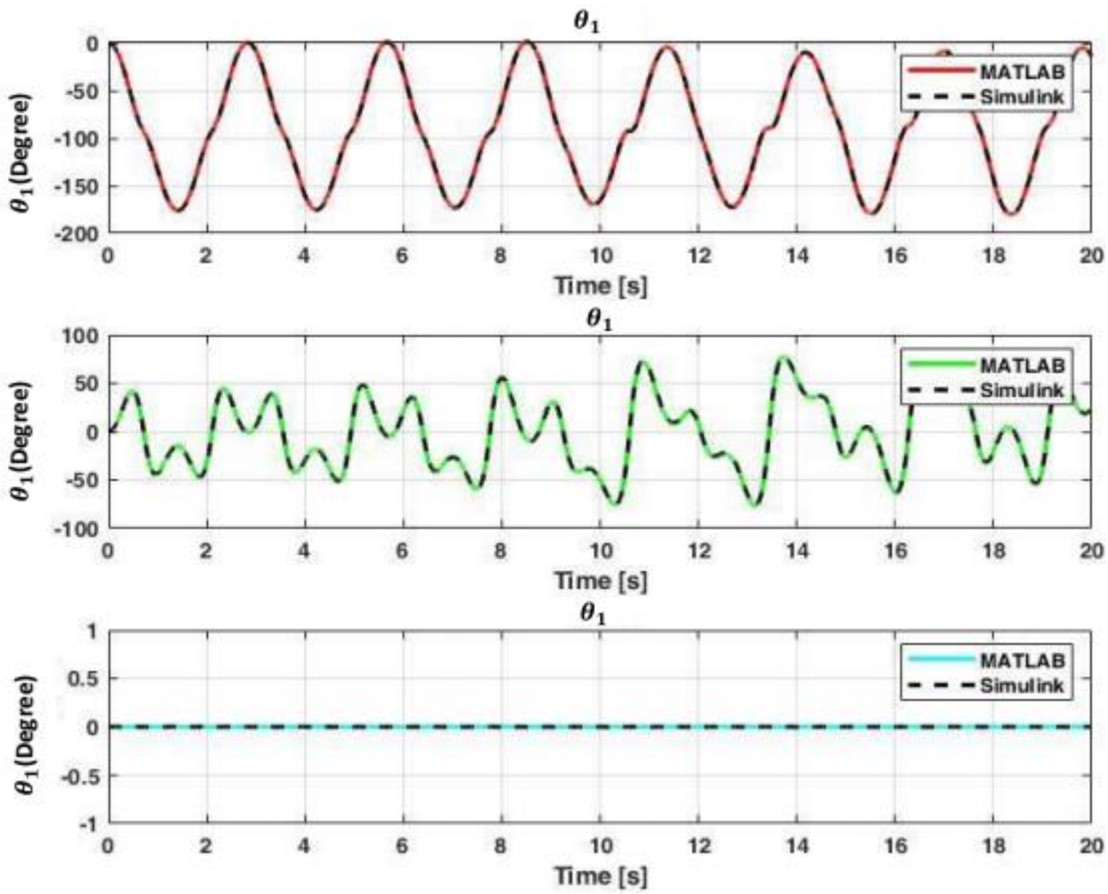
$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\theta}_3}\right) = \left(\cos^2(\theta_1)\ddot{\theta}_3 - 2\cos(\theta_1)\sin(\theta_1)\dot{\theta}_1\dot{\theta}_3\right)\left[22.5 + m_2 l_1^2\right] + 10\left\{\cos^2(\theta_1 + \theta_2)\ddot{\theta}_3 - 2\sin(\theta_1 + \theta_2)\cos(\theta_1 + \theta_2)[\dot{\theta}_1 + \dot{\theta}_2]\dot{\theta}_3\right\} + 10\ddot{\theta}_3$$

= u_3

----(28) ***EOM3

Matlab: simulation of the system relating to EOM

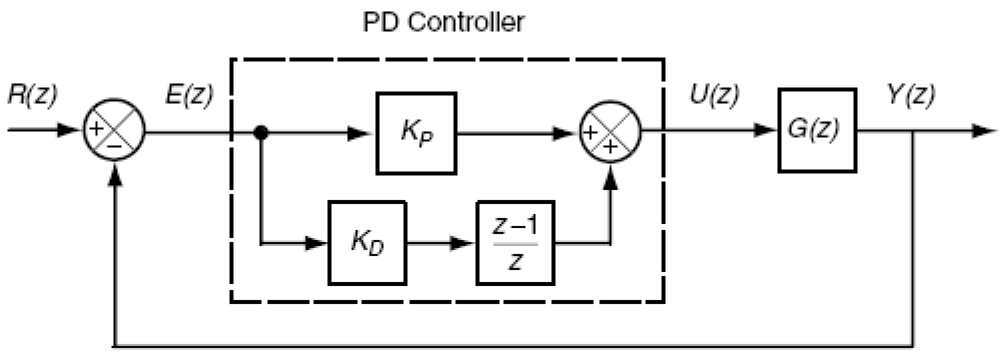
Theta 1,2 and 3 from Simulink model and Matlab (ode45) are almost the same. Note that in this case torque u1, u2 and u3 = 0 (Free falling)



Thetas from Matlab(ode45) vs from Simulink model

Controller's analysis

By using PD control, we can control robot's trajectory by controlling theta 1,2 and 3. Note that theta 1,2 and 3 can control with external torques u1, u2 and u3

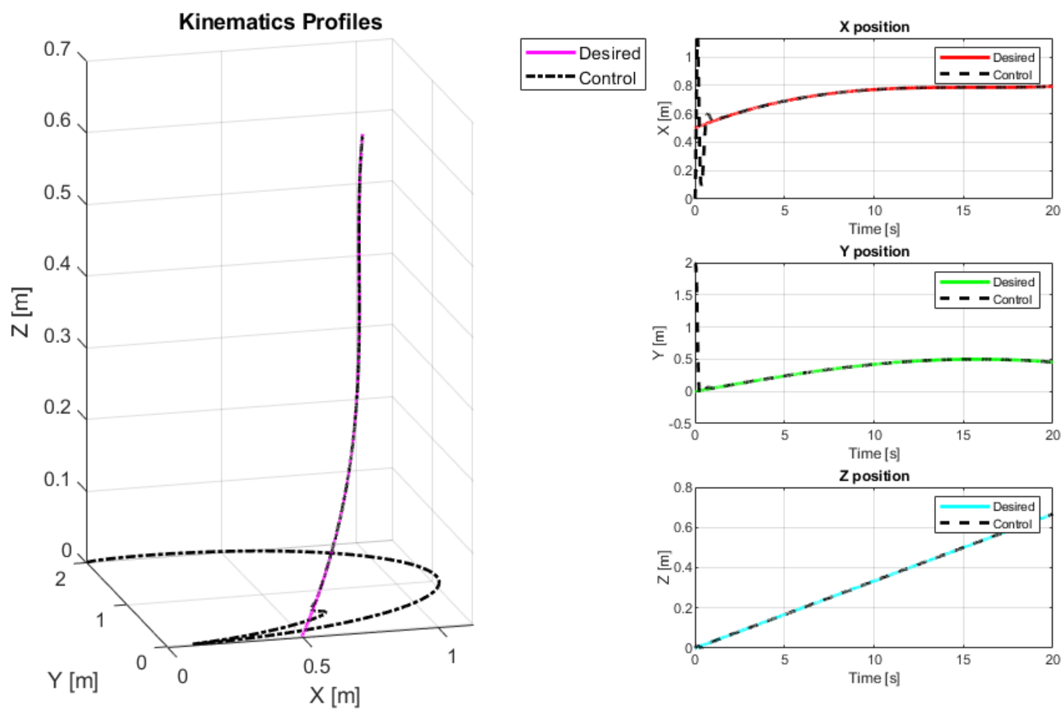


PD control

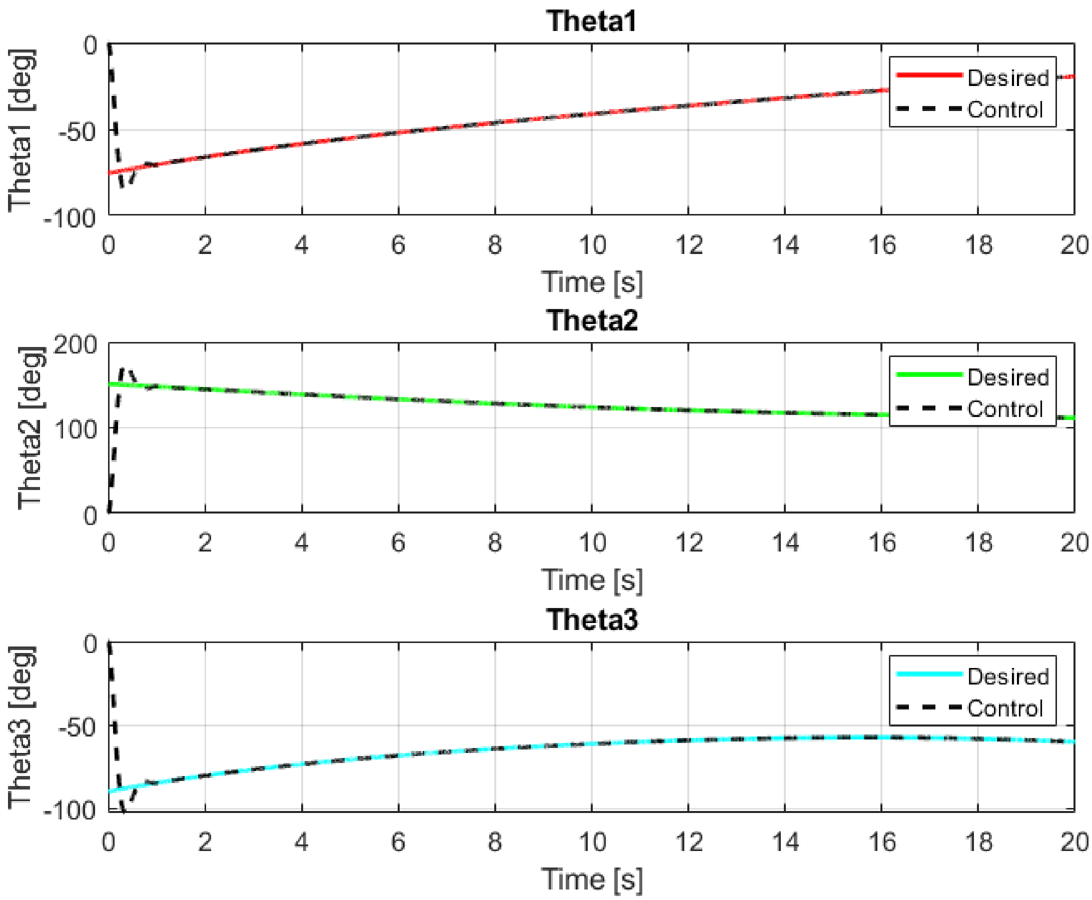
R(Z) = external torques , G(Z) = thetas

Matlab: simulation of the system with controllers

By using PD control, At first, both controlled path and controlled thetas are not the same as desired values but when the time passes, the result is acceptable.



Desired path vs controlled path



Desired thetas vs controlled thetas

References

1. 2103-617 Advanced Dynamics , Thitima Jintanawan.

2. Robotics Modeling, Planning and control, Springer.

3. 3-DOFs Scalar Robot Manipular Dynamics and Control

Created by Kan Kanjanapas (Ph.D.).

4. Moon, F. C., Applied Dynamics: With Applications to Multibody and Mechatronic Systems, John Wiley & Sons,

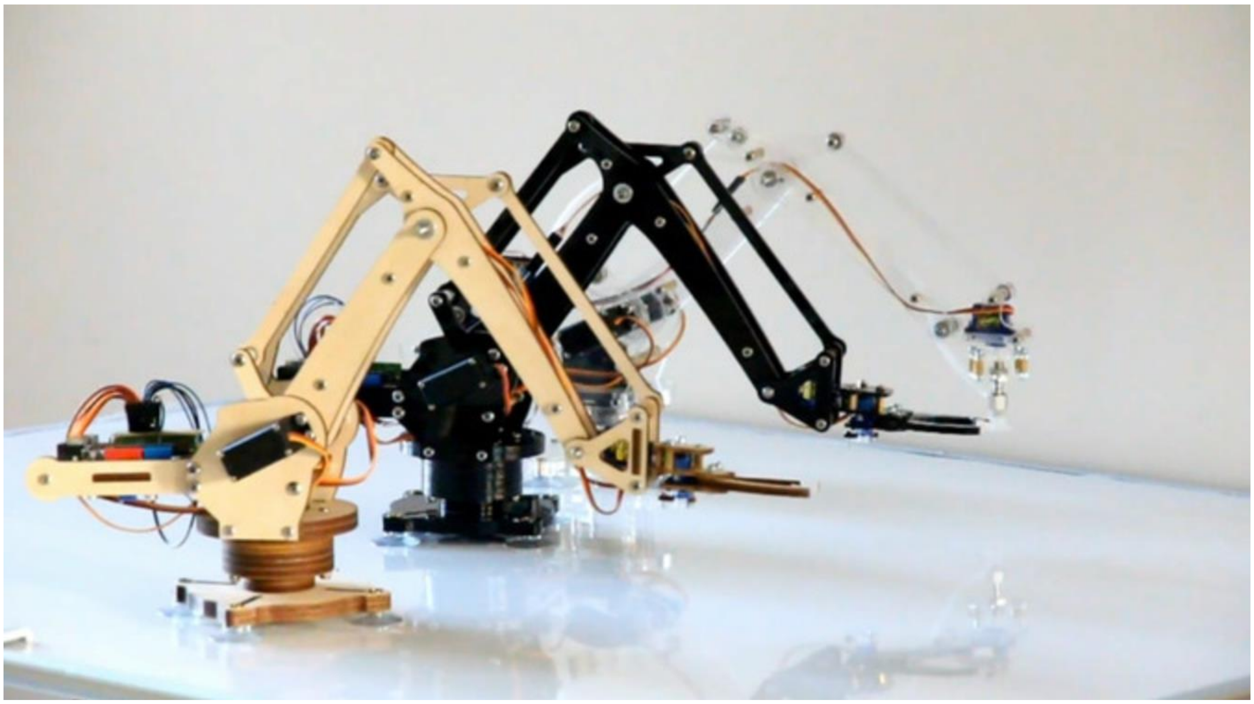
1998.

5. Ginsberg, J. H., Advanced Engineering Dynamics, Cambridge, 1998.

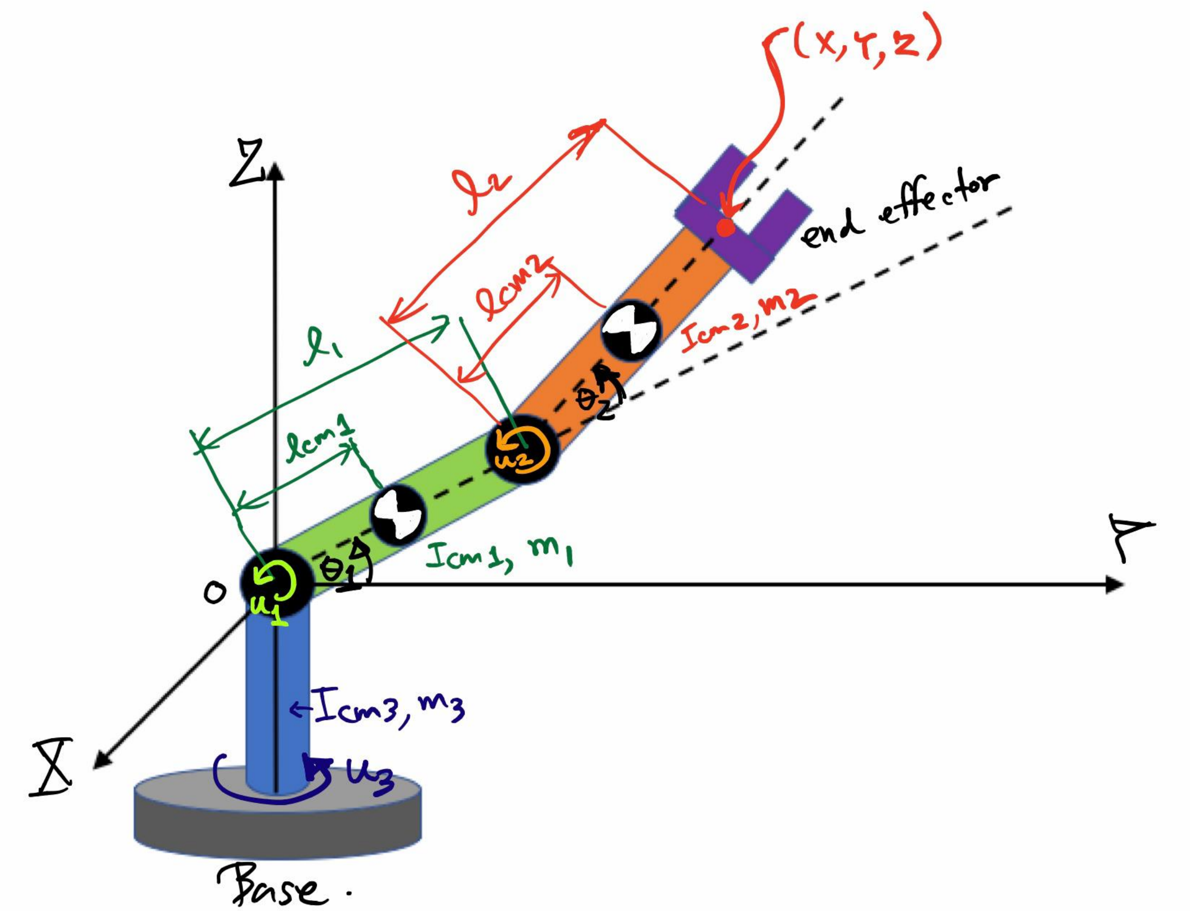
6. Greenwood, D. T., Principles of Dynamics, Prentice Hall, 1988.

Appendix

1. A real robot’s picture



2. A coordinating system and a robot's diagram



Note that (XYZ) is a coordinate system that is fixed to the O point without moving with the robot and (xyz) is a coordinate at the O point as well but rotates with the robot.

3. All robot's parameters

Parameters:

l1:	Length of link 1 (Green) [1 m]
l2:	Length of link 2 (Orange) [1 m]
lcm1:	Center of mass position of link 1 with respect to joint 1 (@u1) [0.5 m]
lcm2:	Center of mass position of link 2 with respect to joint 2 (@u2) [0.5 m]
theta1, theta2, theta3:	Angle of joint 1,2 and 3 respectively [rad, deg]
m1, m2, m3:	Mass of joint 1,2 and 3 respectively [50, 50, 50 Kg]
Icm1, Icm2, Icm3:	Moment of inertia @ CM [10, 10, 10 Kg.m^2]
(x,y,z):	End effector position [m]
XYZ:	Global coordinate system as world reference
u1, u2, u3:	Control inputs = Motor torque [N.m]
kr1, kr2, kr3:	Gear ratio of motor at joint 1,2,3 [kri = 100/1]
Imotor1, Imotor2, Imotor3:	Moment of inertia of motor at joint 1,2,3 [0.01, 0.01. 0.01 Kg.m^2]
g:	Gravity (9.81 m/s^2)

4. Matlab program (forward and inverse kinematics)

Parameters

```
params = [];
params.l1       = 1;        % Length of link 1 [m]
params.l2       = 1;        % Length of link 2 [m]
params.l3       = 0;        % Length of link 3 [m]
params.lcm1     = 0.5;      % CM position of link 1 [m]
params.lcm2     = 0.5;      % CM position of link 2 [m]
params.lcm3     = 0.5*0;      % CM position of link 3 [m]
params.m1       = 50;       % Mass of link 1 [kg]
params.m2       = 50;       % Mass of link 2 [kg]
params.m3       = 50*0;       % Mass of link 2 [kg]
params.I_lcm1   = 10;       % Moment of inertia of link 1 [kg.m^2]
params.I_lcm2   = 10;       % Moment of inertia of link 2 [kg.m^2
params.I_lcm3   = 10;       % Moment of inertia of link 3 [kg.m^2]]
params.kr1      = 100*0;      % Gear ratio of motor at joint 1 [-]
params.kr2      = 100*0;      % Gear ratio of motor at joint 2 [-]
params.kr3      = 100*0;      % Gear ratio of motor at joint 3 [-]
params.m_motor1 = 5*0;        % Mass of rotor at joint 1 [kg]
params.m_motor2 = 5*0;        % Mass of rotor at joint 2 [kg]
params.m_motor3 = 5*0;        % Mass of rotor at joint 3 [kg]
params.I_motor1 = 0.01*0;     % Moment of inertia of rotor at joing 1 [kg.m^2]
params.I_motor2 = 0.01*0;     % Moment of inertia of rotor at joing 2 [kg.m^2]
params.I_motor3 = 0.01*0;     % Moment of inertia of rotor at joing 3 [kg.m^2]
params.g        = 9.81;     % Gravity [m/s^2]
```

Setup Kinematics

```
Ts = 10^-3;
tmax = 20;
tspan = [0:Ts:tmax]';
n = length(tspan);
```

Desired Trajectory

```
r = 0.5;
Xd = r*cos(tspan/10)+0.05*tspan;
Yd = r*sin(tspan/10);
Zd = tspan/30;

% Inverse kinematics
[theta1_d,theta2_d,theta3_d] = inverseki(Xd,Yd,Zd,params);
%Foward kinematics
[Xdd,Ydd,Zdd] = forwardki(theta1_d,theta2_d,theta3_d,params);
```

Plot Kinematics Profiles

```
% plot profile
figure;
subplot(3,5,[1,2,3,6,7,8,11,12,13])
set(gcf, 'Position', [350 100 2000 1200]/2);
plot3(Xd,Yd,Zd,'LineWidth', 2, 'Color', 'g');
hold on
plot3(Xdd,Ydd,Zdd,'LineWidth', 2,'LineStyle', '--', 'Color', 'k');
legend('Desired Path','Calculated Path')
xlabel('X [m]'); ylabel('Y [m]'); zlabel('Z [m]')
set(gca, 'FontSize', 12); grid on;title('Desired Profiles');
view(-10,10)
grid on;
% plot position
subplot(3,5,[4,5])
plot(tspan,Xd, 'LineWidth', 2, 'Color', 'y');
hold on
plot(tspan,Xdd, 'LineWidth', 2,'LineStyle', '--', 'Color', 'r');
xlabel('Time [s]'); ylabel('X [m]');
set(gca, 'FontSize', 8); grid on;title('X position');
legend('Desired Path','Calculated Path')

subplot(3,5,[9,10])
plot(tspan,Yd, 'LineWidth', 2, 'Color', 'y');
hold on
plot(tspan,Ydd, 'LineWidth', 2,'LineStyle', '--', 'Color', 'r');
xlabel('Time [s]'); ylabel('Y [m]');
set(gca, 'FontSize', 8); grid on;title('Y position');
legend('Desired Path','Calculated Path')

subplot(3,5,[14,15])
```

```
plot(tspan,Zd, 'LineWidth', 2, 'Color', 'y');
hold on
plot(tspan,Zdd, 'LineWidth', 2,'LineStyle', '--', 'Color', 'r');
xlabel('Time [s]'); ylabel('Z [m]');
set(gca, 'FontSize', 8); grid on;title('Z position');
legend('Desired Path','Calculated Path')

% plot theta
figure;
set(gcf, 'Position', [350 100 1500 1200]/2);
subplot(3,1,1)
plot(tspan,theta1_d*180/pi, 'LineWidth', 2, 'Color', 'r'); xlabel('Time [s]'); ylabel('theta1
[deg]');
set(gca, 'FontSize', 10); grid on;title('Theta1');
subplot(3,1,2)
plot(tspan,theta2_d*180/pi, 'LineWidth', 2, 'Color', 'b'); xlabel('Time [s]'); ylabel('theta2
[deg]');
set(gca, 'FontSize', 10); grid on;title('Theta2');
subplot(3,1,3)
plot(tspan,theta3_d*180/pi, 'LineWidth', 2, 'Color', 'g'); xlabel('Time [s]'); ylabel('theta3
[deg]');
set(gca, 'FontSize', 10); grid on;title('Theta3');
```

```
function [theta1,theta2,theta3] = inverseki(x,y,z,params)
l1 = params.l1;
l2 = params.l2;
r1 = sqrt(x.^2+y.^2);
r2 = z;
r3 = sqrt(r1.^2+r2.^2);
theta3 = atan2(-x,y);
theta3 = unwrap(theta3);
theta1 = (atan2(r2,r1)-acos((l2^2-l1^2-r3.^2)./(-2*l1.*r3)));
theta1 = unwrap(theta1);
theta2 = pi-acos((r3.^2-l1^2-l2^2)/(-2*l1*l2));
theta2 = unwrap(theta2);
end

function [X,Y,Z]=forwardki(a,b,c,params)
l1 = params.l1;
l2 = params.l2;
X=-(l1*cos(a)+l2.*cos(a+b)).*sin(c);
Y=(l1*cos(a)+l2.*cos(a+b)).*cos(c);
Z=l1*sin(a)+l2.*sin(a+b);
end
```

5. Matlab program (simulation of the system relating to EOM)

Desired Kinematics Profiles (Theta_d, Theta_d_dot, Theta_d_ddot)

```
% Start at Zero Velocity
theta1_d_dot = [0; diff(theta1_d)/Ts];
theta2_d_dot = [0; diff(theta2_d)/Ts];
theta3_d_dot = [0; diff(theta3_d)/Ts];

% Start at Zero Acceleration
theta1_d_ddot = [0; diff(theta1_d_dot)/Ts];
theta2_d_ddot = [0; diff(theta2_d_dot)/Ts];
theta3_d_ddot = [0; diff(theta3_d_dot)/Ts];

Kinematics_Profiles = [];

Kinematics_Profiles.theta1_d      = theta1_d;
Kinematics_Profiles.theta1_d_dot  = theta1_d_dot;
Kinematics_Profiles.theta1_d_ddot = theta1_d_ddot;

Kinematics_Profiles.theta2_d      = theta2_d;
Kinematics_Profiles.theta2_d_dot  = theta2_d_dot;
Kinematics_Profiles.theta2_d_ddot = theta2_d_ddot;

Kinematics_Profiles.theta3_d      = theta3_d;
Kinematics_Profiles.theta3_d_dot  = theta3_d_dot;
Kinematics_Profiles.theta3_d_ddot = theta3_d_ddot;

figure;
set(gcf, 'Position', [350 100 2000 1200]/2);
for ii = 1:1

    % Theta1 --------------------------------------------------------------

    subplot(3,3,1); plot(tspan, theta1_d*180/pi, 'LineWidth', 2, 'Color', 'b');
    xlabel('Time'); ylabel('Theta1d [deg]'); set(gca, 'FontSize', 11);title('Theta1d'); grid on;

    subplot(3,3,4); plot(tspan, theta1_d_dot*180/pi, 'LineWidth', 2, 'Color', 'b');
    xlabel('Time'); ylabel('Theta1d-dot [deg/s]'); set(gca, 'FontSize', 11);title('Theta1d-dot');
grid on;

    subplot(3,3,7); plot(tspan, theta1_d_ddot*180/pi, 'LineWidth', 2, 'Color', 'b');
    xlabel('Time'); ylabel('Theta1d-ddot [deg/s^2]'); set(gca, 'FontSize', 11);title('Theta1d-
ddot');grid on;ylim([-10 10]);

    % Theta2 --------------------------------------------------------------
    subplot(3,3,2); plot(tspan, theta2_d*180/pi, 'LineWidth', 2, 'Color', [0  0.5  0]);
    xlabel('Time'); ylabel('Theta2d [deg]'); set(gca, 'FontSize', 11);title('Theta2d'); grid on;

    subplot(3,3,5); plot(tspan, theta2_d_dot*180/pi, 'LineWidth', 2, 'Color', [0  0.5  0]);
    xlabel('Time'); ylabel('Theta2d-dot [deg/s]'); set(gca, 'FontSize', 11);title('Theta2d-dot');
grid on;
```

```
    subplot(3,3,8); plot(tspan, theta2_d_ddot*180/pi, 'LineWidth', 2, 'Color', [0  0.5  0]);
    xlabel('Time'); ylabel('Theta2d-ddot [deg/s^2]'); set(gca, 'FontSize', 11);title('Theta2d-ddot'); grid on;ylim([-10 10]);

    % Theta3 --------------------------------------------------------------
    subplot(3,3,3); plot(tspan, theta3_d*180/pi, 'LineWidth', 2, 'Color', [1  0  0]);
    xlabel('Time'); ylabel('Theta3d [deg]'); set(gca, 'FontSize', 11);title('Theta3d');grid on;

    subplot(3,3,6); plot(tspan, theta3_d_dot*180/pi, 'LineWidth', 2, 'Color', [1  0  0]);
    xlabel('Time'); ylabel('Theta3d-dot [deg/s]'); set(gca, 'FontSize', 11);title('Theta3d-dot');grid on;

    subplot(3,3,9); plot(tspan, theta3_d_ddot*180/pi, 'LineWidth', 2, 'Color', [1  0  0]);
    xlabel('Time'); ylabel('Theta3d-ddot [deg/s^2]'); set(gca, 'FontSize', 11);title('Theta3d-ddot');grid on;ylim([-10 10]);
end
```

Solve EOM (Open Loop)

```
%initial condition [position:velocity]
X0 = [pi/2;0;0;0;0;0];

fun = @(t,X)advdyManipulator(t, X, params);
[tout, Xout] = ode45(fun, tspan, X0, []);

theta1     = Xout(:,1);
theta1_dot = Xout(:,2);
theta2     = Xout(:,3);
theta2_dot = Xout(:,4);
theta3     = Xout(:,5);
theta3_dot = Xout(:,6);
```

Simulation Kinematics Plots

```
ind_plot = 1;
if(ind_plot)

figure;
set(gcf, 'Position', [500 300 2000 1200]/2);
grid on;
axis(2*[-1 1 -1 1 -1 1]);
grid on;
for ii = 1:100:length(theta1)
    %link 1
     plot3([0 -params.l1*cos(theta1(ii))*sin(theta3(ii))],...
          [0 params.l1*cos(theta1(ii))*cos(theta3(ii))],...
          [0 params.l1*sin(theta1(ii))],...
           'linewidth', 2, 'Color', 'r');
    hold all;
    grid on;
```

```
    xlabel('X [m]');
    ylabel('Y [m]');
    zlabel('Z [m]');
    set(gca, 'FontSize', 16);
    view(0,0);
    %motor 2
     plot3(-params.l1*cos(theta1(ii))*sin(theta3(ii)),...
           params.l1*cos(theta1(ii))*cos(theta3(ii)),...
           params.l1*sin(theta1(ii)),...
           'lineStyle', 'None', 'Marker', 'o', 'MarkerSize', 15, 'MarkerFaceColor', 'r',
'MarkerEdgeColor', 'r');
    %link 2
    plot3([-params.l1*cos(theta1(ii))*sin(theta3(ii)) -
(params.l1*cos(theta1(ii))+params.l2*cos(theta1(ii)+theta2(ii)))*sin(theta3(ii))],...
          [params.l1*cos(theta1(ii))*cos(theta3(ii))
(params.l1*cos(theta1(ii))+params.l2*cos(theta1(ii)+theta2(ii)))*cos(theta3(ii))],...
          [params.l1*sin(theta1(ii))
params.l1*sin(theta1(ii))+params.l2*sin(theta1(ii)+theta2(ii))],...
        'linewidth', 2, 'Color', 'g');
    %End effector
    plot3(-(params.l1*cos(theta1(ii))+params.l2*cos(theta1(ii)+theta2(ii)))*sin(theta3(ii)),...
          (params.l1*cos(theta1(ii))+params.l2*cos(theta1(ii)+theta2(ii)))*cos(theta3(ii)),...
           params.l1*sin(theta1(ii))+params.l2*sin(theta1(ii)+theta2(ii)),...
           'lineStyle', 'None', 'Marker', 'o', 'MarkerSize', 15, 'MarkerFaceColor','g',
'MarkerEdgeColor', 'g');
    %Axis
    plot3([-2 2],[0 0],[0 0],'LineWidth', 2, 'Color', 'k');
    plot3([0 0],[-2 2],[0 0],'LineWidth', 2, 'Color', 'k');
    plot3([0 0],[0 0] ,[-2 2],'LineWidth', 2, 'Color', 'k');
    pause( 0.1 );
    if (ii~= length(theta1))
        clf;
    else
        % Do nothing, Do not clear figure if not final post
    end

end

end % if(ind_plot)
```

Simmechanic Free fall

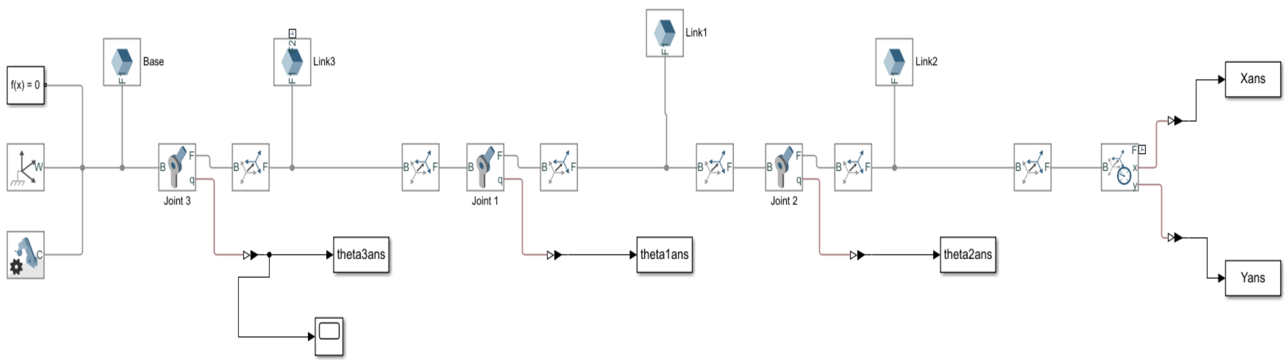
```
sim('Scalar3DOF_final');
figure;
set(gcf, 'Position', [350 100 1500 1200]/2);
% X
subplot(3,1,1)
plot(tspan,theta1*180/pi, 'LineWidth', 2, 'Color', 'r'); xlabel('Time [s]'); ylabel('Theta1
[deg]');
hold on
plot(theta1ans.time,theta1ans.data*180/pi, 'LineWidth', 2, 'LineStyle', '--', 'Color', 'k');
set(gca, 'FontSize', 10); grid on; title('Theta1');
```

```
legend('MATLAB','Simulink');

% Y
subplot(3,1,2)
plot(tspan,theta2*180/pi, 'LineWidth', 2, 'Color', 'g'); xlabel('Time [s]'); ylabel('Theta2
[deg]');
hold on
plot(theta2ans.time,theta2ans.data*180/pi, 'LineWidth', 2, 'LineStyle', '--', 'Color', 'k');
set(gca, 'FontSize', 10); grid on; title('Theta2');
legend('MATLAB','Simulink');

% Z
subplot(3,1,3)
plot(tspan,theta3*180/pi, 'LineWidth', 2, 'Color', 'c'); xlabel('Time [s]'); ylabel('Theta2
[deg]');
hold on
plot(theta3ans.time,theta3ans.data*180/pi, 'LineWidth', 2, 'LineStyle', '--', 'Color', 'k');
set(gca, 'FontSize', 10); grid on; title('Theta3');ylim([-1,1]);
legend('MATLAB','Simulink');
```

```
function X_dot = advdyManipulator(t, X, params)

% State ------------------------------------------------------------------

theta1      = X(1);
theta1_dot  = X(2);
theta2      = X(3);
theta2_dot  = X(4);
theta3      = X(5);
theta3_dot  = X(6);

% Parameters -------------------------------------------------------------
params;
l1          = params.l1;            % Length of link 1 [m]
l2          = params.l2;            % Length of link 2 [m]
l3          = params.l3;            % Length of link 3 [m]
lcm1        = params.lcm1;          % CM position of link 1 [m]
lcm2        = params.lcm2;          % CM position of link 2 [m]
m1          = params.m1;            % Mass of link 1 [kg]
m2          = params.m2;            % Mass of link 2 [kg]
m3          = params.m3;            % Mass of link 3 [kg]
Icm1        = params.I_lcm1;        % Moment of inertia of link 1 [kg.m^2]
Icm2        = params.I_lcm2;        % Moment of inertia of link 2 [kg.m^2]
Icm3        = params.I_lcm3;        % Moment of inertia of link 2 [kg.m^2]
kr1         = params.kr1;           % Gear ratio of motor at joint 1 [-]
kr2         = params.kr2;           % Gear ratio of motor at joint 2 [-]
kr3         = params.kr3;           % Gear ratio of motor at joint 2 [-]
M1          = params.m_motor1;      % Mass of rotor at joint 1 [kg]
M2          = params.m_motor2;      % Mass of rotor at joint 2 [kg]
M3          = params.m_motor3;      % Mass of rotor at joint 2 [kg]
I_M1        = params.I_motor1;      % Moment of inertia of rotor at joing 1
[kg.m^2]
```

```
I_M2         = params.I_motor2;      % Moment of inertia of rotor at joing 2
[kg.m^2]
I_M3         = params.I_motor3;      % Moment of inertia of rotor at joing 2
[kg.m^2]
g            = params.g;             % Gravity [m/s^2]
Io1          = Icm1+m1*lcm1^2;     % Moment of inertia of link 1 around origin
[kg.m^2]

%% Dynamics

% B(theta)*theta_ddot + C(theta,theta_dot)*theta_dot + Fv*theta_dot +
Fs*sign(theta_dot) + g(theta) = u - J'(theta)*he

b11 = 32.5+m2*l1^2;
b12 = 10+m2*l2*lcm2*cos(theta2);
b13 = 0;
b21 = b12;
b22 = 10+m2*lcm2^2;
b23 = 0;
b31 = 0;
b32 = 0;
b33 = 10+10*cos(theta1+theta2)^2+cos(theta1)^2*(22.5+m2*l1^2);

c11 = 0;
c12 = -m2*l2*lcm2*sin(theta2)*theta2_dot;
c13 = (sin(theta1)*cos(theta1)*22.5+m2*l1^2+10*sin(theta1+theta2)*...
    cos(theta1+theta2))*theta3_dot;
c21 = 2*m2*l1*lcm2*sin(theta2)*theta2_dot;
c22 = 0;
c23 = 10*sin(theta1+theta2)*cos(theta1+theta2)*theta3_dot;
c31 = 0;
c32 = 0;
c33 = -2*cos(theta1)*sin(theta1)*theta1_dot*(22.5+m2*l1^2)-
20*sin(theta1+theta2)...
    *cos(theta1+theta2)*(theta1_dot+theta2_dot);

g11 = m1*g*lcm1*cos(theta1)+m1*g*l1*cos(theta1)+m2*g*lcm2*cos(theta1+theta2);
g21 = m2*g*lcm2*cos(theta1+theta2);
g31 = 0;

B_theta = [b11 b12 b13;
           b21 b22 b23;
           b31 b32 b33];

C_theta_theta_dot = [c11 c12 c13;
                     c21 c22 c23;
                     c31 c32 c33];

G_theta = [g11;g21;g31];

%% Control Law
u = zeros(3,1);

Theta_DOT = [theta1_dot; theta2_dot; theta3_dot];
Theta_DDOT = pinv(B_theta)*[ u - C_theta_theta_dot*Theta_DOT - G_theta ];
```

```
%%

X_dot(1,1) = theta1_dot;
X_dot(2,1) = Theta_DDOT(1);
X_dot(3,1) = theta2_dot;
X_dot(4,1) = Theta_DDOT(2);
X_dot(5,1) = theta3_dot;
X_dot(6,1) = Theta_DDOT(3);

end
```



Simulink model 'Scalar3DOF_final'

6. Matlab program (simulation of the system with controllers)

Closed Loop EOM

```
X0 = [0;0;0;0;0;0];

fun_fb = @(t,X)advdyfeedback(t, X, tspan, params, Xd, Yd, Kinematics_Profiles);
[tout_fb, Xout_fb] = ode45(fun_fb, tspan, X0, []);

theta1_fb     = Xout_fb(:,1);
theta1_dot_fb = Xout_fb(:,2);
```

```
theta2_fb     = Xout_fb(:,3);
theta2_dot_fb = Xout_fb(:,4);
theta3_fb     = Xout_fb(:,5);
theta3_dot_fb = Xout_fb(:,6);

l1 = params.l1;
l2 = params.l2;
l3 = params.l3;

[X_fb,Y_fb,Z_fb] = forwardki(theta1_fb,theta2_fb,theta3_fb,params);
```

Plot closed-loop profile

```
% X-Y-Z destired vs control
figure;
set(gcf, 'Position', [350 100 2000 1200]/2);
subplot(3,5,[1,2,3,6,7,8,11,12,13])
plot3(Xd, Yd, Zd,'LineWidth', 2, 'Color', 'm');
hold on;
plot3(X_fb, Y_fb,Z_fb, 'LineWidth', 2, 'LineStyle', '-.', 'Color', 'k');
grid on
view(-15,10);
xlabel('X [m]'); ylabel('Y [m]'); zlabel('Z [m]')
set(gca, 'FontSize', 12); grid on;title('Kinematics Profiles');
legend('Desired','Control')

% X destired vs control
subplot(3,5,[4,5])
plot(tspan,Xd, 'LineWidth', 2, 'Color', 'r'); xlabel('Time [s]'); ylabel('X [m]');
hold on
plot(tspan,X_fb, 'LineWidth', 2, 'LineStyle', '--', 'Color', 'k');
set(gca, 'FontSize', 8); grid on;title('X position');
legend('Desired','Control')

% Y destired vs control
subplot(3,5,[9,10])
plot(tspan,Yd, 'LineWidth', 2, 'Color', 'g'); xlabel('Time [s]'); ylabel('Y [m]');
hold on
plot(tspan,Y_fb, 'LineWidth', 2, 'LineStyle', '--', 'Color', 'k');
set(gca, 'FontSize', 8); grid on;title('Y position');
legend('Desired','Control')

% Z destired vs control
subplot(3,5,[14,15])
plot(tspan,Zd, 'LineWidth', 2, 'Color', 'c'); xlabel('Time [s]'); ylabel('Z [m]');
hold on
plot(tspan,Z_fb, 'LineWidth', 2, 'LineStyle', '--', 'Color', 'k');
set(gca, 'FontSize', 8); grid on;title('Z position');
legend('Desired','Control')

% Theta destired vs control
figure;
```

```
set(gcf, 'Position', [350 100 1500 1200]/2);
legend('Desired','Control')

% theta1 destired vs control
subplot(3,1,1)
plot(tspan,theta1_d*180/pi, 'LineWidth', 2, 'Color', 'r'); xlabel('Time [s]'); ylabel('Theta1
[deg]');
hold on
plot(tspan,theta1_fb*180/pi, 'LineWidth', 2, 'LineStyle', '--', 'Color', 'k');
set(gca, 'FontSize', 12); grid on;title('Theta1');
legend('Desired','Control')

% theta2 destired vs control
subplot(3,1,2)
plot(tspan,theta2_d*180/pi, 'LineWidth', 2, 'Color', 'g'); xlabel('Time [s]'); ylabel('Theta2
[deg]');
hold on
plot(tspan,theta2_fb*180/pi, 'LineWidth', 2, 'LineStyle', '--', 'Color', 'k');
set(gca, 'FontSize', 12); grid on;title('Theta2');
legend('Desired','Control')

% theta3 destired vs control
subplot(3,1,3)
plot(tspan,theta3_d*180/pi, 'LineWidth', 2, 'Color', 'c'); xlabel('Time [s]'); ylabel('Theta3
[deg]');
hold on
plot(tspan,theta3_fb*180/pi, 'LineWidth', 2, 'LineStyle', '--', 'Color', 'k');
set(gca, 'FontSize', 12); grid on;title('Theta3');
legend('Desired','Control')
```

Simulation+Control

```
ind_plot = 1;
if(ind_plot)

figure;
set(gcf, 'Position', [500 300 2000 1200]/2);
grid on;
axis(2*[-1 1 -1 1 -1 1]);


for ii = 1:100:length(theta1_fb)
    plot3(Xd,Yd,Zd,'linewidth', 3, 'Color', 'b');
    hold all;
    plot3(X_fb, Y_fb,Z_fb, 'LineWidth', 2, 'LineStyle', '-.', 'Color', 'k');
    %link 1
     plot3([0 -params.l1*cos(theta1_fb(ii))*sin(theta3_fb(ii))],...
          [0 params.l1*cos(theta1_fb(ii))*cos(theta3_fb(ii))],...
          [0 params.l1*sin(theta1_fb(ii))],...
           'linewidth', 2, 'Color', 'r');
    grid on;
    xlabel('X [m]');
```

```
    ylabel('Y [m]');
    zlabel('Z [m]');
    set(gca, 'FontSize', 16);
    axis(1.5*[-1 1 -1 1 -1 1]);
    view(-5,40);
    %motor 2
     plot3(-params.l1*cos(theta1_fb(ii))*sin(theta3_fb(ii)),...
           params.l1*cos(theta1_fb(ii))*cos(theta3_fb(ii)),...
           params.l1*sin(theta1_fb(ii)),...
           'lineStyle', 'None', 'Marker', 'o', 'MarkerSize', 15, 'MarkerFaceColor', 'r',
'MarkerEdgeColor', 'r');
    %link 2
    plot3([-params.l1*cos(theta1_fb(ii))*sin(theta3_fb(ii)) -
(params.l1*cos(theta1_fb(ii))+params.l2*cos(theta1_fb(ii)+theta2_fb(ii)))*sin(theta3_fb(ii))],...
          [params.l1*cos(theta1_fb(ii))*cos(theta3_fb(ii))
(params.l1*cos(theta1_fb(ii))+params.l2*cos(theta1_fb(ii)+theta2_fb(ii)))*cos(theta3_fb(ii))],...
          [params.l1*sin(theta1_fb(ii))
params.l1*sin(theta1_fb(ii))+params.l2*sin(theta1_fb(ii)+theta2_fb(ii))],...
        'linewidth', 2, 'Color', 'g');
    %End effector
    plot3(-
(params.l1*cos(theta1_fb(ii))+params.l2*cos(theta1_fb(ii)+theta2_fb(ii)))*sin(theta3_fb(ii)),...

(params.l1*cos(theta1_fb(ii))+params.l2*cos(theta1_fb(ii)+theta2_fb(ii)))*cos(theta3_fb(ii)),...
           params.l1*sin(theta1_fb(ii))+params.l2*sin(theta1_fb(ii)+theta2_fb(ii)),...
           'lineStyle', 'None', 'Marker', 'o', 'MarkerSize', 15, 'MarkerFaceColor','g',
'MarkerEdgeColor', 'g');
    pause( 0.1 );
    if (ii~= length(theta1_fb))
        clf;
    else
        % Do nothing, Do not clear figure if not final post
    end

end

end % if(ind_plot)
```

Simmechanic Control

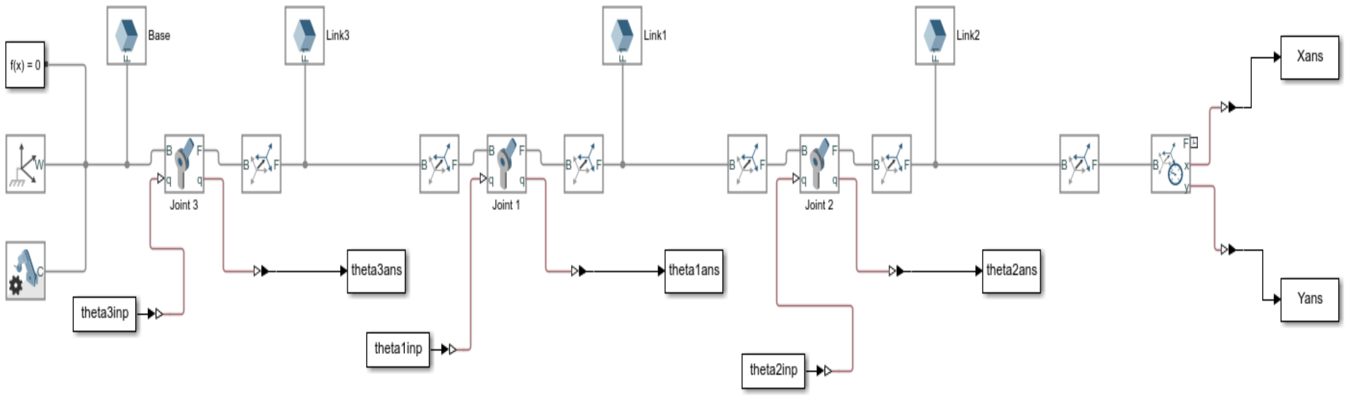
```
theta1inp = [tspan theta1_fb];
theta2inp = [tspan theta2_fb];
theta3inp = [tspan theta3_fb];
sim('Scalar3DOF_control_final');
```

```
function X_dot = advdyfeedback(t, X, tspan, params, Xd, Yd,
Kinematics_Profiles)

% State ------------------------------------------------------------------

theta1       = X(1);
```

```
theta1_dot  = X(2);
theta2      = X(3);
theta2_dot  = X(4);
theta3      = X(5);
theta3_dot  = X(6);


% Parameters ------------------------------------------------------------
params;
l1          = params.l1;            % Length of link 1 [m]
l2          = params.l2;            % Length of link 2 [m]
l3          = params.l3;            % Length of link 3 [m]
lcm1        = params.lcm1;          % CM position of link 1 [m]
lcm2        = params.lcm2;          % CM position of link 2 [m]
m1          = params.m1;            % Mass of link 1 [kg]
m2          = params.m2;            % Mass of link 2 [kg]
m3          = params.m3;            % Mass of link 3 [kg]
Icm1        = params.I_lcm1;        % Moment of inertia of link 1 [kg.m^2]
Icm2        = params.I_lcm2;        % Moment of inertia of link 2 [kg.m^2]
Icm3        = params.I_lcm3;        % Moment of inertia of link 2 [kg.m^2]
kr1         = params.kr1;           % Gear ratio of motor at joint 1 [-]
kr2         = params.kr2;           % Gear ratio of motor at joint 2 [-]
kr3         = params.kr3;           % Gear ratio of motor at joint 2 [-]
M1          = params.m_motor1;      % Mass of rotor at joint 1 [kg]
M2          = params.m_motor2;      % Mass of rotor at joint 2 [kg]
M3          = params.m_motor3;      % Mass of rotor at joint 2 [kg]
I_M1        = params.I_motor1;      % Moment of inertia of rotor at joing 1
[kg.m^2]
I_M2        = params.I_motor2;      % Moment of inertia of rotor at joing 2
[kg.m^2]
I_M3        = params.I_motor3;      % Moment of inertia of rotor at joing 2
[kg.m^2]
g           = params.g;             % Gravity [m/s^2]
Io1         = Icm1+m1*lcm1^2;     % Moment of inertia of link 1 around origin
[kg.m^2]

%% Kinematics Profiles

theta1_d      = Kinematics_Profiles.theta1_d;
theta1_d_dot  = Kinematics_Profiles.theta1_d_dot;
theta1_d_ddot = Kinematics_Profiles.theta1_d_ddot;

theta2_d      = Kinematics_Profiles.theta2_d;
theta2_d_dot  = Kinematics_Profiles.theta2_d_dot;
theta2_d_ddot = Kinematics_Profiles.theta2_d_ddot;

theta3_d      = Kinematics_Profiles.theta3_d;
theta3_d_dot  = Kinematics_Profiles.theta3_d_dot;
theta3_d_ddot = Kinematics_Profiles.theta3_d_ddot;

%% Dynamics

% B(theta)*theta_ddot + C(theta,theta_dot)*theta_dot + Fv*theta_dot +
Fs*sign(theta_dot) + g(theta) = u - J'(theta)*he
```

```
b11 = 32.5+m2*l1^2;
b12 = 10+m2*l2*lcm2*cos(theta2);
b13 = 0;
b21 = b12;
b22 = 10+m2*lcm2^2;
b23 = 0;
b31 = 0;
b32 = 0;
b33 = 10+10*cos(theta1+theta2)^2+cos(theta1)^2*(22.5+m2*l1^2);

c11 = 0;
c12 = -m2*l2*lcm2*sin(theta2)*theta2_dot;
c13 = (sin(theta1)*cos(theta1)*22.5+m2*l1^2+10*sin(theta1+theta2)*...
    cos(theta1+theta2))*theta3_dot;
c21 = 2*m2*l1*lcm2*sin(theta2)*theta2_dot;
c22 = 0;
c23 = 10*sin(theta1+theta2)*cos(theta1+theta2)*theta3_dot;
c31 = 0;
c32 = 0;
c33 = -2*cos(theta1)*sin(theta1)*theta1_dot*(22.5+m2*l1^2)-
20*sin(theta1+theta2)...
    *cos(theta1+theta2)*(theta1_dot+theta2_dot);

g11 = m1*g*lcm1*cos(theta1)+m1*g*l1*cos(theta1)+m2*g*lcm2*cos(theta1+theta2);
g21 = m2*g*lcm2*cos(theta1+theta2);
g31 = 0;

B_theta = [b11 b12 b13;
           b21 b22 b23;
           b31 b32 b33];

C_theta_theta_dot = [c11 c12 c13;
                     c21 c22 c23;
                     c31 c32 c33];

G_theta = [g11;g21;g31];


%% Control Law
u = zeros(3,1);

% B(theta)*theta_ddot + C(theta,theta_dot)*theta_dot + Fv*theta_dot +
Fs*sign(theta_dot) + g(theta) = u - J'(theta)*he
% B(theta)*theta_ddot + C(theta,theta_dot)*theta_dot + 0            + 0
+ g(theta) = u - 0

theta1_d_t      = interp1(tspan, theta1_d,      t);
theta1_d_dot_t  = interp1(tspan, theta1_d_dot,  t);
theta1_d_ddot_t = interp1(tspan, theta1_d_ddot, t);

theta2_d_t      = interp1(tspan, theta2_d,      t);
theta2_d_dot_t  = interp1(tspan, theta2_d_dot,  t);
theta2_d_ddot_t = interp1(tspan, theta2_d_ddot, t);

theta3_d_t      = interp1(tspan, theta3_d,      t);
```

```
theta3_d_dot_t  = interp1(tspan, theta3_d_dot,  t);
theta3_d_ddot_t = interp1(tspan, theta3_d_ddot, t);

Kp = diag([100  100 100]);
Kd = diag([10   10 10]);

% r = [Theta_d_ddot] + Kd*[Theta_d_dot] + Kp*[Theta_d]
r =    [theta1_d_ddot_t;theta2_d_ddot_t;theta3_d_ddot_t] + ...
    Kd*[theta1_d_dot_t;theta2_d_dot_t;theta3_d_dot_t]  + ...
    Kp*[theta1_d_t;theta2_d_t;theta3_d_t];

% y = -Kp*[Theta] - Kd*[Theta_dot] + r
y = -Kp*[theta1; theta2; theta3] - Kd*[theta1_dot; theta2_dot; theta3_dot] +
r;

% u = B_theta*y + C(theta,theta_dot)*theta_dot + g(theta)
Theta_DOT = [theta1_dot; theta2_dot; theta3_dot];
u = B_theta*y + C_theta_theta_dot*Theta_DOT + G_theta;


%% Dynamics

Theta_DDOT = pinv(B_theta) * [ u - C_theta_theta_dot*Theta_DOT - G_theta ];


%% Extract State Information

X_dot(1,1) = theta1_dot;
X_dot(2,1) = Theta_DDOT(1);
X_dot(3,1) = theta2_dot;
X_dot(4,1) = Theta_DDOT(2);
X_dot(5,1) = theta3_dot;
X_dot(6,1) = Theta_DDOT(3);


end
```



Simulink model 'Scalar3DOF_control_final’