Mechatronic System Laboratory

A project report on,

"Manipulator Robot – Pick and Place Task"

Winter semester 2020/21

Date: 30th Jan 2021

Submitted By	**Matriculation Number**
Rajababu Udainarayan Singh	1530580
Suraj Bhardwaj	1531066
Amit Kumar Gupta	1537224
Ravindra Rankawat	1531011
Kapadia Dharmeshbhai Manubhai	1522606

In Guidance of,

Lecturer: Prof. Dr.–Ing. Hubert Roth

Instructor: Nasser Gyagenda

Lab assistant: Peter Sahm

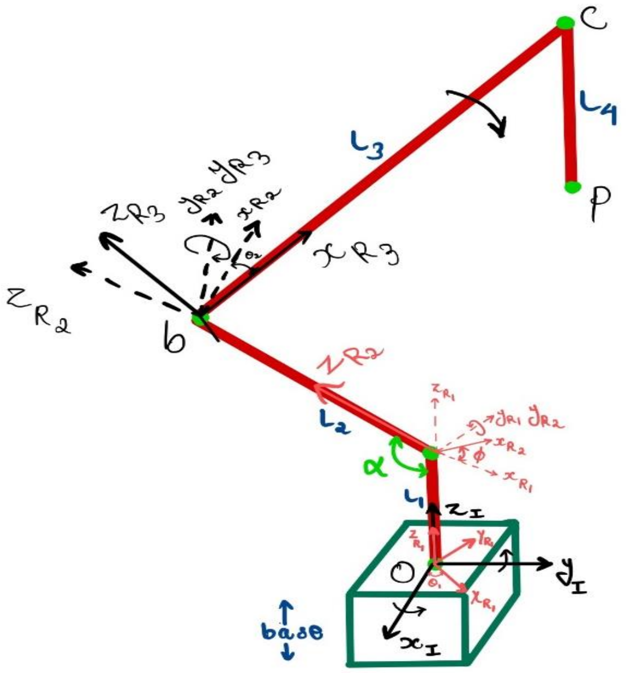
Contents

List of figures

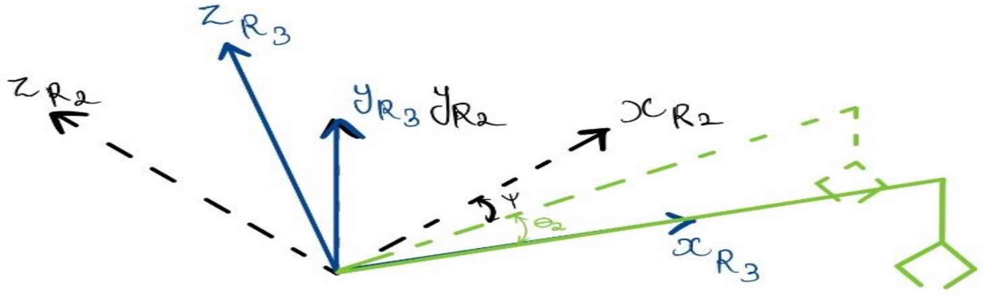
Objectives:

1. To develop the inverse kinematic model (equations) for this manipulator robot
2. To develop behaviours for picking, placing and homing
3. Pick the ball from station C and place it at station A
4. Pick the ball from station A and place it at station B
5. Pick the ball from station B and place it at station C

Task 1: Inverse Kinematics



1. LEGO arm Free Body Diagram



2.Link 3 Free Body Diagram

$$
\begin{aligned}
{}_{I}r_{op} &= \underline{\underline{A}}_{IR1}\,.\,r_{oc_{R1}} + {}_{I}r_{cp} \\
&= \underline{\underline{A}}_{IR1}\,.\,[{}_{R1}r_{oa} + \underline{\underline{A}}_{R1R2}] + {}_{I}r_{cp} \\
&= \underline{\underline{A}}_{IR1}\,.\,[{}_{R1}r_{oa} + \underline{\underline{A}}_{R1R2}\,.\,[{}_{R2}r_{ab} + {}_{R2}r_{bc}]] + {}_{I}r_{cp} \\
&= [\underline{\underline{A}}_{IR1}\,.\,[{}_{R1}r_{oa} + \underline{\underline{A}}_{IR1R2}\,.\,[{}_{R2}r_{ab} + \underline{\underline{A}}_{R2R3}\ {}_{R3}r_{bc}]]\ ] + [{}_{I}r_{cp}\ ]
\end{aligned}
$$

$$
= \left[\begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \left\{ \begin{bmatrix} 0 \\ 0 \\ L_1 \end{bmatrix} + \begin{bmatrix} \cos\emptyset & 0 & -\sin\emptyset \\ 0 & 1 & 0 \\ \sin\emptyset & 0 & \cos\emptyset \end{bmatrix} \left[\begin{bmatrix} 0 \\ 0 \\ L_2 \end{bmatrix} + \begin{bmatrix} \cos\theta'_2 & 0 & \sin\theta'_2 \\ 0 & 1 & 0 \\ -\sin\theta'_2 & 0 & \cos\theta'_2 \end{bmatrix} \begin{bmatrix} L_3 \\ 0 \\ 0 \end{bmatrix} \right] \right\} \right] + \begin{bmatrix} 0 \\ 0 \\ -L_4 \end{bmatrix}
$$

$$
I^{R_{op}} = \begin{bmatrix} L_3\cos\emptyset\ \cos\theta'_2\ \ \cos\theta_1 - L_2\sin\emptyset\cos\theta_1 + L_3\sin\emptyset\sin\theta'_2\ \cos\theta_1 \\ L_3\cos\emptyset\ \cos\theta'_2\ \sin\theta_1 - L_2\sin\emptyset\sin\theta_1 + L_3\sin\emptyset\sin\theta'_2\ \sin\theta_1 \\ L_1 + L_3\cos\theta'_2\sin\emptyset + L_2\cos\emptyset - L_3\cos\emptyset\sin\theta'_2 - L_4 \end{bmatrix}
$$

$$x = \cos\theta_1\ .[L_3\ .\ \cos(\theta'_2 - \emptyset) - L_2\sin\emptyset]$$

$$y = \sin\theta_1\ .[L_3\ .\ \cos(\theta'_2 - \emptyset) - L_2\sin\emptyset]$$

$$y = \sin\theta_1\ .\frac{x}{\cos\theta_1}$$

$$y = \tan\theta_1\ .x$$

$\boldsymbol{\theta_1} = \boldsymbol{tan^{-1}(y/x)}$***..motor C eqn***

$$Z = L_1 + L_2\cos\emptyset + L_3\sin(\emptyset - \theta'_2) - L_4$$

$$\sin(\emptyset - \theta'_2) = \frac{Z - L_1 - L_2\cos\emptyset + L_4}{L_3}$$

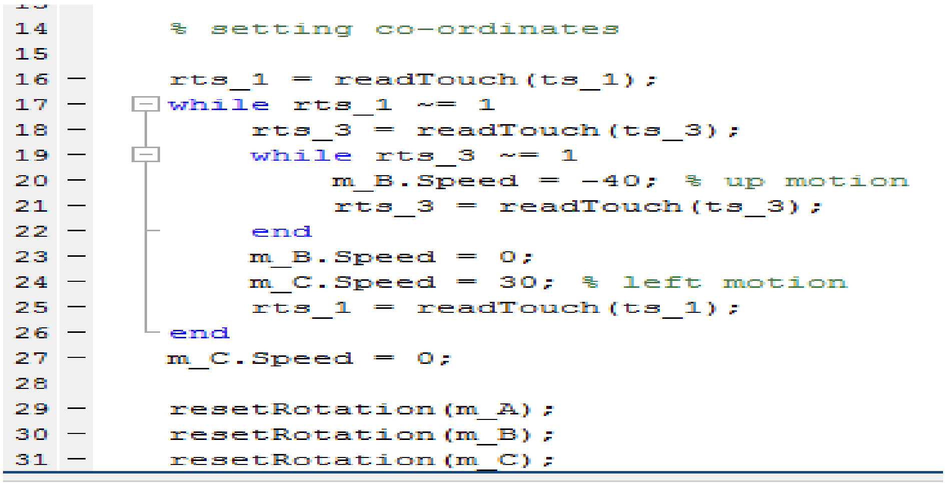
$$\theta'_2 = \emptyset - \sin^{-1}\left[\frac{Z - L_1 - L_2\cos\emptyset + L_4}{L_3}\right]$$

$$\theta'_2 = \varphi + \theta_2$$

$\boldsymbol{\theta_2} = \emptyset - \boldsymbol{\varphi} - \boldsymbol{sin}^{-1}\left[\frac{\boldsymbol{Z - L_1 - L_2\ cos\,\emptyset + L_4}}{\boldsymbol{L_3}}\right]$ ***...mototr B eqn***

Task 2,3,4,5: Picking, Placing & Homing

Step 1: Homing (Setting coordinates)



```
    % setting co-ordinates

    rts_1 = readTouch(ts_1);
    while rts_1 ~= 1
        rts_3 = readTouch(ts_3);
        while rts_3 ~= 1
            m_B.Speed = -40; % up motion
            rts_3 = readTouch(ts_3);
        end
        m_B.Speed = 0;
        m_C.Speed = 30; % left motion
        rts_1 = readTouch(ts_1);
    end
    m_C.Speed = 0;

    resetRotation(m_A);
    resetRotation(m_B);
    resetRotation(m_C);
```

3.Code Snippet for setting coordinates

Step 2: Development of PID controller

$u_k = u_{k-1} + \Delta u_k$

$u_k = u_{k-1} + (K_P + \frac{K_D}{T})\, e_k + (\, K_I\, T - \; K_P - 2\frac{K_D}{T})\, e_{k-1} \; + \frac{K_D}{T}\, e_{k-2}$

$C_0 = (K_P + \frac{K_D}{T})$

$C_1 = (\, K_I\, T - \; K_P - 2\frac{K_D}{T})$

$C_2 = \frac{K_D}{T}$

$u_k = u_{k-1} + C_0\, e_k \; + C_1\, e_{k-1} \; + C_2\, e_{k-2}$

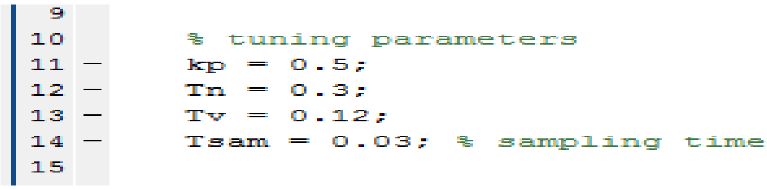
Where,

u_k = current Speed

u_{k-1} = past Speed

Δu_k = Change in Speed

Tuning parameters,

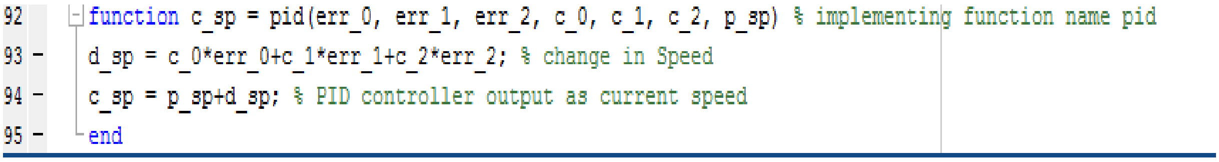


```

    % tuning parameters
    kp = 0.5;
    Tn = 0.3;
    Tv = 0.12;
    Tsam = 0.03; % sampling time

```

4.Code snippet for Tuning Parameters

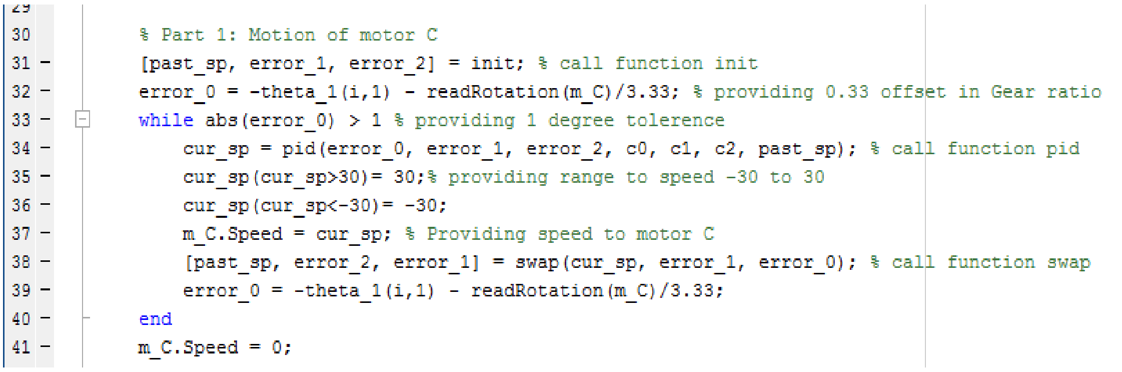


```
function c_sp = pid(err_0, err_1, err_2, c_0, c_1, c_2, p_sp) % implementing function name pid
d_sp = c_0*err_0+c_1*err_1+c_2*err_2; % change in Speed
c_sp = p_sp+d_sp; % PID controller output as current speed
end
```

5. Code snippet for PID

Step 3: Part 1: Motor C motion

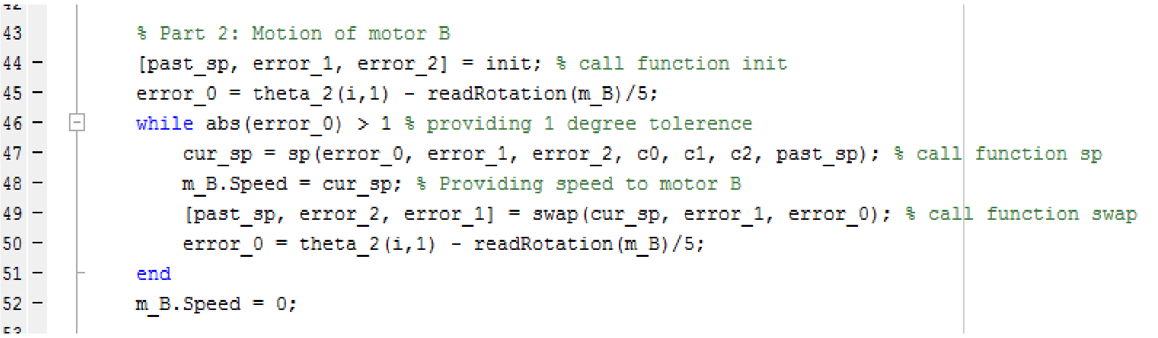
Giving 0.33 offset in Gear Ratio.



```
% Part 1: Motion of motor C
[past_sp, error_1, error_2] = init; % call function init
error_0 = -theta_1(i,1) - readRotation(m_C)/3.33; % providing 0.33 offset in Gear ratio
while abs(error_0) > 1 % providing 1 degree tolerence
    cur_sp = pid(error_0, error_1, error_2, c0, c1, c2, past_sp); % call function pid
    cur_sp(cur_sp>30)= 30;% providing range to speed -30 to 30
    cur_sp(cur_sp<-30)= -30;
    m_C.Speed = cur_sp; % Providing speed to motor C
    [past_sp, error_2, error_1] = swap(cur_sp, error_1, error_0); % call function swap
    error_0 = -theta_1(i,1) - readRotation(m_C)/3.33;
end
m_C.Speed = 0;
```

6. Code snippet for Motor C

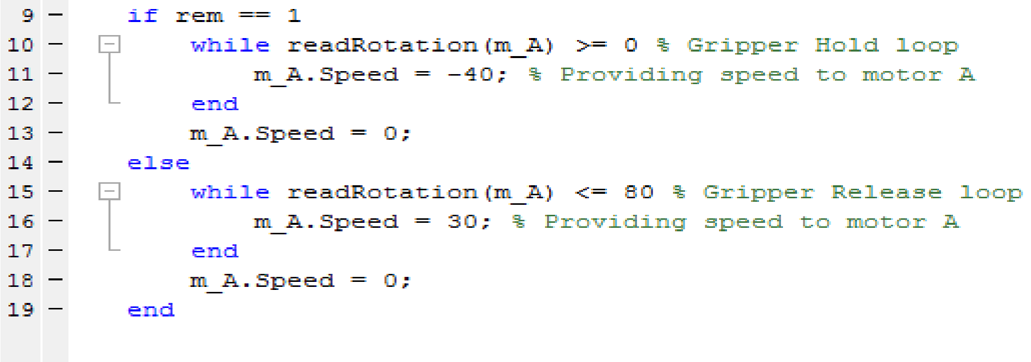
Step 4: Part 2: Motor B motion



```
% Part 2: Motion of motor B
[past_sp, error_1, error_2] = init; % call function init
error_0 = theta_2(i,1) - readRotation(m_B)/5;
while abs(error_0) > 1 % providing 1 degree tolerence
    cur_sp = sp(error_0, error_1, error_2, c0, c1, c2, past_sp); % call function sp
    m_B.Speed = cur_sp; % Providing speed to motor B
    [past_sp, error_2, error_1] = swap(cur_sp, error_1, error_0); % call function swap
    error_0 = theta_2(i,1) - readRotation(m_B)/5;
end
m_B.Speed = 0;
```

7. Code snippet for Motor B motion

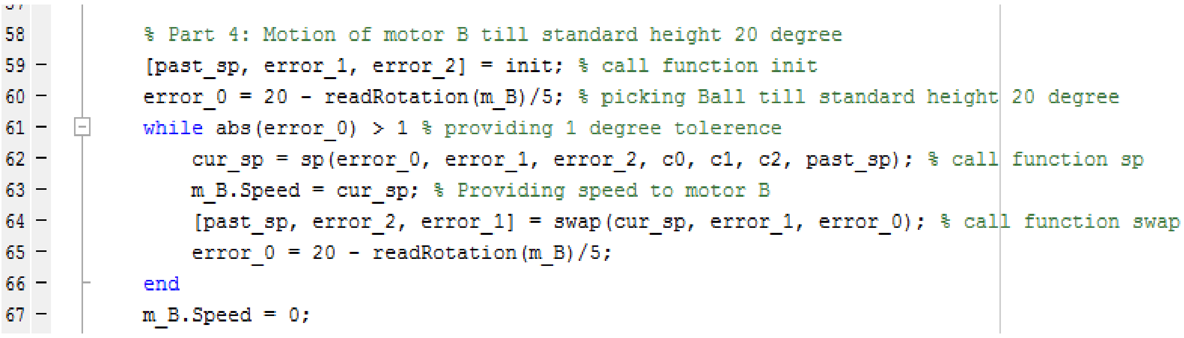
Step 5: Part 3: Gripper



```
if rem == 1
    while readRotation(m_A) >= 0 % Gripper Hold loop
        m_A.Speed = -40; % Providing speed to motor A
    end
    m_A.Speed = 0;
else
    while readRotation(m_A) <= 80 % Gripper Release loop
        m_A.Speed = 30; % Providing speed to motor A
    end
    m_A.Speed = 0;
end
```

8. Code snippet for gripper

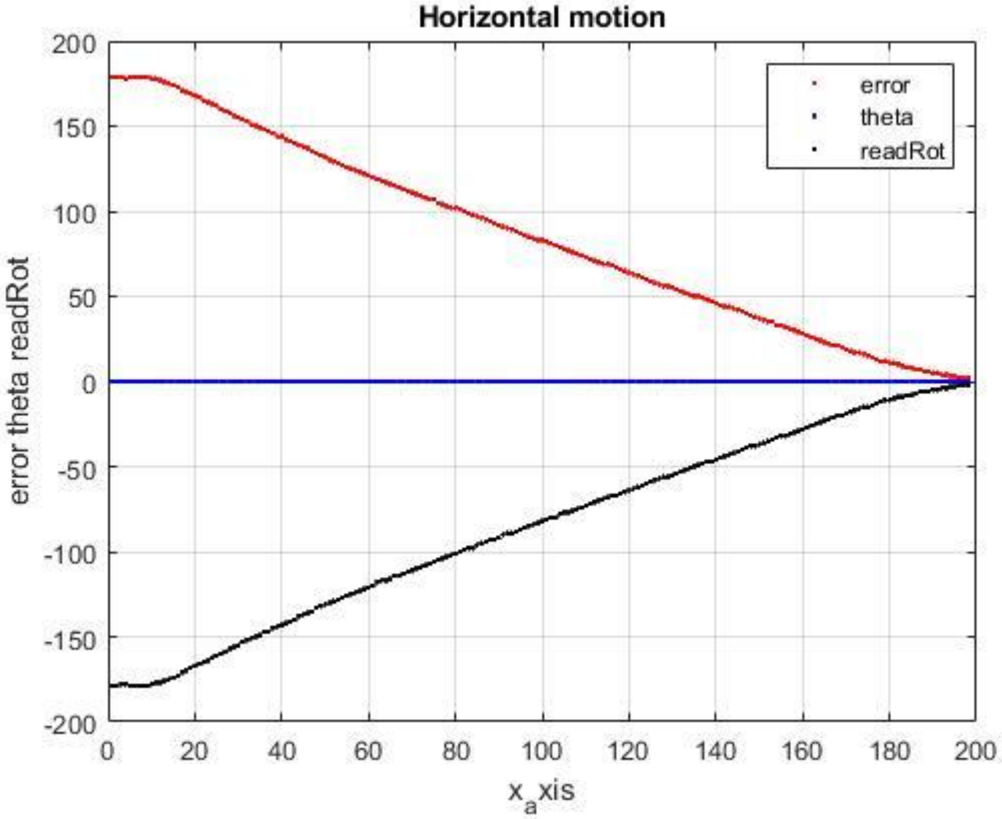
Step 6: Part 4: Standard Position



```
% Part 4: Motion of motor B till standard height 20 degree
[past_sp, error_1, error_2] = init; % call function init
error_0 = 20 - readRotation(m_B)/5; % picking Ball till standard height 20 degree
while abs(error_0) > 1 % providing 1 degree tolerence
    cur_sp = sp(error_0, error_1, error_2, c0, c1, c2, past_sp); % call function sp
    m_B.Speed = cur_sp; % Providing speed to motor B
    [past_sp, error_2, error_1] = swap(cur_sp, error_1, error_0); % call function swap
    error_0 = 20 - readRotation(m_B)/5;
end
m_B.Speed = 0;
```

9. Code snippet for Standard Height

Result:



9. Result

References:

https://de.mathworks.com/help/supportpkg/legomindstormsev3io/