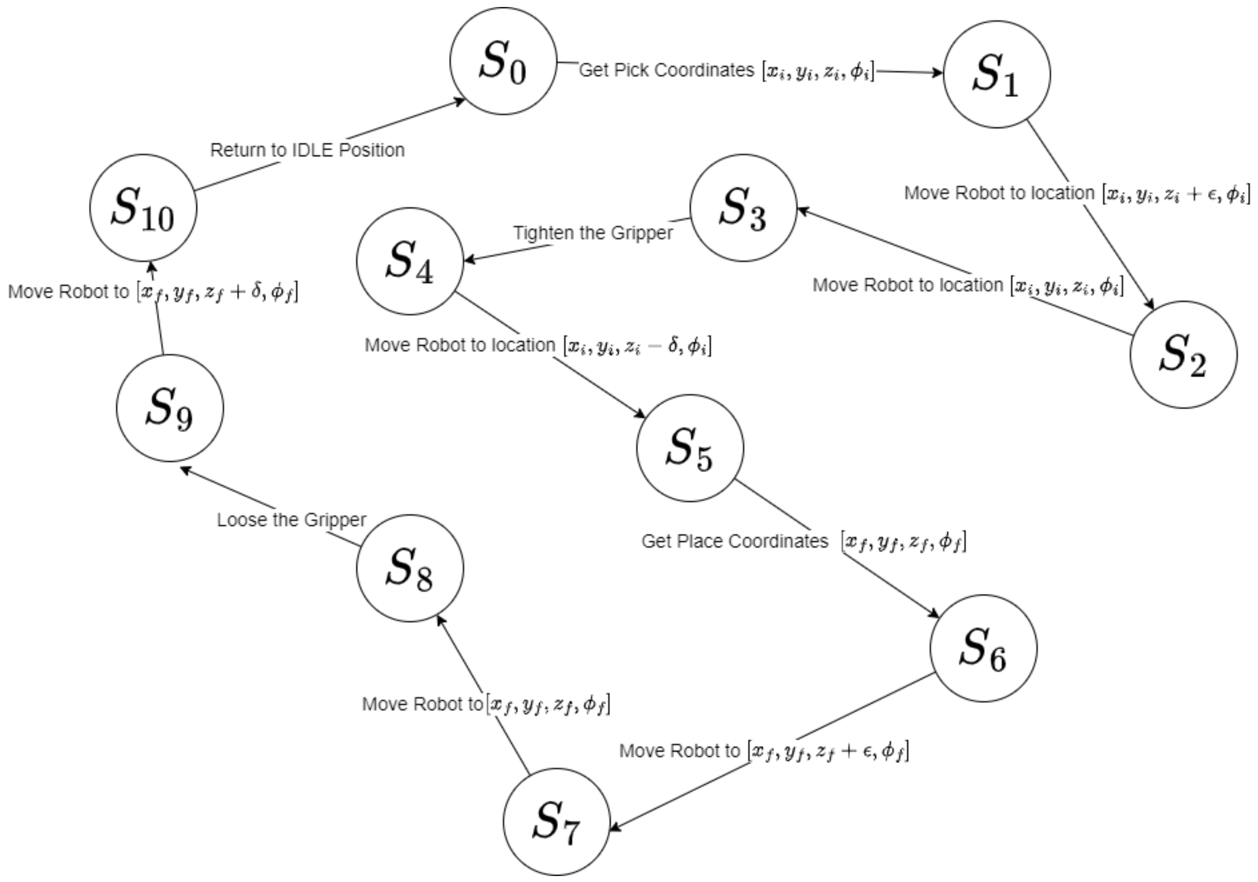
Introduction to Robotics
Lab 6 - A complete Motion Control System

Names: Hamad Abdul Razzaq & Muhammad Ali Siddiqui
IDs: hr06899 & ms06875

Lab Report

Task 6.1

Our FSM is demonstrated by the following State Diagram:



S_0
Get Pick Coordinates $[x_i, y_i, z_i, \phi_i]$
S_1
Move Robot to location $[x_i, y_i, z_i + \epsilon, \phi_i]$
S_2
Move Robot to location $[x_i, y_i, z_i, \phi_i]$
S_3
Tighten the Gripper
S_4
Move Robot to location $[x_i, y_i, z_i - \delta, \phi_i]$
S_5
Get Place Coordinates $[x_f, y_f, z_f, \phi_f]$
S_6
Move Robot to $[x_f, y_f, z_f + \epsilon, \phi_f]$
S_7
Move Robot to $[x_f, y_f, z_f, \phi_f]$
S_8
Loose the Gripper
S_9
Move Robot to $[x_f, y_f, z_f + \delta, \phi_f]$
S_{10}
Return to IDLE Position

Task 6.2

Code

```
State = 0;
global arb;
arb = Arbotix('port', 'COM4', 'nservos', 5);
temp = arb.getpos();
arb.setpos(temp, [64 64 64 64 64]);
motor_speeds = [64 64 64 64 64];
pause(2);
global grip_val;
grip_val = 0;
while 1
    if (State == 0)
        setPosition([3*pi/4 3*pi/4 pi/4 pi/4], grip_val);
        pause(3);
    x = 0; y = -21.6; z = -2; phi = -pi/2 ;
    Pick = [x y z phi];
    disp('Pick Position:');
    disp(Pick);
    State = 1;
elseif (State == 1)
    disp('State:');
    disp(State);
    Pick(3) = Pick(3) + 3;
    theta_arr = findOptimalSolution(Pick(1), Pick(2), Pick(3), Pick(4));
    setPosition(theta_arr, grip_val);
    pause(2);
    State = 2;
elseif State == 2
    disp('State:');
    disp(State);
    Pick(3) = Pick(3) - 3;
    theta_arr = findOptimalSolution(Pick(1), Pick(2), Pick(3), Pick(4));
    setPosition(theta_arr, grip_val);
    pause(2);
    State = 3;

elseif State == 3
    grip_val = 1.2;
    disp('State:');
    disp(State);
    b = arb.getpos();
    b(5) = grip_val;
    arb.setpos(b, [64 64 64 64 64]);
    pause(3);
    State = 4;
elseif State == 4
    disp('State:');
    disp(State);
    Pick(3) = Pick(3) + 6;
    theta_arr = findOptimalSolution(Pick(1), Pick(2), Pick(3), Pick(4));
    setPosition(theta_arr, grip_val);
    pause(2);
    State = 5;
elseif State == 5
    disp('State:');
    disp(State);
    Place = [-21.6 0 0 -pi/2];
    State = 6;
elseif State == 6
    disp('State:');
    disp(State);
    Place(3) = Place(3) + 3;
    theta_arr = findOptimalSolution(Place(1), Place(2), Place(3), Place(4));
    setPosition(theta_arr, grip_val);
    pause(pause_time);
    State = 7;
elseif State == 7
    disp('State:');
    disp(State);
    Place(3) = Place(3) - 5;
    theta_arr = findOptimalSolution(Place(1), Place(2), Place(3), Place(4));
```

```
        setPosition(theta_arr, 1.2);
        findPicnher(theta_arr);
        pause(2);
        State = 8;
    elseif State == 8
        grip_val = 0;
        disp('State:');
        disp(State);
        b = arb.getpos();
        b(5) = grip_val;
        arb.setpos(b, [64 64 64 64 64]);
        pause(2);
        State = 9;
    elseif State == 9
        disp('State:');
        disp(State);
        Place(3) = Place(3) + 5;
        theta_arr = findOptimalSolution(Place(1), Place(2), Place(3), Place(4));
        setPosition(theta_arr, grip_val);
        pause(1);
        State =10;
    elseif State == 10
        disp('State:');
        disp(State);
        arb.setpos([pi/4 pi/4 pi/4 pi/4 grip_val], [64 64 64 64 64]);
        break
    end

end
```

Explanation

In our designed FSM, there are 11 states in total. Our strategy coould be well explained by explanation of each state, which is done below:

State 0

This is our initial state. At this stage, we set our Robot to our selected IDLE position. The IDLE Position for our robot is:

$$\begin{aligned}\theta_1 &= \tfrac{\pi}{4}\text{ rad}\\ \theta_2 &= \tfrac{\pi}{4}\text{ rad}\\ \theta_3 &= \tfrac{\pi}{4}\text{ rad}\\ \theta_4 &= \tfrac{\pi}{4}\text{ rad}\end{aligned}$$

We also receive the pick coordinates in this state. For the sake of demonstration we pick the point $[x, y, z, \phi] = \left[0, -21.6, -2, -\frac{\pi}{2}\right]$ as our Pick coordinates. Next, we move on to State 1.

State 1

In State 1, our goal is to move our Robot towards the desired position. Instead of moving directly towards the desired position we first move to some small distance $\epsilon = 3$cm above the object along the $z - axis$. So, we find the optimal inverse kinematic solution to the given Position coordinates with the $z - value$ incremented by ϵ. Then, we move our Robot to the desired location. Next, we move to State 2.

State 2

In State 2, we decrease the $z-$value by ϵ so that our gripper can surround the object. Also, the gripper value is already initialized to 0, because of which we are sure that our gripper is already open. Our FSM goes to State 3.

State 3

In State 3, we simply tighten the gripper so that our object is gripped firmly before picking it up. We move to State 4 next.

State 4

In State 4, we move our Robot up by some small distance $\delta = 6$cm so that the object is lifted from the ground. Then we move to State 5.

State 5

In State 5, we simply input the Place coordinates. For the sake of demonstration, we took the Place coordinates to be $[x, y, z, \phi] = [-21.6, 0, 0, -pi/2]$. Now we move to State 6.

State 6

In state 6, we move our robot to the Place position but at a certain height $\epsilon = 3cm$ along the $z - axis$ so that our object doesn't collide with the ground while placing it. From here, we move to State 7

State 7

In State 7, we decrease the $z-$value by ϵ so that our object comes to the desired Place location. We also verify that our object is placed at the right location. We move to State 8.

State 8

In State 8, we open our gripper to place place the object and proceed to State 9.

State 9

In State 9, we move verticall up along the $z - axis$ by some small distance so that the object is not dislocated while moving the arm. We move to the final state then, which is State 10. The System also

State 10

In State 10, We move our Robot to the IDLE Position. Our Pick and Place Scenario is completed at this point.

Improvement

- The code could be generalized by taking any input location for both, Pick and Place
- There could be some constraints imposed in this code, such as to continuously take inpput unless a correct input is received, to make the state transition when user presses a key, etc. for better working and demonstration of Robot.

Task 6.3

The following code generates the required jacobian:

```
syms t
syms('theta_1(t)', 'theta_2(t)', 'theta_3(t)', 'theta_4(t)', 'alpha_1', 'alpha_2', ...
   'alpha_3', 'alpha_4', 'a_1', 'a_2', 'a_3', 'a_4', 'd_1', 'd_2', 'd_3', 'd_4');
syms('T_01', 'T_12', 'T_23', 'T_34', 'T_04')
alpha_1 = pi/2; alpha_2 = 0; alpha_3 = 0; alpha_4 = 0;
d_1 = 5.4; d_2 = 0; d_3 = 0; d_4 = 0;
a_1 = 0; a_2 = 10.8; a_3 = 10.8; a_4 = 7.6;
T_01 = [cos(theta_1), -sin(theta_1)*cos(alpha_1), sin(theta_1)*sin(alpha_1), a_1 * cos(theta_1);
        sin(theta_1), cos(theta_1)*cos(alpha_1), -cos(theta_1)*sin(alpha_1), a_1 * sin(theta_1);
        0, sin(alpha_1), cos(alpha_1), d_1;
        0 0 0 1];
TT_01 = T_01(t);
T_12 = [cos(theta_2), -sin(theta_2)*cos(alpha_2), sin(theta_2)*sin(alpha_2), a_2 * cos(theta_2);
        sin(theta_2), cos(theta_2)*cos(alpha_2), -cos(theta_2)*sin(alpha_2), a_2 * sin(theta_2);
        0, sin(alpha_2), cos(alpha_2), d_2;
        0 0 0 1];
T_02 = T_01*T_12;
TT_02 = T_02(t);
T_23 = [cos(theta_3), -sin(theta_3)*cos(alpha_3), sin(theta_3)*sin(alpha_3), a_3 * cos(theta_3);
        sin(theta_3), cos(theta_3)*cos(alpha_3), -cos(theta_3)*sin(alpha_3), a_3 * sin(theta_3);
        0, sin(alpha_3), cos(alpha_3), d_3;
        0 0 0 1];
T_03 = T_02*T_23;
TT_03 = T_03(t);
T_34 = [cos(theta_4), -sin(theta_4)*cos(alpha_4), sin(theta_4)*sin(alpha_4), a_4 * cos(theta_4);
        sin(theta_4), cos(theta_4)*cos(alpha_4), -cos(theta_4)*sin(alpha_4), a_4 * sin(theta_4);
        0, sin(alpha_4), cos(alpha_4), d_4;
        0 0 0 1];
T_04 = T_03*T_34;
TT_04 = T_04(t);
O_04 = TT_04(1:3,4);
Jv = [diff(O_04, theta_1) diff(O_04, theta_2) diff(O_04, theta_3) diff(O_04, theta_4)];
Jw = [[0;0;1] TT_01(1:3,3) TT_02(1:3,3) TT_03(1:3,3)];
J = [Jv ; Jw]
```

The obtained expression for jacobian is:

$$J[1] = \begin{bmatrix} 10.8\sin(\theta_3(t))\,\sigma_{16} - 10.8\cos(\theta_2(t))\sin(\theta_1(t)) - 10.8\cos(\theta_3(t))\,\sigma_{17} - 7.6\cos(\theta_4(t))\,\sigma_{14} + 7.6\sin(\theta_4(t))\,\sigma_{15} \\ 3.8\cos(\sigma_3) + 5.4\cos(\sigma_7) + 5.4\cos(\sigma_9) + 3.8\cos(\sigma_2) + 5.4\cos(\sigma_6) + 5.4\cos(\sigma_8) \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$J[2] = \begin{bmatrix} 5.4\sin(\sigma_8) - 3.8\sin(\sigma_3) - 5.4\sin(\sigma_7) - 5.4\sin(\sigma_9) - 3.8\sin(\sigma_2) - 5.4\sin(\sigma_6) \\ -\sigma_{13} - \sigma_{12} - \sigma_{11} - \sigma_{10} - 10.8\sin(\theta_1(t))\sin(\theta_2(t)) \\ 10.8\cos(\theta_2(t)) + \sigma_1 + \sigma_5 \\ \sin(\theta_1(t)) \\ \sigma_4 \\ 0 \end{bmatrix}$$

$$J[3] = \begin{bmatrix} -3.8\sin(\sigma_3) - 5.4\sin(\sigma_7) - 3.8\sin(\sigma_2) - 5.4\sin(\sigma_6) \\ -\sigma_{13} - \sigma_{12} - \sigma_{11} - \sigma_{10} \\ \sigma_1 + \sigma_5 \\ \sin(\theta_1(t)) \\ \sigma_4 \\ 0 \end{bmatrix}$$

$$J[4] = \begin{bmatrix} -3.8\sin(\sigma_3) - 3.8\sin(\sigma_2) \\ -\sigma_{11} - \sigma_{10} \\ \sigma_1 \\ \sin(\theta_1(t)) \\ \sigma_4 \\ 0 \end{bmatrix}$$

Here, $J[i]$ correspinds to i^{th} column of J

where

$$\sigma_1 = 7.6\cos\left(\theta_2(t) + \theta_3(t) + \theta_4(t)\right)$$

$$\sigma_2 = \theta_2(t) - \theta_1(t) + \theta_3(t) + \theta_4(t)$$

$$\sigma_3 = \theta_1(t) + \theta_2(t) + \theta_3(t) + \theta_4(t)$$

$$\sigma_4 = -\cos\left(\theta_1(t)\right)$$

$$\sigma_5 = 10.8\cos\left(\theta_2(t) + \theta_3(t)\right)$$

$$\sigma_6 = \theta_2(t) - \theta_1(t) + \theta_3(t)$$

$$\sigma_7 = \theta_1(t) + \theta_2(t) + \theta_3(t)$$

$$\sigma_8 = \theta_1(t) - \theta_2(t)$$

$$\sigma_9 = \theta_1(t) + \theta_2(t)$$

$$\sigma_{10} = 7.6\sin\left(\theta_4(t)\right)\sigma_{14}$$

$$\sigma_{11} = 7.6\cos\left(\theta_4(t)\right)\sigma_{15}$$

$$\sigma_{12} = 10.8\sin\left(\theta_3(t)\right)\sigma_{17}$$

$$\sigma_{13} = 10.8\cos\left(\theta_3(t)\right)\sigma_{16}$$

$$\sigma_{14} = \cos\left(\theta_3(t)\right)\sigma_{17} - \sin\left(\theta_3(t)\right)\sigma_{16}$$

$$\sigma_{15} = \cos\left(\theta_3(t)\right)\sigma_{16} + \sin\left(\theta_3(t)\right)\sigma_{17}$$

$$\sigma_{16} = \sin\left(\theta_1(t)\right)\sin\left(\theta_2(t)\right)$$

$$\sigma_{17} = \cos\left(\theta_2(t)\right)\sin\left(\theta_1(t)\right)$$