Robotics Lab 6

Syed Mustafa

Hazika Farooq

Table of Contents

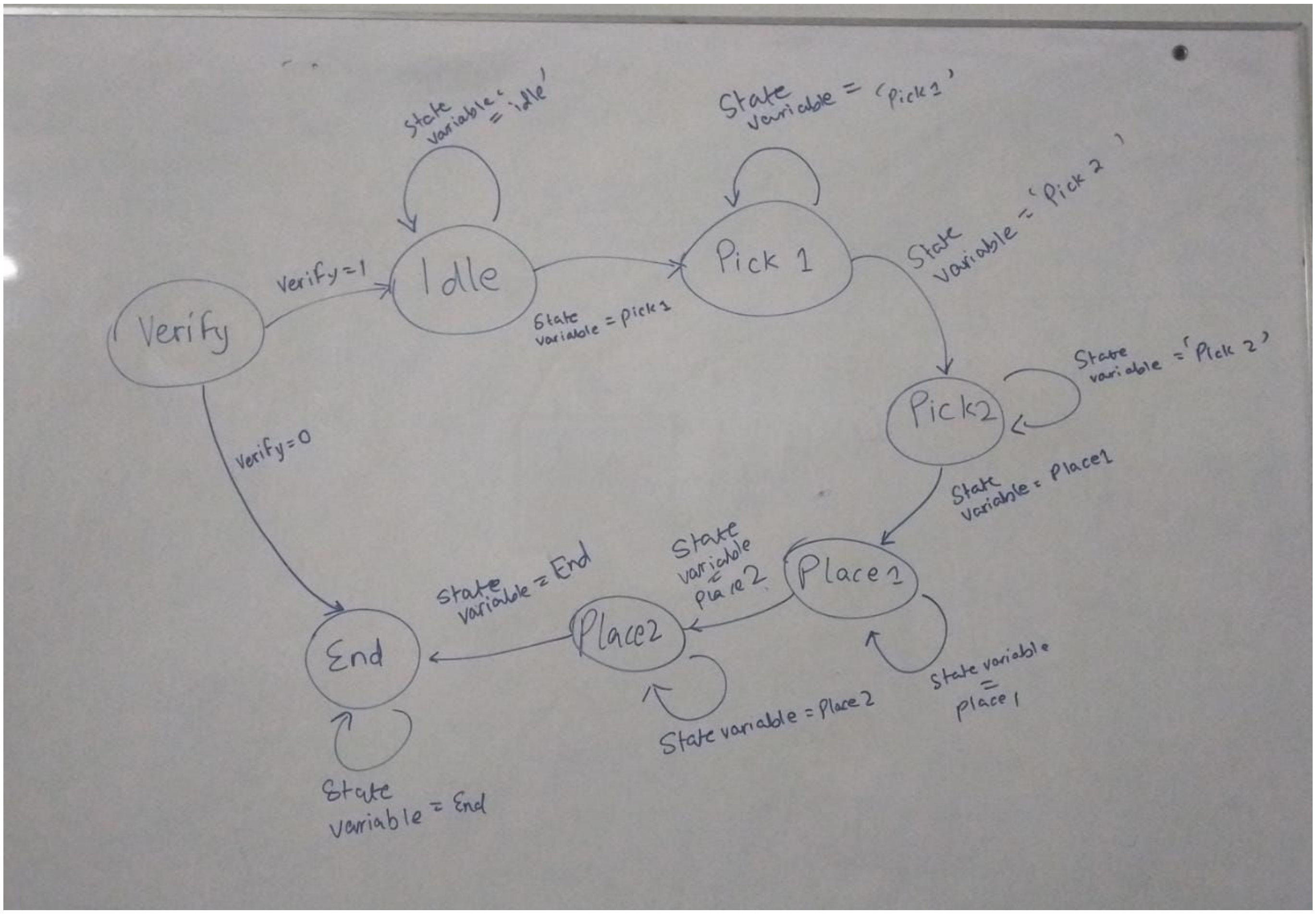
Task 1 FSM

Task 6.1 FSM (30 points)

Draw a state-transition diagram of an FSM corresponding to the following scenario:

- System is in idle state till it receives pick location, (x_1, y_1, z_1, ϕ_1) and place location, (x_2, y_2, z_2, ϕ_2).
- Geometry of the object to be picked and placed, including its orientation, is known before hand.
- The locations can be assumed to lie in the interior of the manipulator's workspace, and the object is in an orientation so that it can be picked.
- System should verify the final placement location, before determining that the task has concluded.
- Smooth motion and accurate placement[a] is desirable.

[a] You'll have to plan your gripper picking and releasing strategy, considering the accuracy of your system, determined in earlier labs.



Explaination of the states

1. Verify
2. Idle
3. Pick 1
4. Pick 2
5. Place 1
6. Place 2
7. End

Verify

In this state the FSM will check if the place coordinates are valid.

Note: An assumption here is that the workspace here is the area where $|\phi|$ = 90. Thus with this constraint the reachable worspace becomes $r < a_2 + a_3 \ni r$ is the distance of the end effector from the origin {0} on $x_0 - y_0$

Idle

The joint coordinates are [0 0 0 0]

Pick 1

The end effector is at z+5 pick coordinateas

Pick 2

The end effector is at pick coordinates

The jaw is closed

Place 1

The end effector is at z+5 place coordinateas

Place 2

The end effector is at place coordinates

Th jaw is opened

End

The object has been placed and arm is back to idle position

Task 2 FSM Implementation

Task 6.2 FSM Implementation (50 points)

Implement the system described by the previous FSM in MATLAB[a] for Phantom X Pincher and the cube object. In addition to your implementation code, submit an explanation of your strategy, especially functions that were not developed previously, a video of your best execution, and identify and comment on points of improvement.

[a] You can implement the FSM using usual text-based programming, or Stateflow, a graphical programming environment for implementing FSMs in MATLAB. To learn further about Staeflow, see https://www.mathworks.com/help/stateflow/gs/finite-state-machines.html.

We have implemented a FSM in a function that would input the pick and place

```
function error = FSM(pick, place)
    %calculating the joint angles

    epsilon = 0.05;
    arb = Arbotix('port','COM5', 'nservos',5);
    %  pick/place = [x, y, z, phi]




    %Initialiing the state variables
    verify = 0;
    stateVariable = 'Idle';
    error = 0; %if error is zero no error has occured

    % Verify the place location
    % Given the constraint, i.e phi = -pi/2 and z = -1;
    a2 = 10.8; a3 = 10.8;
    r = a2 + a3;

    rPlace = sqrt(place(1)^2+place(2)^2);
```

```
if (rPlace <= r)
    verify = 1;
end
while verify
    curr_pos = arb.getpos();
    % Starting from the idle state.
    if str_comp(stateVariable, 'Idle')
        setPosition(arb, idle)
        % Apply a condition to see if the robot is in the idle state,
        % given a tolerance of epsilon to the sum.
        while  ~(sum(curr_pos) <= sum(idle)+ epsilon && sum(curr_pos) >= sum(idle) - epsilon)
            curr_pos = arb.getpos();
            stateVariable = 'Pick1';
        end
    elseif str_comp(stateVariable,'Pick1')
    % In this state the end effector will be at z + 5, z coordinates.
        pick1_angles = findOptimalSolution(pick(1), pick(2), pick(3) + 5, pick(4));
        setPosition(arb, pick1_angles);
        while ~(sum(curr_pos) <= sum(pick1_angles)+ epsilon && sum(curr_pos) >= sum(pick1_angles) - epsilo
            curr_pos = arb.getpos();
            %Waiting condition
        end
        stateVariable = 'Pick2';
    elseif str_comp(stateVariable,'Pick2')
        pick2_angles = findOptimalSolution(pick(1), pick(2), pick(3), pick(4));
        setPosition(arb, pick2_angles);
        arb.setpos(5, 1.25,64);
        while ~ (sum(curr_pos) <= sum([pick2_angles 1.25])+ epsilon && sum(curr_pos) >= sum([pick2_angles
            curr_pos = arb.getpos();
            % Waiting Condition
        end
        stateVariable = 'Place1';
    elseif str_comp(stateVariable, 'Place1')
        place1_angles = findOptimalSolution(place(1), place(2), place(3) + 5, place(4));
        setPosition(arb, place1_angles)
        while ~(sum(curr_pos) <= sum(place1_angles)+ epsilon && sum(curr_pos) >= sum(place1_angles) - epsi
            curr_pos = arb.getpos();
            %waiting_condition
        end
        stateVariable = 'Place2';
    elseif str_comp(stateVariable, 'Place2')
        place2_angles = findOptimalSolution(place(1), place(2), place(3), place(4));
        setPosition(arb, place2_angles)
        arb.setpos(5,0,64)
        while ~ (sum(curr_pos) <= sum(place2_angles)+ epsilon && sum(curr_pos) >= sum(place2_angles) - eps
            curr_pos = arb.getpos();
        end
        stateVariable = 'End';
    elseif str_comp(stateVariable, 'End') %Note this state is al
        setPosition(arb, idle)
        while ~(sum(curr_pos) <= sum(idle)+ epsilon && sum(curr_pos) >= sum(idle) - epsilon)
            curr_pos = arb.getpos();
        end
        verify = 0;


    end
```

```
        end
    end
```

```
Pick = [16 -16 -1 -pi/2];
place = [-10 -16 -1 -pi/2];
FSM(Pick,place);
```

Note: The video link can be found in the comments of the submission

Model Jacobian

Task 6.3 Manipulator Jacobian (20 points)

Use the DH parameters and homogeneous transformation, ${}^0T_4$, obtained in the previous lab to find the Jacobian for the manipulator in the lab.

- For convenience, a MATLAB function `createA(theta,d,a,alpha)` is available on canvas to easily create homogeneous transformations in symbolic form.
- Define your joint variables θ_i as functions of time, so that you can differentiate them.

```
syms theta_1(t) theta_2(t) theta_3(t) theta_4(t)
A1 = createA(theta_1,'d_1',0,-pi/2)
```

- The homogeneous transformation you'll obtain will be a 4×4 matrix function of t. To extract a particular entry of this matrix, you'll have to first evaluate it at a value of t and save it in an intermediate variable. For example, if `B` is a matrix symbolic function and you want to find matrix entry $(1,2)$, then use:

```
tempVar = B(t);
entry = tempVar(1,2);
```

- You can find derivative of a symbolic expression using the MATLAB function `diff`. For example, `diff(f,x)` computes $\frac{\partial f}{\partial x}$.
- Chain rule will frequently yield simplified expressions.

Link	a_i	α_i	d_i	θ_i
1	0	90	14 cm	θ_1
2	10.5 cm	0	0	θ_2
3	10.5cm	0	0	θ_3
4	7.5 cm	0	0	θ_4

Table 1: Joint DH Parameter Table

$$J = \begin{bmatrix} J_v \\ J_w \end{bmatrix} = \begin{bmatrix} \text{Cross}\left({}^0z_{i-1}, \left({}^0p_n - {}^0p_{i-1}\right)\right) \\ {}^0z_{i-1} \end{bmatrix}$$

```
syms('theta_1(t)','theta_2(t)','theta_3(t)','theta_4(t)')
% createA(theta,d,a,alpha)
A1 = createA(theta_1, 14, 0, 90);
A2 = createA(theta_2,0, 10.5,0);
A3 = createA(theta_3, 0, 10.5, 0);
A4 = createA(theta_4, 0, 7.5, 0);
T1 = A1;
T1 = T1(t)
```

T1 =

$$\begin{pmatrix} \cos(\theta_1(t)) & -\sin(\theta_1(t))\cos(90) & \sin(\theta_1(t))\sin(90) & 0 \\ \sin(\theta_1(t)) & \cos(\theta_1(t))\cos(90) & -\cos(\theta_1(t))\sin(90) & 0 \\ 0 & \sin(90) & \cos(90) & 14 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
T2 = A1*A2;
T2 = T2(t)
```

T2 =

$$\left(\begin{matrix} \cos(\theta_1(t))\cos(\theta_2(t)) - \sin(\theta_1(t))\sin(\theta_2(t))\cos(90) & -\cos(\theta_1(t))\sin(\theta_2(t)) - \cos(\theta_2(t))\sin(\theta_1(t))\cos(90) \\ \cos(\theta_2(t))\sin(\theta_1(t)) + \cos(\theta_1(t))\sin(\theta_2(t))\cos(90) & \cos(\theta_1(t))\cos(\theta_2(t))\cos(90) - \sin(\theta_1(t))\sin(\theta_2(t)) \\ \sin(\theta_2(t))\sin(90) & \cos(\theta_2(t))\sin(90) \\ 0 & 0 \end{matrix}\right.$$

```
T3 = T2 * A3;
T3 = T3(t)
```

T3 =

$$\left(\begin{array}{cc}
\cos(\theta_3(t))\,\sigma_1 - \sin(\theta_3(t))\,\sigma_4 & -\cos(\theta_3(t))\,\sigma_4 - \sin(\theta_3(t))\,\sigma_1 \\
\cos(\theta_3(t))\,\sigma_3 - \sin(\theta_3(t))\,\sigma_2 & -\cos(\theta_3(t))\,\sigma_2 - \sin(\theta_3(t))\,\sigma_3 \\
\cos(\theta_2(t))\sin(\theta_3(t))\sin(90) + \cos(\theta_3(t))\sin(\theta_2(t))\sin(90) & \cos(\theta_2(t))\cos(\theta_3(t))\sin(90) - \sin(\theta_2(t))\sin(\theta_3(t \\
0 & 0
\end{array}\right.$$

where

$$\sigma_1 = \cos(\theta_1(t))\cos(\theta_2(t)) - \sin(\theta_1(t))\sin(\theta_2(t))\cos(90)$$

$$\sigma_2 = \sin(\theta_1(t))\sin(\theta_2(t)) - \cos(\theta_1(t))\cos(\theta_2(t))\cos(90)$$

$$\sigma_3 = \cos(\theta_2(t))\sin(\theta_1(t)) + \cos(\theta_1(t))\sin(\theta_2(t))\cos(90)$$

$$\sigma_4 = \cos(\theta_1(t))\sin(\theta_2(t)) + \cos(\theta_2(t))\sin(\theta_1(t))\cos(90)$$

```
T4 = T3* A4;
T4 = T4(t)
```

```
T4 =
```

$$\begin{pmatrix} \cos(\theta_4(t))\,\sigma_3 - \sin(\theta_4(t))\,\sigma_6 & -\cos(\theta_4(t))\,\sigma_6 - \sin(\theta_4(t))\,\sigma_3 & \sin(\theta_1(t))\sin(90) & \frac{15\cos(\theta_4(t))\,\sigma_3}{2} - \frac{15\sin(\cdots}{2} \\ \cos(\theta_4(t))\,\sigma_4 - \sin(\theta_4(t))\,\sigma_5 & -\cos(\theta_4(t))\,\sigma_5 - \sin(\theta_4(t))\,\sigma_4 & -\cos(\theta_1(t))\sin(90) & \frac{21\cos(\theta_2(t))\sin(\theta_1(t))}{2} + \cdots \\ \sin(\theta_4(t))\,\sigma_1 + \cos(\theta_4(t))\,\sigma_2 & \cos(\theta_4(t))\,\sigma_1 - \sin(\theta_4(t))\,\sigma_2 & \cos(90) & \frac{15\sin(\theta_4(t))\,\sigma_1}{2} + \frac{21\sin(\cdots}{} \\ 0 & 0 & 0 & \end{pmatrix}$$

where

$$\sigma_1 = \cos(\theta_2(t))\cos(\theta_3(t))\sin(90) - \sin(\theta_2(t))\sin(\theta_3(t))\sin(90)$$

$$\sigma_2 = \cos(\theta_2(t))\sin(\theta_3(t))\sin(90) + \cos(\theta_3(t))\sin(\theta_2(t))\sin(90)$$

$$\sigma_3 = \cos(\theta_3(t))\,\sigma_9 - \sin(\theta_3(t))\,\sigma_{10}$$

$$\sigma_4 = \cos(\theta_3(t))\,\sigma_8 - \sin(\theta_3(t))\,\sigma_7$$

$$\sigma_5 = \cos(\theta_3(t))\,\sigma_7 + \sin(\theta_3(t))\,\sigma_8$$

$$\sigma_6 = \cos(\theta_3(t))\,\sigma_{10} + \sin(\theta_3(t))\,\sigma_9$$

$$\sigma_7 = \sin(\theta_1(t))\sin(\theta_2(t)) - \cos(\theta_1(t))\cos(\theta_2(t))\cos(90)$$

$$\sigma_8 = \cos(\theta_2(t))\sin(\theta_1(t)) + \cos(\theta_1(t))\sin(\theta_2(t))\cos(90)$$

$$\sigma_9 = \cos(\theta_1(t))\cos(\theta_2(t)) - \sin(\theta_1(t))\sin(\theta_2(t))\cos(90)$$

$$\sigma_{10} = \cos(\theta_1(t))\sin(\theta_2(t)) + \cos(\theta_2(t))\sin(\theta_1(t))\cos(90)$$

```
J = sym(zeros(6, 4))
```

J =

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

```
pn = T4(1:3, 4)
```

```
pn =
```

$$\left(\begin{array}{c} \frac{15\cos(\theta_4(t))\ (\cos(\theta_3(t))\,\sigma_1 - \sin(\theta_3(t))\,\sigma_4)}{2} - \frac{15\sin(\theta_4(t))\ (\cos(\theta_3(t))\,\sigma_4 + \sin \\ \frac{21\cos(\theta_2(t))\sin(\theta_1(t))}{2} + \frac{15\cos(\theta_4(t))\ (\cos(\theta_3(t))\,\sigma_3 - \sin(\theta_3(t))\,\sigma_2)}{2} - \frac{15\sin(\\ \frac{15\sin(\theta_4(t))\ (\cos(\theta_2(t))\cos(\theta_3(t))\sin(90) - \sin(\theta_2(t))\sin(\theta_3(t))\sin(90))}{2} + \frac{21\sin(\theta_2(t))\sin(90)}{2} + \frac{15\cos(\theta_4 \end{array}\right.$$

where

$$\sigma_1 = \cos(\theta_1(t))\cos(\theta_2(t)) - \sin(\theta_1(t))\sin(\theta_2(t))\cos(90)$$

$$\sigma_2 = \sin(\theta_1(t))\sin(\theta_2(t)) - \cos(\theta_1(t))\cos(\theta_2(t))\cos(90)$$

$$\sigma_3 = \cos(\theta_2(t))\sin(\theta_1(t)) + \cos(\theta_1(t))\sin(\theta_2(t))\cos(90)$$

$$\sigma_4 = \cos(\theta_1(t))\sin(\theta_2(t)) + \cos(\theta_2(t))\sin(\theta_1(t))\cos(90)$$

```
J(4:6, :) = [[0;0;1], T1(1:3,3), T2(1:3,3), T3(1:3,3)];
J(1:3,:) = [diff(T4(1:3,4), theta_1), diff(T4(1:3,4), theta_2), diff(T4(1:3,4), theta_3),diff(T
J1 = J(:,1)
```

```
J1 =
```

$$\left(\begin{array}{c} \frac{15\sin(\theta_4(t))\ (\cos(\theta_3(t))\,\sigma_2 + \sin(\theta_3(t))\,\sigma_3)}{2} - \frac{15\cos(\theta_4(t))\ (\cos(\theta_3(t))\,\sigma_3 - \sin(\theta_3(t))\,\sigma_2)}{2} - \frac{21\cos(\theta_2(t))\sin(\\ \frac{15\cos(\theta_4(t))\ (\cos(\theta_3(t))\,\sigma_1 - \sin(\theta_3(t))\,\sigma_4)}{2} - \frac{15\sin(\theta_4(t))\ (\cos(\theta_3(t))\,\sigma_4 + \sin(\theta_3(t))\,\sigma_1)}{2} + \frac{21\cos(\theta_3(t))\,\sigma_1}{2} - \\ 0 \\ 0 \\ 0 \\ 1 \end{array}\right.$$

where

$$\sigma_1 = \cos(\theta_1(t))\cos(\theta_2(t)) - \sin(\theta_1(t))\sin(\theta_2(t))\cos(90)$$

$$\sigma_2 = \sin(\theta_1(t))\sin(\theta_2(t)) - \cos(\theta_1(t))\cos(\theta_2(t))\cos(90)$$

$$\sigma_3 = \cos(\theta_2(t))\sin(\theta_1(t)) + \cos(\theta_1(t))\sin(\theta_2(t))\cos(90)$$

$$\sigma_4 = \cos(\theta_1(t))\sin(\theta_2(t)) + \cos(\theta_2(t))\sin(\theta_1(t))\cos(90)$$

```
J2 = J(:,2)
```

```
J2 =
```

$$\left(\begin{array}{c} -\frac{21\cos(\theta_1(t))\sin(\theta_2(t))}{2}-\frac{15\cos(\theta_4(t))\ (\cos(\theta_3(t))\,\sigma_4+\sin(\theta_3(t))\,\sigma_1)}{2}-\frac{15\sin(\theta \\ \frac{21\cos(\theta_1(t))\cos(\theta_2(t))\cos(90)}{2}-\frac{21\sin(\theta_1(t))\sin(\theta_2(t))}{2}-\frac{15\sin(\theta_4(t))\ (\cos(\theta_3(\\ \frac{21\cos(\theta_2(t))\sin(90)}{2}-\frac{15\sin(\theta_4(t))\ (\cos(\theta_2(t))\sin(\theta_3(t))\sin(90)+\cos(\theta_3(t))\sin(\theta_2(t))\sin(90))}{2}+\frac{15\cos(\theta \\ \\ \\ \end{array}\right.$$

where

$$\sigma_1 = \cos(\theta_1(t))\cos(\theta_2(t)) - \sin(\theta_1(t))\sin(\theta_2(t))\cos(90)$$

$$\sigma_2 = \sin(\theta_1(t))\sin(\theta_2(t)) - \cos(\theta_1(t))\cos(\theta_2(t))\cos(90)$$

$$\sigma_3 = \cos(\theta_2(t))\sin(\theta_1(t)) + \cos(\theta_1(t))\sin(\theta_2(t))\cos(90)$$

$$\sigma_4 = \cos(\theta_1(t))\sin(\theta_2(t)) + \cos(\theta_2(t))\sin(\theta_1(t))\cos(90)$$

```
J3 = J(:,3)
```

```
J3 =
```

$$
\left(\begin{array}{c}
-\frac{15\cos(\theta_4(t))\ (\cos(\theta_3(t))\,\sigma_4+\sin(\theta_3(t))\,\sigma_1)}{2}-\frac{15\sin(\theta_4(t))\ (\mathrm{c}
\\
-\frac{15\cos(\theta_4(t))\ (\cos(\theta_3(t))\,\sigma_2+\sin(\theta_3(t))\,\sigma_3)}{2}-\frac{15\sin(\theta_4(t))\ (\mathrm{c}
\\
\frac{15\cos(\theta_4(t))\ (\cos(\theta_2(t))\cos(\theta_3(t))\sin(90)-\sin(\theta_2(t))\sin(\theta_3(t))\sin(90))}{2}-\frac{15\sin(\theta_4(t))\ (\cos(\theta_2(t))\sin(\theta_3(t
\\
\sin(\theta_1(t)
\\
-\cos(\theta_1(t
\\
\cos(
\end{array}\right.
$$

where

$$\sigma_1 = \cos(\theta_1(t))\cos(\theta_2(t)) - \sin(\theta_1(t))\sin(\theta_2(t))\cos(90)$$

$$\sigma_2 = \sin(\theta_1(t))\sin(\theta_2(t)) - \cos(\theta_1(t))\cos(\theta_2(t))\cos(90)$$

$$\sigma_3 = \cos(\theta_2(t))\sin(\theta_1(t)) + \cos(\theta_1(t))\sin(\theta_2(t))\cos(90)$$

$$\sigma_4 = \cos(\theta_1(t))\sin(\theta_2(t)) + \cos(\theta_2(t))\sin(\theta_1(t))\cos(90)$$

```
J4 = J(:,4)
```

```
J4 =
```

$$
\left(\begin{array}{c}
-\frac{15\cos(\theta_4(t))\ (\cos(\theta_3(t))\,\sigma_4+\sin(\theta_3(t))\,\sigma_1)}{2}-\frac{15\sin(\theta_4(t))\ (\cos(\theta_3(t))\,\sigma_1-\mathrm{si}}{2}
\\
-\frac{15\cos(\theta_4(t))\ (\cos(\theta_3(t))\,\sigma_2+\sin(\theta_3(t))\,\sigma_3)}{2}-\frac{15\sin(\theta_4(t))\ (\cos(\theta_3(t))\,\sigma_3-\mathrm{si}}{2}
\\
\frac{15\cos(\theta_4(t))\ (\cos(\theta_2(t))\cos(\theta_3(t))\sin(90)-\sin(\theta_2(t))\sin(\theta_3(t))\sin(90))}{2}-\frac{15\sin(\theta_4(t))\ (\cos(\theta_2(t))\sin(\theta_3(t
\\
\sin(\theta_1(t))\sin(90)
\\
-\cos(\theta_1(t))\sin(90)
\\
\cos(90)
\end{array}\right.
$$

where

$$\sigma_1 = \cos(\theta_1(t))\cos(\theta_2(t)) - \sin(\theta_1(t))\sin(\theta_2(t))\cos(90)$$

$$\sigma_2 = \sin(\theta_1(t))\sin(\theta_2(t)) - \cos(\theta_1(t))\cos(\theta_2(t))\cos(90)$$

$$\sigma_3 = \cos(\theta_2(t))\sin(\theta_1(t)) + \cos(\theta_1(t))\sin(\theta_2(t))\cos(90)$$

$$\sigma_4 = \cos(\theta_1(t))\sin(\theta_2(t)) + \cos(\theta_2(t))\sin(\theta_1(t))\cos(90)$$