Project Milestone 03

Objective:

The third milestone of the project is with the following details:

Milestone 03	Deadline Date	Description
M-03 (5%)	Sunday 5th of December, 2021	1. Build the Inverse Position Kinematics function (**MATLAB/Simulink**). 2. Derive several Task-Space trajectories for the robot's end effector to follow (**MATLAB\Simulink**). 3. Derive the Joint-Space trajectory for the robot's joints to follow using (**MATLAB\Simulink**). 4. Validate both the joint space and task space trajectories on the Simscape Multibody model to act the same as in the performed simulation analysis.

**The weight of each deliverable is stated above.*

Requirements:

The requirements from this milestone of the project are as follows:

1. Evaluate the Inverse Position Kinematics on the WHOLE robot using Numerical Approach (Newton-Raphson Approach) of your assigned robot:
    a. **On MATLAB:** prepare a MATLAB function to calculate the differentiation of the end effector's position in X, Y and Z to obtain the matrix representing (dF_i/dq_i), then get the inverse of this matrix (normal inverse or pseudo-inverse based on the type of the robot (Exactly-actuated, Under-actuated or Over-actuated). The function should be named as follows: **J_inv = inverse_jacobian_matrix(q)**. (Note that this function should output a matrix representing the inverse of the matrix of each function differentiated using partial differentiation with respect to q for each joint in a symbolic manner. The function should take as input the angles of the robot in a symbolic manner and the lengths as constant numbers and call the previously created function **X = forward_kinematics_func()** implemented on Milestone 02 to get the end effector's position).

Project Milestone 03

b. **On MATLAB:** prepare a MATLAB function to calculate the values of the joints that are required to move the robot to the desired position in X, Y and Z of the end effector. The function should be named as follows: **q = inverse_kinematics_func(q0, X)**. (Note that the function should take as input the initial guess of all qs as a vector (as vector q0) and the desired X, Y and Z positions of the end effector (as vector X). This function should output a vector representing the angles obtained from the inverse position kinematics approach to move the robot in the X, Y and Z inputted to the system).

2. **Task Space Trajectory Tracking:**
Plan **THREE** different trajectories for your assigned robot to follow using task-space trajectory planning approach (e.g.: Straight Line, Circle, Spiral, Ellipse, Helix, Cone, etc.). These trajectories should be implemented by performing the following steps on MATLAB\Simulink environment **(Build on the MATLAB functions implemented in previous milestones)**:
   a. **On MATLAB:** prepare a MATLAB function for each trajectory to build the desired trajectory by defining the initial point and the destination point for the end effector and the time taken for the end effector to reach from the initial point to the destination point. (Desirable to define the time taken to be not more than 10 seconds to avoid too many computations). The function should be named as follows: **Task_Space = task_traj(X0, Xf, Tf, Ts)**. (Note: that the function should take as input the initial X0, Y0 and Z0 as a vector named **X0** of the robot and the desired final Xf, Yf and Zf as a vector named **Xf** of the end effector, the duration of the trajectory as **Tf** and the sampling time as **Ts.** The function outputs a vector for X, Y and Z of the robot for each time step. (Use a sample time of 0.1 seconds (Ts=0.1) to determine the X, Y and Z values reached at each time step)) (Repeat this function for the other two trajectories).
   b. **On MATLAB:** Use the previously implemented function **q = inverse_kinematics_func(q0, X)** to calculate the values of the joints using Newton Raphson approach for each X, Y and Z obtained by the function at each time step.

Project Milestone 03

3. **<u>Joint Space Trajectory Tracking:</u>**
   Plan the motion of the end effector using joint-space trajectory planning approach by performing the following steps on MATLAB\Simulink environment **<u>(Build on the MATLAB functions implemented in previous milestones)</u>**:
   a. **<u>On MATLAB:</u>** prepare a MATLAB function to derive the equations that should be used for each joint (<u>Use Cubic 3rd order polynomial equations</u>) (Specify the suitable constraints needed for the robot to solve the problem). To build the desired trajectory by defining the initial point and the destination point for the end effector and the time taken for the end effector to reach from the initial point to the destination point. (<u>Desirable to define the time taken to be not more than 10 seconds</u>). You should firstly use the previously implemented function **<u>q = inverse_kinematics_func(q0, X)</u>** to obtain the initial qs **<u>(q0)</u>** given the initial X, Y and Z of the end effector and then call the function with the final X, Y and Z given to obtain the final qs **(qf)**.
   b. Using the outputted q0 and qf, implement a function that should be named as follows: **<u>Joint_Space = joint_traj(q0, qf, qdot0, qdotf, Tf, Ts)</u>** to obtain the value of q of each joint at each time step as a vector. <u>(Note that the function should take as input the initial and final angles of the robot as vectors **q0** and **qf**, respectively. The initial and final angular velocities as vectors **qdot0** and **qdotf**, the duration of the trajectory as **Tf** and the sampling time **Ts.** The function outputs a vector of qs of the robot for each time step. (Use a sample time of 0.1 seconds (Ts=0.1) to determine the position reached at each Ts)).</u>
   c. **<u>On MATLAB:</u>** Use the previously implemented function in Milestone 02 **<u>X = forward_kinematics_func()</u>** to calculate the values of the end effector position for each q obtained by the function **<u>(joint_traj)</u>** at each time step <u>(The function should output a vector of X, Y and Z at each time step)</u>.

4. **<u>In Simscape Multibody:</u>** Simulate the motion of the robot using the same input angles obtained from the task-space trajectories and the joint-space trajectories and visualize the motion of the robot's end effector and compare it with the desired end effector's motion to obtain the X, Y and Z required in both cases the task-space and joint-space.

Project Milestone 03

Submission:

The submission of the 3rd milestone will be in the form of a **Google Drive Link** having a ZIP file named (EDPT1009-TeamNumber-M03.zip), the Google Drive Link should be submitted through:
https://docs.google.com/forms/d/e/1FAIpQLSdO0DH-PgPEqSTHGMygPRwctSmviwfjDU_6cKMZxge5xjOGFQ/viewform?usp=sf_link

The contents of the zip file:

1. **PDF file** includes the following requirements. **Note build on the report done in the previous milestone.**
   - The Inverse Position Kinematics using Newton Raphson.
   - The trajectories using task space trajectory tracking and the corresponding angles obtained. The joint-space trajectory tracking performance that shows the angles and their rate of change.
   - Screenshots of the simulations performed and comments on the results of the cases tested. Results of Simscape Multibody (MATLAB/Simulink) including the comments on the system performance based on the different trajectories implemented. Comments on the overall performance of the robot in all cases.
2. **ZIP file** includes the following items:
   - **Simulink .slx file** of the system actuated by the inputs required (Simulink and Simscape Multibody).
   - **MATLAB.m files** of the required matlab functions.
3. **Narrated video** that includes the following:
   - **Video1.mp4** the inverse position kinematics results and validate the system using the forward position kinematics function implemented in the previous milestone. Input to the inverse position kinematics function the required end effector position to obtain the required angles, then input the outputted angles to the forward position kinematics function to get the same end effector position required.
   - **Video2.mp4** for the task space trajectories (different trajectories) commenting on all the outputs obtained.
   - **Video3.mp4** for the joint space trajectories (cubic equations) commenting on all the outputs obtained.

Project Milestone 03

- **Video4.mp4** testing the output angles from both the task-space and joint-space trajectories on Simscape Multibody validating the performance of the robot following the required trajectory.

The deadline of the **submission** is on **Sunday 5th of December, 2021 at 11:59 PM**. Late Submissions will result in deduction from the grade of this deliverable.