Applied Robotics

Task description (MA Robotics)

IGUS Delta 360 Projekt

ILV

Master - Mechatronics

2. Semester

Autor: Mehmet Ismet Can Dede, Serafin Kollegger

May 14, 2025

Contents

1 Task Description

The task is to generate a model for the Delta 360 parallel manipulator by IGUS in MATLAB using Simulink Multibody blockset. After generating this dynamic model and visual representation, develop Matlab functions to run this model and automatically generate joint-space trajectories.

Figure 1.1: IGUS Delta 360 Robot

2 Multibody Modeling

a. Theory & Example: 3. 10 - 03 - Matlab Multibody Model of Delta 360 Robot and 2. 00 - 03 - Adjustment of Multibody Blocks.

   Hint: Since the Inventor add-an for creating multibody models is no longer available, therefore start with the import of the xml-File provided to you. Some videos on the Sakai page may not be relevant to the project.

b. Task:

i. transfer Multibody model information from the CAD model to Simscape.

ii. make the adjustments of the multibody blocks to finalize the Multibody model of your robot.

3 Forward and Inverse Kinematic

a. Theory & Example: Work through 3. 10 — 02 — Position Level Kinematic Analyses of Delta 360 Robot

b. Task:

i. Implement the forward kinematics equations in the Simulink model and verify it using the Multibody model of your robot.

ii. Report the verification results in terms of errors in terms of forward kinematics.

iii. Report the verification results in terms of errors for inverse kinematics

4 Joint Space Motion

a. Point-to-Point

i. Choose three task space positions within the workspace.

ii. Move this information from task space to joint space via inverse kinematics.

iii. Generate a synchronized trapezoidal velocity trajectory for each joint moving from one point to another in joint space (using maximum acceleration, and maximum velocity of the joints provided in the table 4.1)

iv. Generate a file with time series data see appendix A.2 and plot the resulting task space trajectory to show the deviation from the straight line.

v. **For extra points**, repeat (iii) with trapezoidal acceleration trajectory using maximum jerk, acceleration and velocity limits.

4.1 JOINT SPACE MOTION LIMITS

Parameter	Value
Maximum velocity	50 mm/s
Maximum acceleration	100 mm/s^2
Maximum jerk	5 m/s^3

Table 4.1: Joint space motion limits

5 Task Space Motion

a. Point-to-Point

i. Use the same task space positions as in task 4.i.

ii. Generate a trapezoidal velocity trajectory for moving from one point to the other in task space (using, maximum acceleration, and maximum velocity of the end-effector provided in the table 5.1)

iii. Generate a file with time series data in joint space (using the inverse kinematic for each position data of the trajectory), see appendix A.2. Compare the results with Task 4 in terms of joint velocity, acceleration and total time.

iv. **For extra points**, repeat (iii) with trapezoidal acceleration trajectory using maximum jerk, acceleration and velocity limits in task space.

5.1 TASK SPACE MOTION LIMITS

Parameter	Value
Maximum velocity	40 mm/s
Maximum acceleration	70 mm/s^2
Maximum jerk	5 m/s^3
Nominal velocity	20 mm/s
Nominal acceleration	50 mm/s^2
Nominal jerk	3 m/s^3

Table 5.1: Task space motion limits

6 Continuous Trajectory

Create continous trajectory function

a. Create continuous trajectory in between the chosen task space points from chapter 4.i without any full-stops at the via position. Choose the speed of operation for the up and down motion as the half of the maximum possible joint motion speed. Compare the results with task 4 and 5 in terms of task completion duration and motion.

i. Calculate the deviation from the straight-line while performing this motion with and without extra via points and discuss on the results.

7 Pick-and-Place Function

7.1 PICK-AND-PLACE MOTION DESCRIPTION

a. The necessary input parameters are the pick-up location and the drop-down location.

i. Refer to Figure A.1 for the location identifiers → A2 to K3 or H3 to F1 . . .

ii. The function call can look similar to $\text{pick\_and\_place\_motion}(\text{A2}, \text{K3}, \ldots)$

iii. Result of the function is again a time series, see Appendix A.2.

b. **For extra points**, you should create a function that can perform multiple pick and place operations one after another.

c. The following Motion Path should be completed using the pick and place function

i. The robot should initiate from the home position and return to the home position after one pick-and-place task is completed. Then, further pick-and-place task can be started. (Initiate at home position)

ii. Robot should move from the home position to the the pick-up location with an offset of about -15 mm in the Z axis. This motion should be completed in minimum amount of time using the joint space trajectory planning with joint space motion limits.

iii. Move down 15 mm to complete the pick-up, stay at that location for 1 second and move up 15 mm. This motion is to be carried out with nominal motion limitations.

iv. Move to the drop-down location in minimal time and stop at 15 mm above the drop-down location. This motion should be completed in minimum amount of time using the task space trajectory planning with task space motion limits.

v. Move down 15 mm to complete the drop-down, stay at that location for 1 second and move up 15 mm. This motion is to be carried out with nominal motion limitations.

vi. Move to the home position and come to a full-stop. This motion should be completed in minimum amount of time using the joint space trajectory planning with joint space motion limits.

7.2 FUNCTION REQUIREMENTS

Further Parameters needed for calculating the trajectories are:

i. In joint space: maximum velocity and acceleration limits. **For extra points**, also maximum jerk limit.

ii. In task space: maximum velocity and acceleration limits. **For extra points**, also maximum jerk limit.

iii. In task space: nominal velocity and acceleration limits. **For extra points**, also nominal jerk limit.

Define the function as follows:

- The function's syntax is: `pick_and_place_motion(A2, K3, joint_max_vel, joint_max_acc, task_max_vel, task_max_acc, task_nom_vel, task_nom_acc)`
- For extra points, the function's syntax is: `pick_and_place_motion(([A2, H3, ...], [K3, F1, ...], joint_max_vel, joint_max_acc, joint_max_jerk, task_max_vel, task_max_acc, task_max_jerk, task_nom_vel, task_nom_acc, task_nom_jerk)`

Outputs expected from the function are as follows:

1. Generate a file with 4 columns:
   - The first column should have the time stamps starting from 0 seconds with 10 ms increments.
   - The second column has the joint positions for Leg 1, the third column for Leg 2, and the fourth column for Leg 3.
2. The Multibody model should be initiated and run by using the generated file in the previous step.

8 Submission

Submit the work in a report format including the created function, outputs of the function in terms of the file, warning messages and plots indicating the verification made.

Furthermore, create a video (or multiple videos) of the manipulators motion and explain the results and verification made in each task. The videos must be created using your on voice, no text to speech tools are allowed.

A Appendix

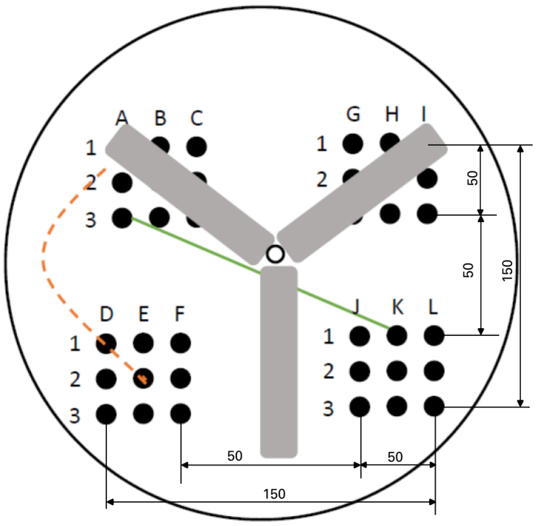

Figure A.1: Table-top piece under the robot with location markers. The dashed orange line represents the joint-space trajectory planning result for moving from one location to another. The solid green line represents the task-space trajectory planning result for moving from one location to another.

	1	2	3	4
1	0	0.0200	0.1000	0.0800
2	0.0100	0.0202	0.0999	0.0800
3	0.0200	0.0204	0.0998	0.0799
4	0.0300	0.0206	0.0997	0.0799
5	0.0400	0.0208	0.0996	0.0798
6	0.0500	0.0210	0.0995	0.0798
7	0.0600	0.0212	0.0994	0.0797
8	0.0700	0.0214	0.0993	0.0797
9	0.0800	0.0216	0.0992	0.0796
10	0.0900	0.0218	0.0991	0.0796
11	0.1000	0.0220	0.0990	0.0795
12	0.1100	0.0222	0.0989	0.0795
13	0.1200	0.0224	0.0988	0.0794
14	0.1300	0.0226	0.0987	0.0794

Figure A.2: Possible format of time series data to be used in Simulink. With first column as time, second column as first joint position data, third column as second joint position data and fourth column as third joint position data.

Figure A.3: The From Workspace block can be used to feed the trajectory into the Simscape model.

List of Figures

List of Tables