MSE893 - Advanced Kinematics for Robotic Systems

Robot for Airport Luggage Handling: Project Phase3 Report

Cheng-Lin Wu

301606107
Department of Engineering Science
Simon Fraser University
cheng-lin_wu@sfu.ca

July 7, 2024

Table of Content

Abstract

The manipulator for airport luggage designed in the project faces a heavy-duty task. To minimize the sudden changes in motion imposing high mechanical stresses on the joints and links of the manipulator, in this report, we demonstrate the implementation of trajectory planning satisfying C2 continuity (continuous velocity and acceleration). To achieve the continuity, a high-order polynomials method is adopted for joint movement, and the relevant profile of joint position, velocity, and acceleration are visualized. By combining the trajectory planning with the kinematics functionality done in the previous phases, an animated simulation of manipulator picking/placing luggage is also obtained in this phase. Finally, we derive the Jacobian for the manipulator and analyze the inverse static force problem for each joint along with potential singularity.

1. Design of Luggage Manipulator

As a recap, the luggage manipulator consists of two revolute joints for planar movement (XY plane) and one prismatic joint stretching vertically to reach to target luggage. The following Figure 1 shows the schematics of the robot and Figures 2 and 3 visualize the corresponding reachable workspace.

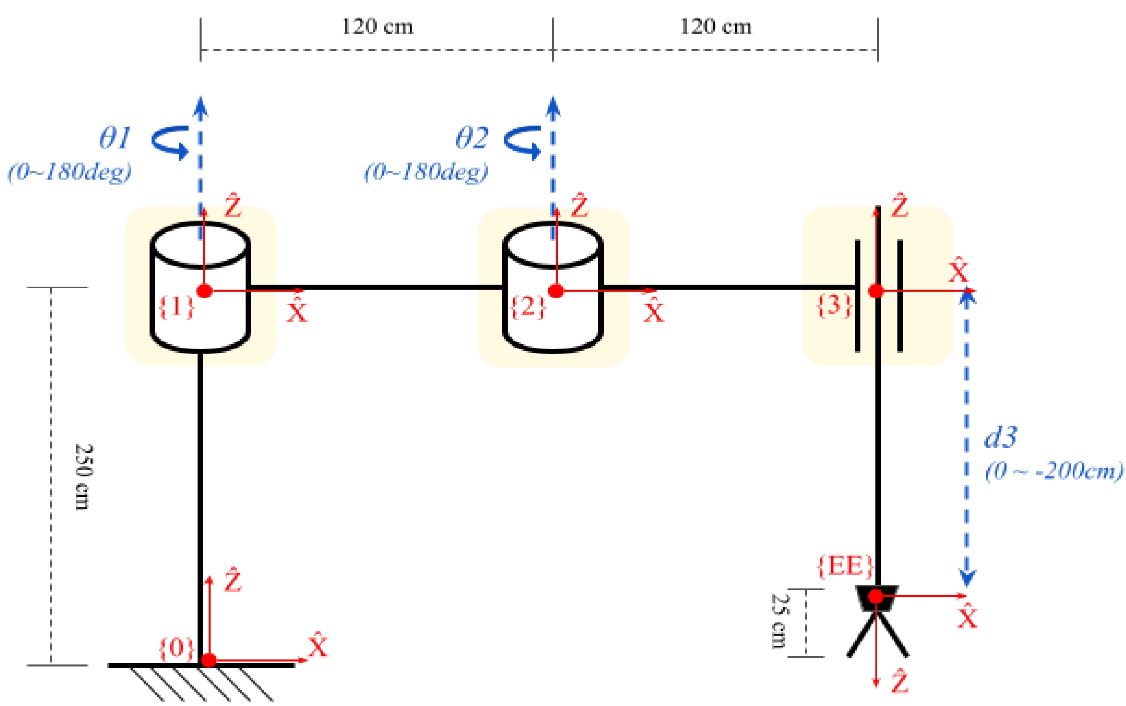


Figure 1. Manipulator kinematic layout with joint frames attached

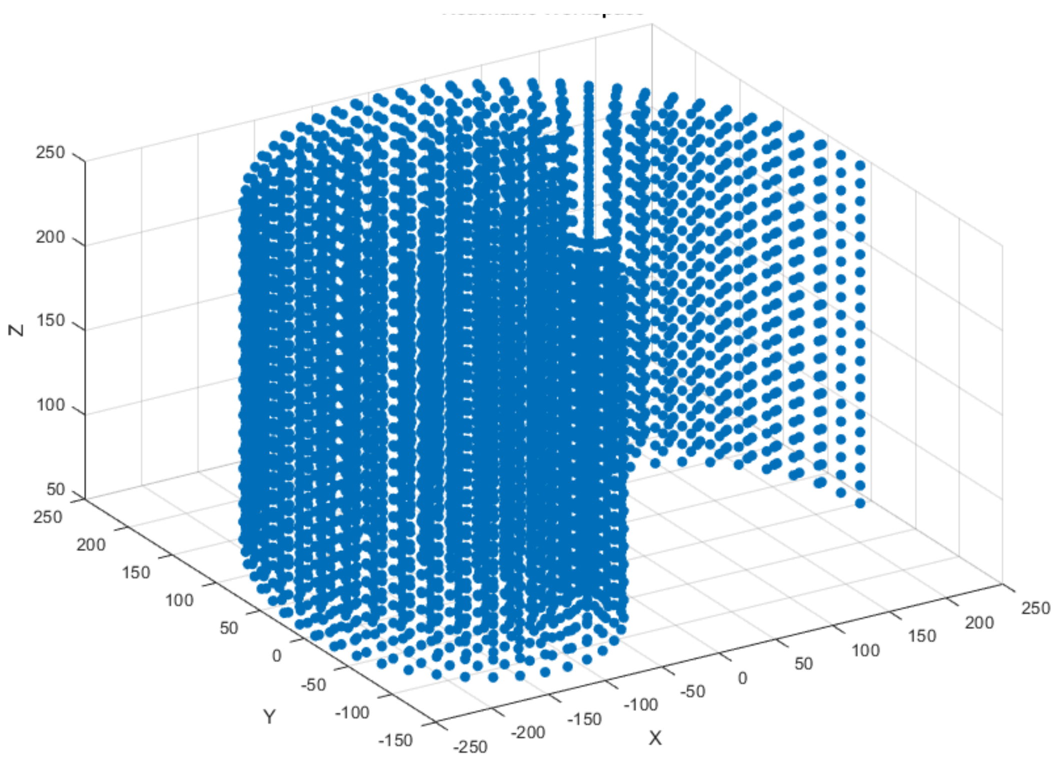


Figure 2. 3-dimensional visualization of the manipulator's reachable workspace

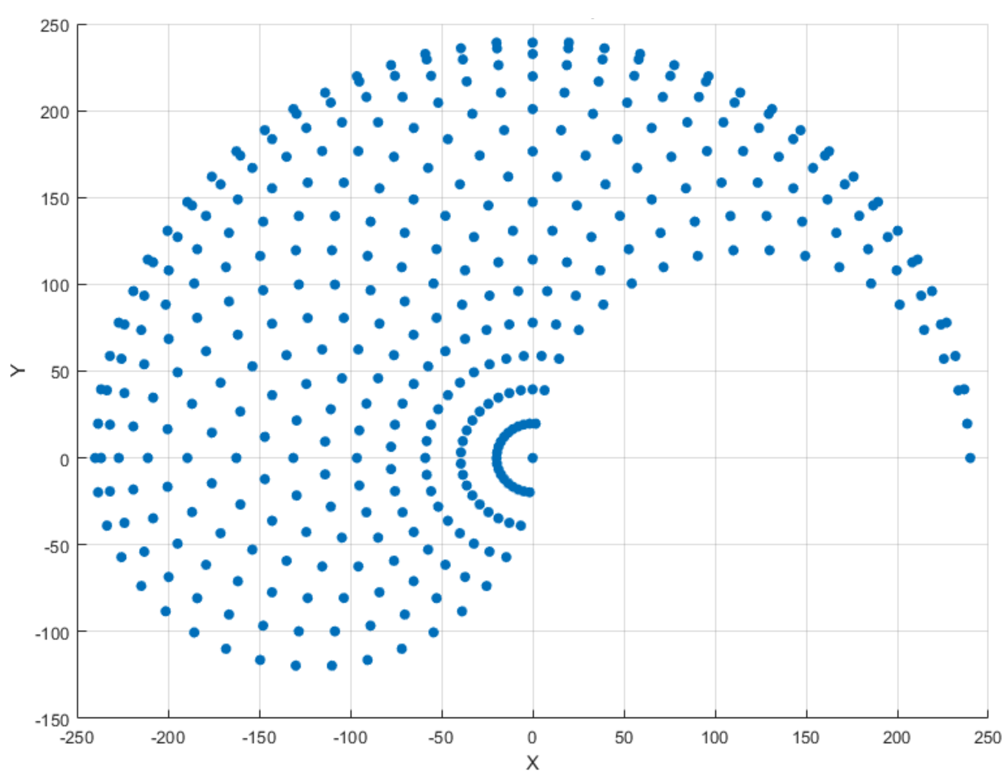


Figure 3. XY projection of manipulator's reachable workspace

2. Path Generation

With the reachable workspace visualized, we are now able to design a path consisting of waypoints in the workspace. Typically, the pose of a waypoint is represented by a position vector (x, y, z) and an orientation vector (α, β, γ). However, since all the revolute joints' z-axis are parallel to the z-axis of the station frame (frame $\{0\}$), the orientation vector of each waypoint can be simplified as (ϕ), which indicates the rotation about the z-axis.

Table 1. Waypoints of the designed path to pick and place a target luggage.

Waypoint No.	Position: X	Position: Y	Position: Z	Orientation: ϕ
1	240	0	200	0
2	240	0	150	0
3	240	0	250	0
4	60	103.92	200	0
5	-103.92	60	200	0
6 (Pick Luggage)	200.76	115.91	50	0
7	-28.94	219.83	150	0
8	-28.94	219.83	150	90
9 (Place Luggage)	-103.92	-60	50	90
10	84.85	204.85	200	0

Table 1 lists waypoints of a path for the manipulator to move a target luggage to a designated place. In the first several waypoints, the manipulator stretches vertically and swings horizontally without load, this is to simulate the action of object sensing and locating the target. In practical use, an RGB camera with depth-sensing features can be mounted onto the end-effector of the manipulator. By doing the initial scanning moves, a 3-dimensional scene can be reconstructed with information on obstacles and target luggage. After understanding the scene, the manipulator reaches the target luggage at waypoint 6 and places it at the destination at waypoint 9. Finally, the robot returns to its initial pose and is ready for the next task.

For simplicity, the scenario of the generated path is for a single target without obstacles. Prior work proposes both algorithmic methods and machine learning methods for route planning with consideration of priority and obstacle avoidance [1][3]. Some pick-and-place strategies for e-commerce settings can also be adopted in the airport luggage dispatching environment[2].

3. Trajectory Generation

To fulfill the continuity requirement of joint velocity and acceleration, we adopt the high-order polynomials which are capable of specifying velocity and acceleration for the start and end of a path segment. The equation of polynomials is as below:

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \tag{1}$$

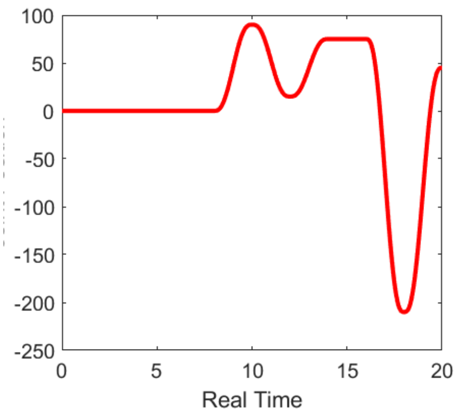
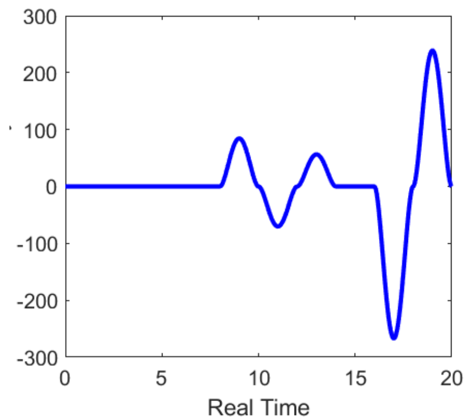
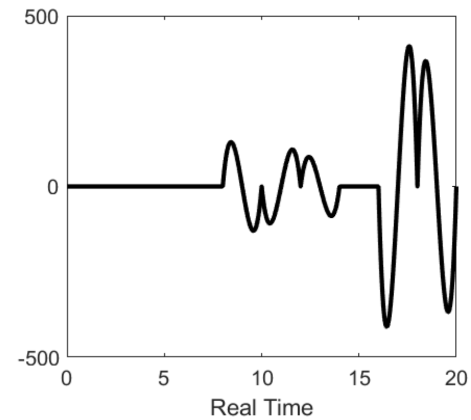
To calculate the coefficients of the polynomials, we need to have joint positions, velocity, and acceleration at the start/end of the segment as constraints. The value of joint position can be obtained by performing inverse kinematics as we have done in the previous phase. For velocity and acceleration, we simply assign them as zero. In other words, the manipulator should gradually slow down as it moves close to the next waypoint and gradually speed up again when it starts the next segment. With the above constraints, we can calculate each coefficient as below:

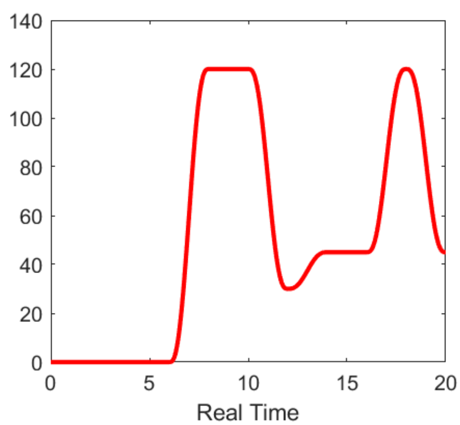
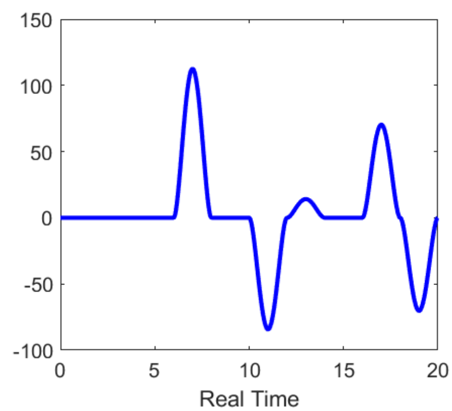
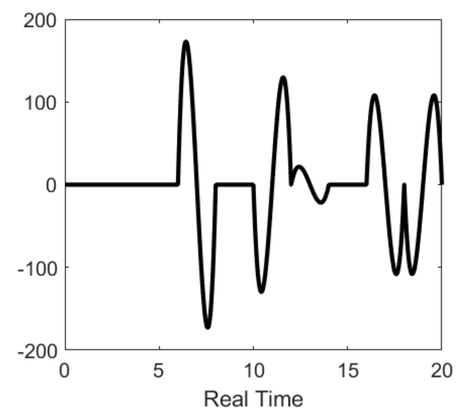
$$\begin{aligned} a_0 &= \theta_0 \\ a_1 &= 0 \\ a_2 &= 0 \\ a_3 &= \frac{20\theta_f - 20\theta_0}{2t_f^3} \\ a_4 &= \frac{30\theta_0 - 30\theta_f}{2t_f^4} \\ a_5 &= \frac{12\theta_f - 12\theta_0}{2t_f^5} \end{aligned} \tag{2}$$

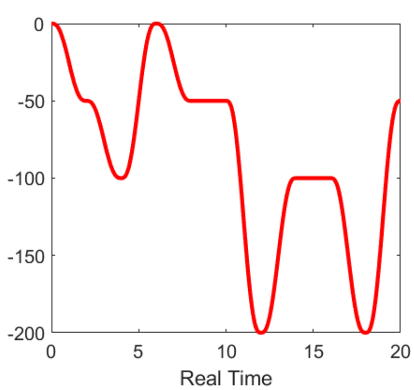
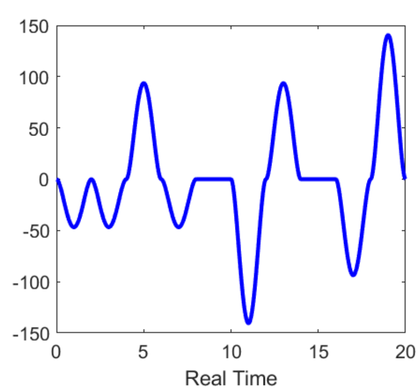
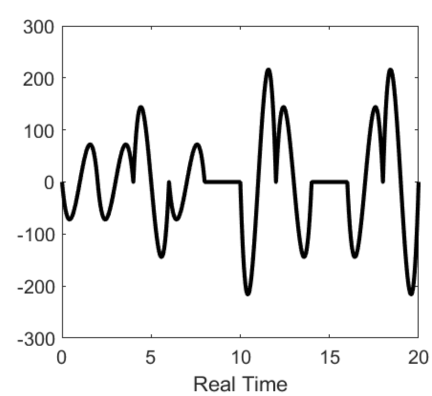
The following is the pseudocode of integrating trajectory planning, inverse kinematics, and forward kinematics together to simulate the luggage manipulation.

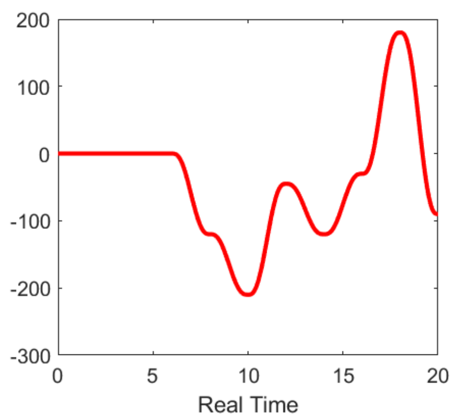
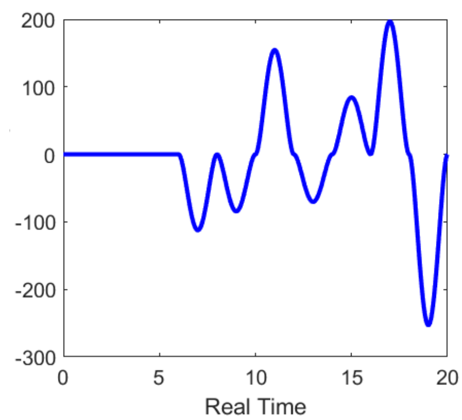
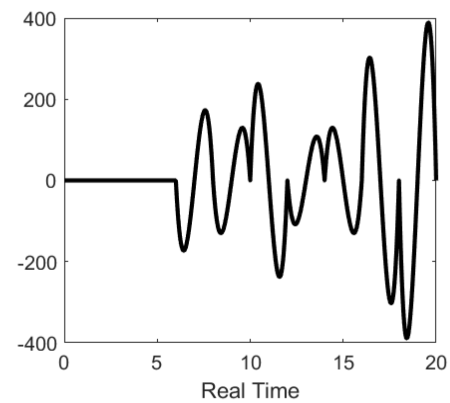
```
sampling rate: 25fps
waypoints list: a list of 10 poses (x, y, z, ϕ)
segment time duration: 2 seconds

for each segment:
    perform inverse kinematics to calculate destination joint values
    for each joint:
        calculate the coefficient of trajectory polynomials
        for each sampled time point t:
            move joint to θ(t)
```

The following are joint profiles of position, velocity, and acceleration:

Joint1 Position	Joint1 Velocity	Joint1 Acceleration
		

Joint2 Position	Joint2 Velocity	Joint2 Acceleration
		

Joint3 Position	Joint3 Velocity	Joint3 Acceleration
		

Joint4 Position	Joint4 Velocity	Joint4 Acceleration
		

The following Fig 4 is the trace of end-effector visualized using comet3 function in MATLAB.

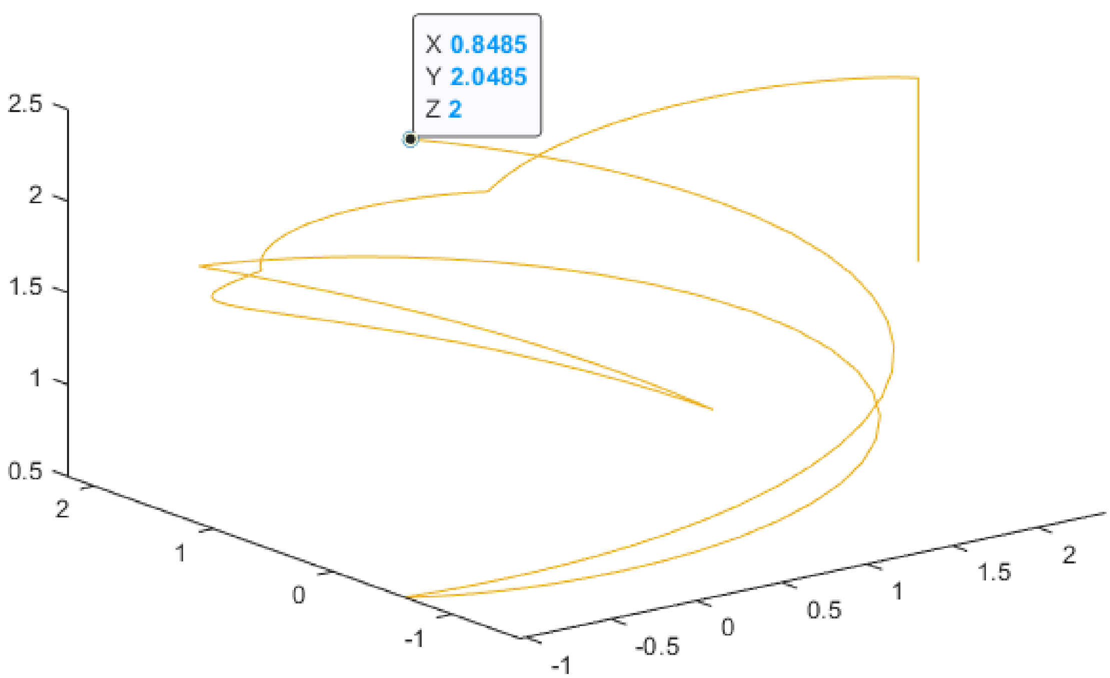


Figure 4. Comet plot of end-effector trace

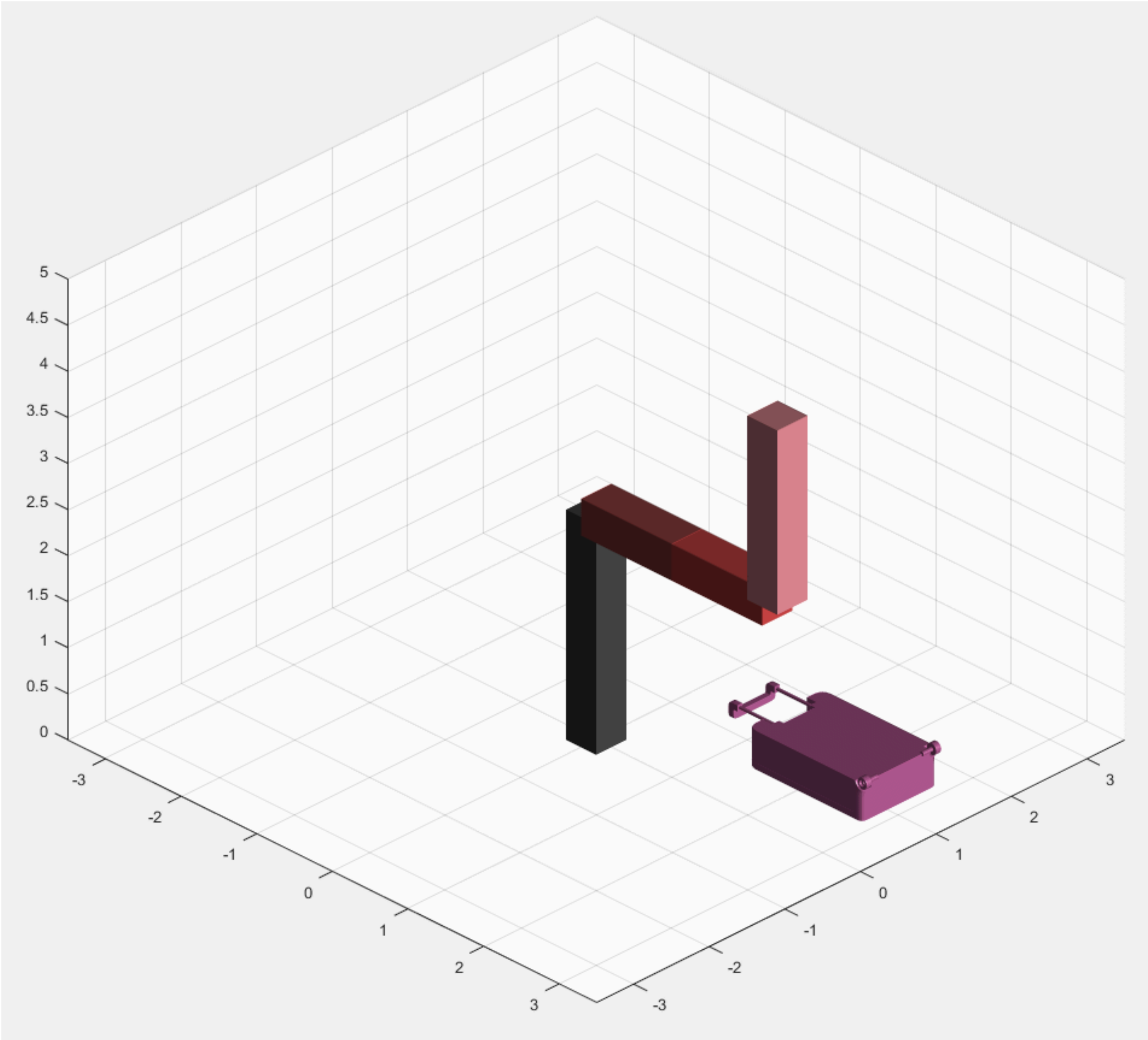


Figure 5. (GIF) Simulation of luggage manipulator moving along a designed path

MATLAB code and visualization results are available at:
https://drive.google.com/file/d/1SkwUiOU6gnkkX0fS6fyOZrudnEzzDtJg/view?usp=sharing

A video of the manipulator simulation is also on YouTube:
https://youtu.be/BK6zJXsJYUs

4. Jacobian And Force Analysis

The Jacobian matrix is a fundamental concept in robotics, providing a relationship between the velocities in joint space and the velocities in Cartesian space. For a manipulator, the Jacobian can be derived from the kinematic equations describing the manipulator's motion.

During the phase of forward kinematics, we obtain a transformation from frame {0} to frame {ee}. The relationship between joint values and the end-effector's Cartesian position can be extracted from the displacement part of the transformation:

$$\begin{aligned} x &= 120cos(\theta_1) + 120cos(\theta_1 + \theta_2) \\ y &= 120sin(\theta_1) + 120sin(\theta_1 + \theta_2) \\ z &= d_3 + 250 \end{aligned} \tag{3}$$

The Jacobian matrix can be derived by doing a partial derivatives in the following form:

$$J = \begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} & \frac{\partial x}{\partial d_3} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} & \frac{\partial y}{\partial d_3} \\ \frac{\partial z}{\partial \theta_1} & \frac{\partial z}{\partial \theta_2} & \frac{\partial z}{\partial d_3} \end{bmatrix}$$

$$= \begin{bmatrix} -120sin(\theta_1) - 120sin(\theta_1+\theta_2) & -120sin(theta_theta_2) & 0 \\ 120cos(\theta_1) + 120cos(\theta_1+\theta_2) & 120cos(\theta_1+\theta_2) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

The singularity happens when the manipulator loses 1 or more degrees of freedom, the condition can be fulfilled when the determinant of Jacobian becomes zero. The expression of the Jacobian can be simplified as below:

$$Det[J] = 120^2 sin(\theta_2) \tag{5}$$

This implies that when $\theta_2 = 0$ *or* $\theta_2 = 180$ the manipulator loses degrees of freedom because Link 1 and Link 2 are either fully extended or folded back to each other.

Jacobian matrix is also an essential tool to solve inverse-force problems, that is, given a force exerted on the end-effector, find out the required force/torque of each joint. The corresponding force/torque vector of each joint τ can be obtained by multiplying J^T with the end-effector's force:

$$\tau = \begin{bmatrix} -120sin(\theta_1) - 120sin(\theta_1+\theta_2) & 120cos(\theta_1) + 120cos(\theta_1+\theta_2) & 0 \\ -120sin(\theta_1+\theta_2) & 120cos(\theta_1+\theta_2) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \tag{6}$$

Suppose the force exerted on the end-effector is $[10N, 5N, 4N]^T$ and $\theta_1 = \theta_2 = 45^\circ$. By plugging these values into equation (6), we can obtain the required force/torque of each joint $\tau = [-16.23N, -12N, 4N]^T$.

5. Conclusion

This report illustrates the implementation of trajectory planning capable of maintaining continuous velocity and acceleration at each waypoint along a designed path. The derivation of trajectory polynomials and the MATLAB simulation pseudocode are thoroughly detailed and complemented by extensive visualizations.

Additionally, we delve into the Jacobian matrix of the manipulator, which is crucial for understanding the relationship between joint rates and end-effector velocity. Using the inverse Jacobian matrix, we also conduct a static force analysis, supported by a numerical example. Future work will focus on addressing obstacle avoidance and handling multiple targets, both common in real-world applications. Moreover, considerations of joint rate and acceleration limitations will be incorporated.

MATLAB code and visualization results:
https://drive.google.com/file/d/1SkwUiOU6gnkkX0fS6fyOZrudnEzzDtJg/view?usp=sharing

Manipulator simulation on YouTube:
https://youtu.be/BK6zJXsJYUs

Reference

[1] Iriondo, Ander, et al. "Pick and place operations in logistics using a mobile manipulator controlled with deep reinforcement learning." Applied Sciences 9.2 (2019): 348.
[2] Kumar, Swagat, et al. "Design and development of an automated robotic pick & stow system for an e-commerce warehouse." arXiv preprint arXiv:1703.02340 (2017).
[3] Silva, João Sousa E., Pedro Costa, and José Lima. "Manipulator path planning for pick-and-place operations with obstacles avoidance: an A* algorithm approach." International Workshop on Robotics in Smart Manufacturing. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.