Technion – Israel Institute of Technology

Faculty of Mechanical Engineering



Project – Part 1

Kinematics, Dynamics, and Control of Robots

036026

|  |  |  |
| --- | --- | --- |
| Alon Spinner | 305184335 | alonspinner@gmail.com |
| Shahar Tsadok |  | Shahar507@gmail.com |

March 10, 2021

# Finding the Direct Kinematics of the Robot

Transformation matrices transfer homogenous points from one coordinate system, say , to the other, .

A transformation matrix from basis to is structed from rotational and translational elements. The rotation matrix has the basis vectors of in its columns with the representation of . The translation vector is the translation in the representation of

|  |  |
| --- | --- |
|  |  |

# Finding Inverse Kinematics Under Assumption of No Constraints

The problem of inverse kinematic can be formulated as such:  
given a target transformation matrix which encapsulates the position and orientation of the tool in world coordinate system

Find the joint values .

|  |  |
| --- | --- |
| To solve the inverse kinematics, we break the problem into 3 parts.   1. We denote point to be the origin of coordinate system and solve for it given the target transformation matrix. 2. Solve for using the transformation matrix found by direct kinematics. 3. Find by equating terms of from two different representations: the target matrix and the direct kinematics. |  |

## Finding Point Given Target Transformation Matrix

Given that the direction of the tool and position is known, finding is straightforward.

## Finding From Point and Joint-Based Transformation Matrix

The location of in system is therefore

A general formula can be applied to our case.

Rewriting the first two equations in

Denoting terms and using the first equation in we can find .

Finding is done by substituting the terms in the second equation of , and reorganizing and squaring the third equation in

Adding the two equations we can find

To find we utilize the third equation in

If both and equate to zero, equation will be impossible to solve, hence we make a small distinction which will nullify in such a case.

## Finding by Equating Two Representations of

Given a target rotation matrix, we can compare it to the rotational part of equation and find the required joint values.

From the placement in equation

From the placement in equation

Substituting for in equation (2.3.1)

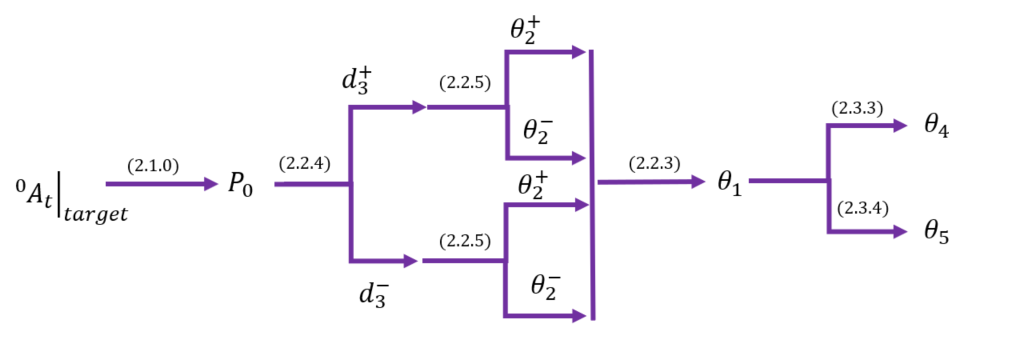
We can then find with equations

To find we will use the placements in equation

Plugging into the general formula in

## Inverse Kinematic Flow and Equations

To sum things up, we provide a figure of the computation flow from the target transformation matrix to the joints. Notice that two joints, and have ambiguities that need to be decided prior to the computation.



# Solving for the Jacobian

The Jacobian matrix transfers the joint velocities to tool velocities. It can be written as two matrices: transfers to linear velocity, and transfers to angular velocity

To find we differentiated the translation vector found by the direct kinematics (equation ) with respect to joint vector **.**

To find the Angular Jacobian we used the Whitney method.  
For each revolute joint ,the respective Jacobian column is the direction in which the joint turns about.

The angular Jacobean for linear joints is the zero vector.  
To find the direction for each revolute joint we used the rotation matrices computed in the direct kinematics found in ‎1.

|  |  |
| --- | --- |
|  |  |

Plugging in the values

To find the Jacobian in the tools system coordinates we multiplied both linear and angular Jacobians by the rotation matrix

# Singular Robot States

We assume and .

The singular states of the robot are obtained by joint values that nullify the robot’s Jacobean determinant. As the assumptions make our robot a de-facto smaller version of itself, we will be looking only on .

This expression equates to zero when either or .

For , the robot can take the following two forms:

=0

In these two positions, the robot movement in the direction is impossible.  
Changing will result in a movement in the direction, while changing either or will move the tool in the direction.

For , the robot will appear as follows:

Here the motion is impossible in the direction when has no effect on the location of the tool.

We would like to note that when both **and** two degrees of freedom will fall and the robot won’t be able to create tool movement in both the and directions. We can prove that by plugging in said values to the linear Jacobean in the tool co-ordinate system:

The first two rows of the Jacobian are zero making the motion in impossible. Also, the second column is zero which prevents changes in to affect the tool’s location.

# Motion Planning

Provided the constant lengths of the robot:

And movement constraints:

We were asked to move the tool from one location to the other, ignoring tool orientation, in three different trajectories: constant velocity, trapezoid velocity and polynomial with zero velocity and acceleration at the end points. The motion is to take .

Analytical trajectory profiles were computed independently for each axis in the 3-D world.

Inverse kinematics was then used to compute the joint position for each tool world position with ambiguity in the form of *elbows* being decided by movement constraints.

Joint velocities and acceleration were computed in two methods:

* Differentiating the joint positions:
* The use of the linear Jacobean computed analytically

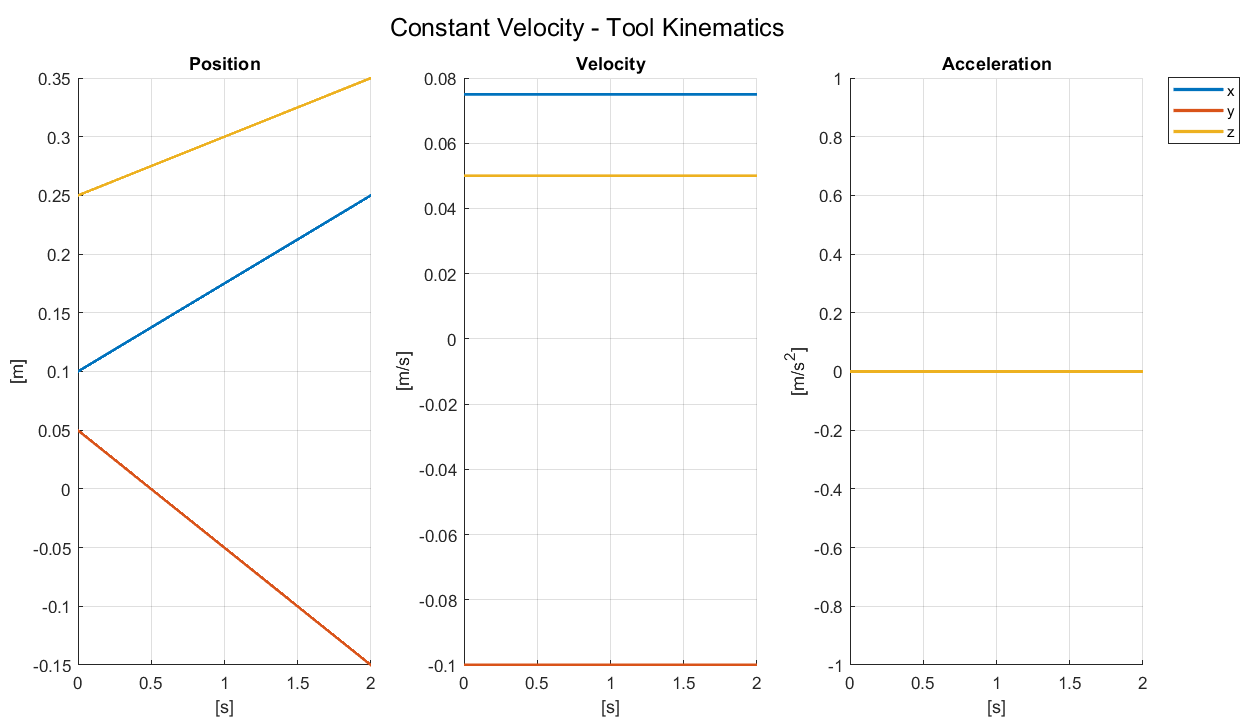
In the subsequent sub-sections, an analytical motion trajectory is derived for each of the required profiles, followed by graphic outputs of its results.  
The last sub-section shows graphic comparison between the motion profiles.  
**Note**: the joint constraints allowed for two possible solutions, we chose the positive one for the inverse kinematics.

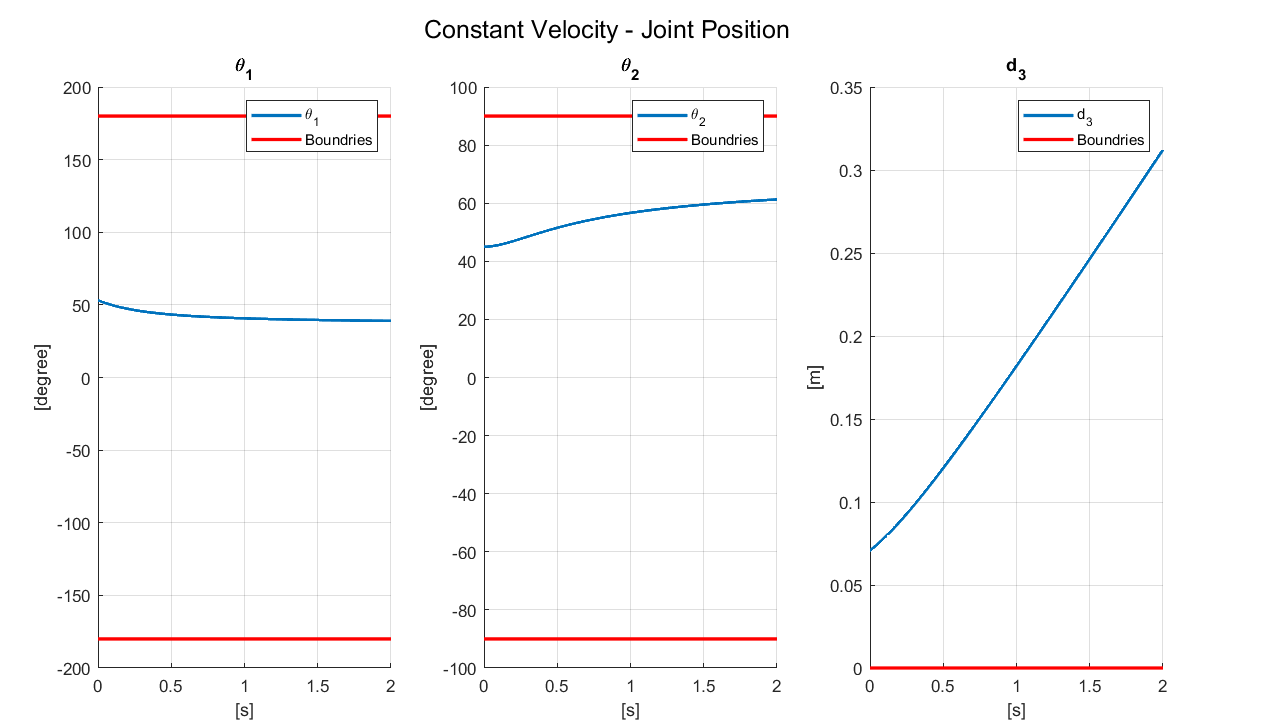
|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

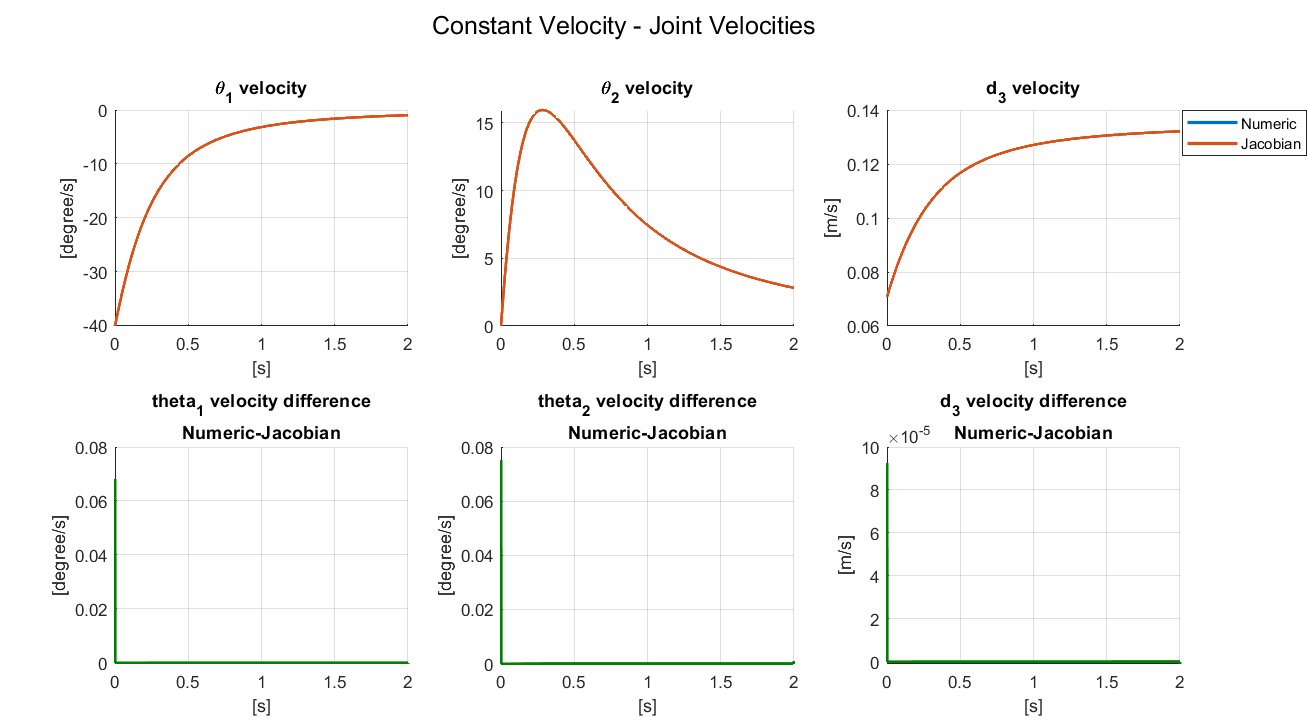
## Constant Velocity

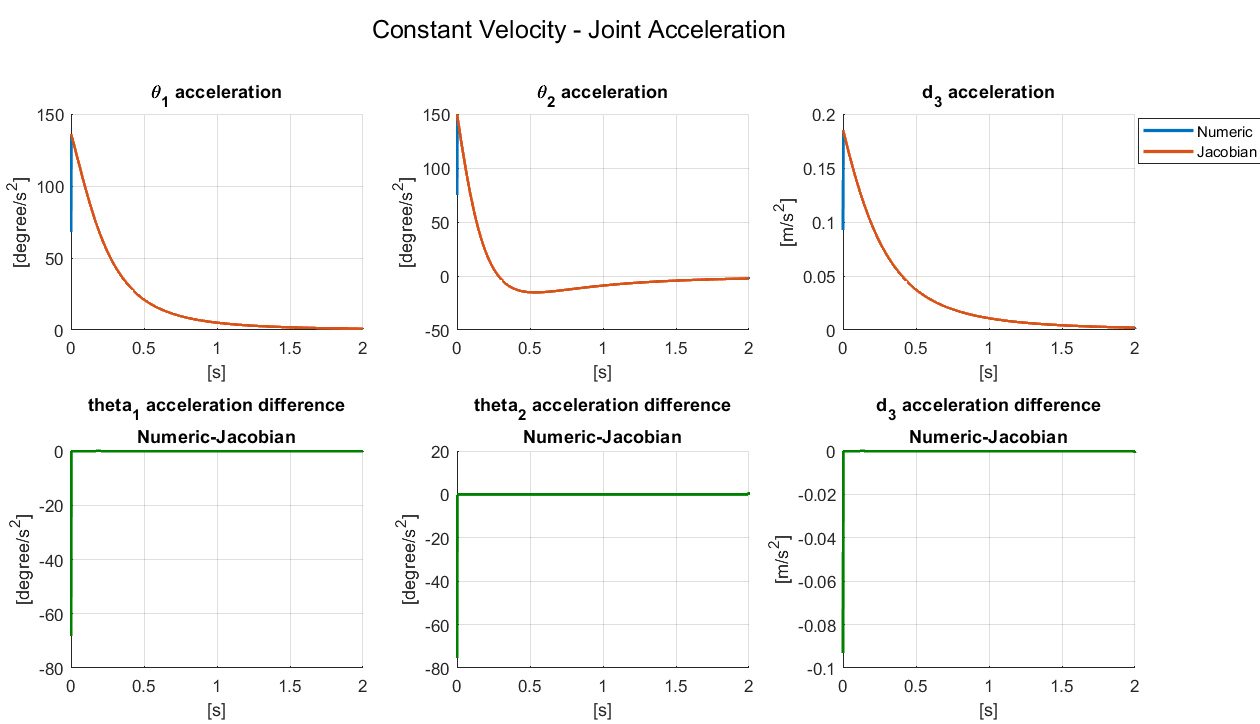
Provided that the velocity is constant between two points [ traversed in time , the kinematics are:

### Graphic Results







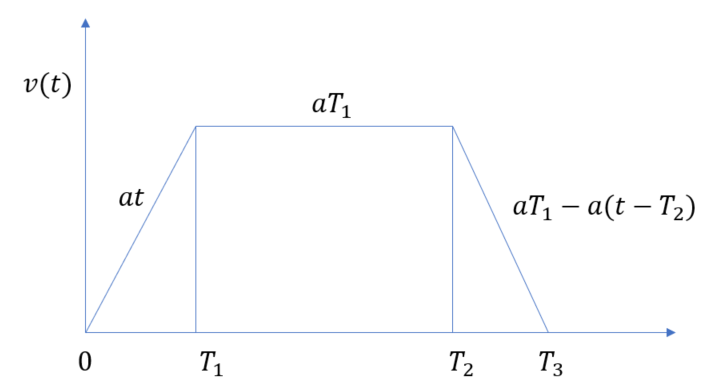


## Trapezoid Velocity

Symmetrical trapezoid velocity movement can be described by as little as three parameters:

* Total Time,
* Zero Acceleration Time
* Amount Traveled

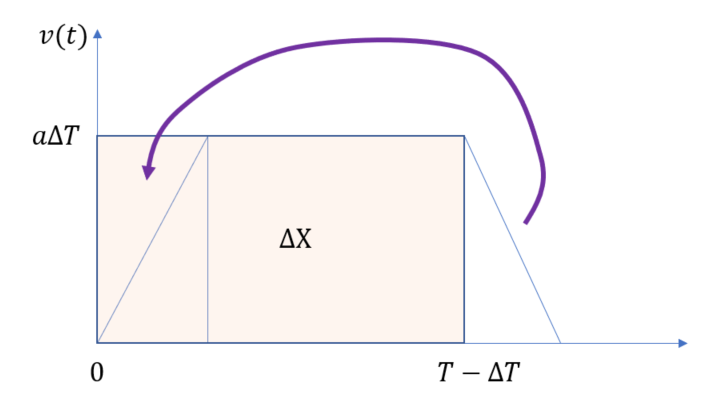
We derivate the acceleration on the velocity ramp-up from these properties.



To solve for we equate the final position to a known number .

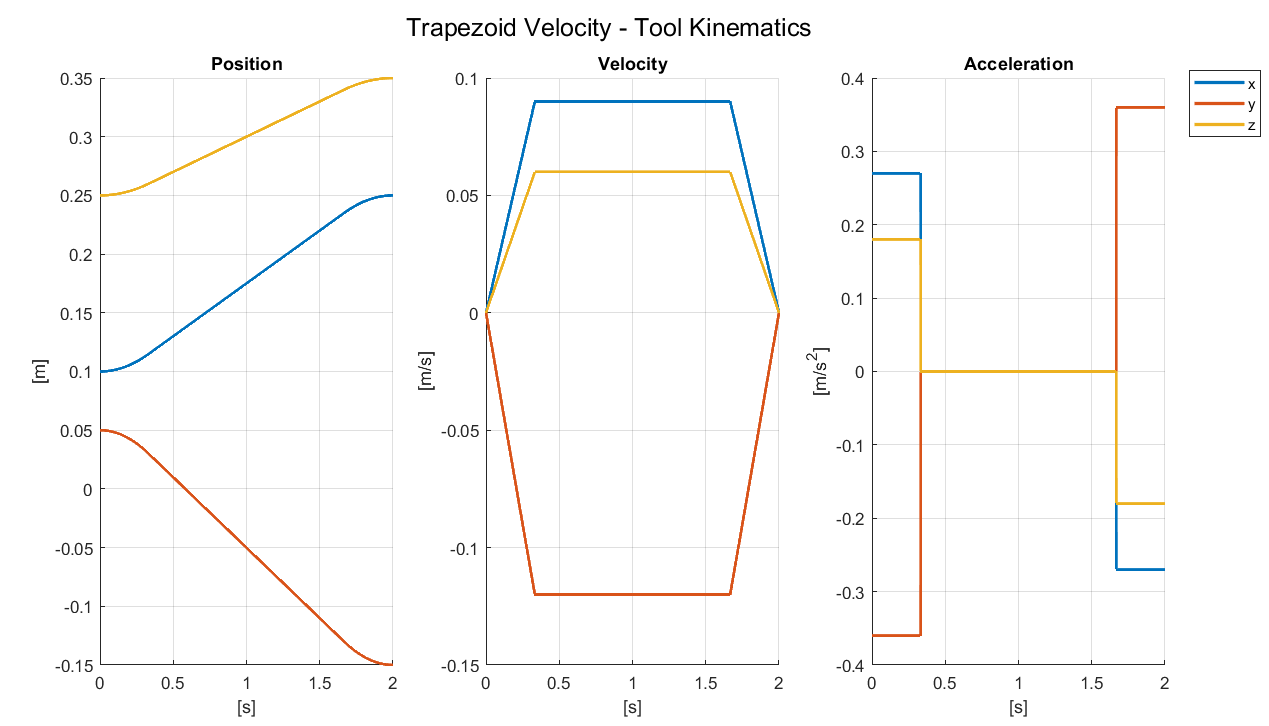
Subsisting and

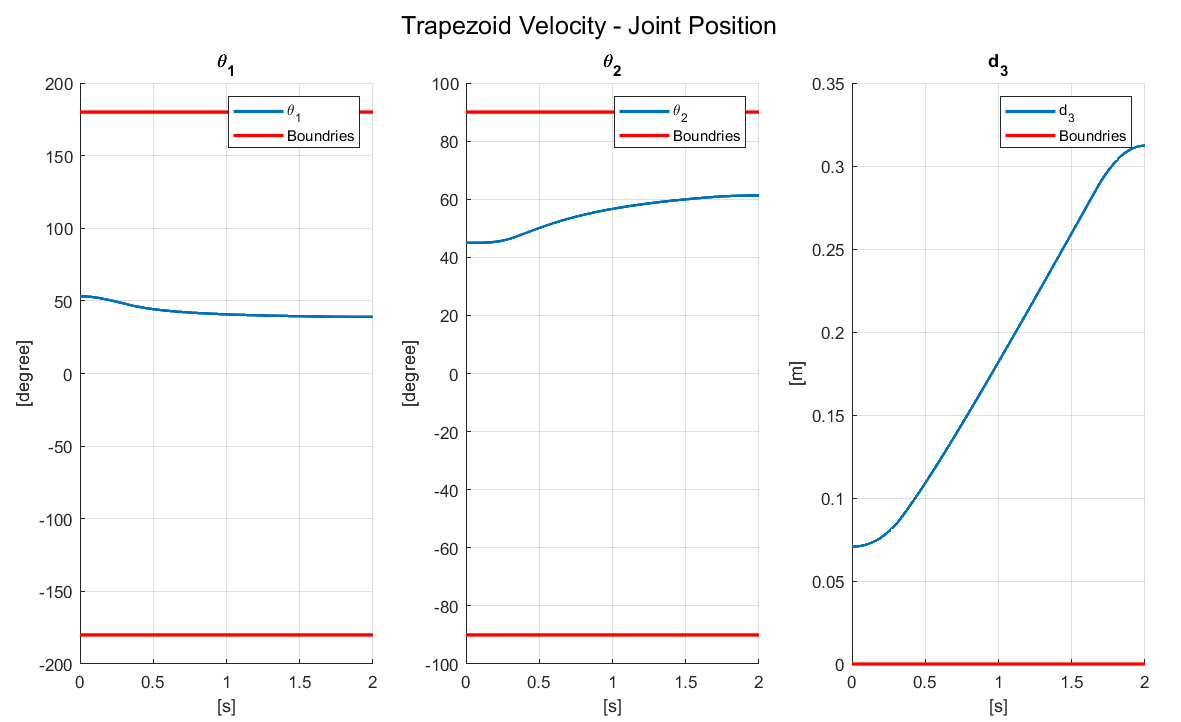
This result sits well with integration by ‘rectangular-ing’ the trapezoid:

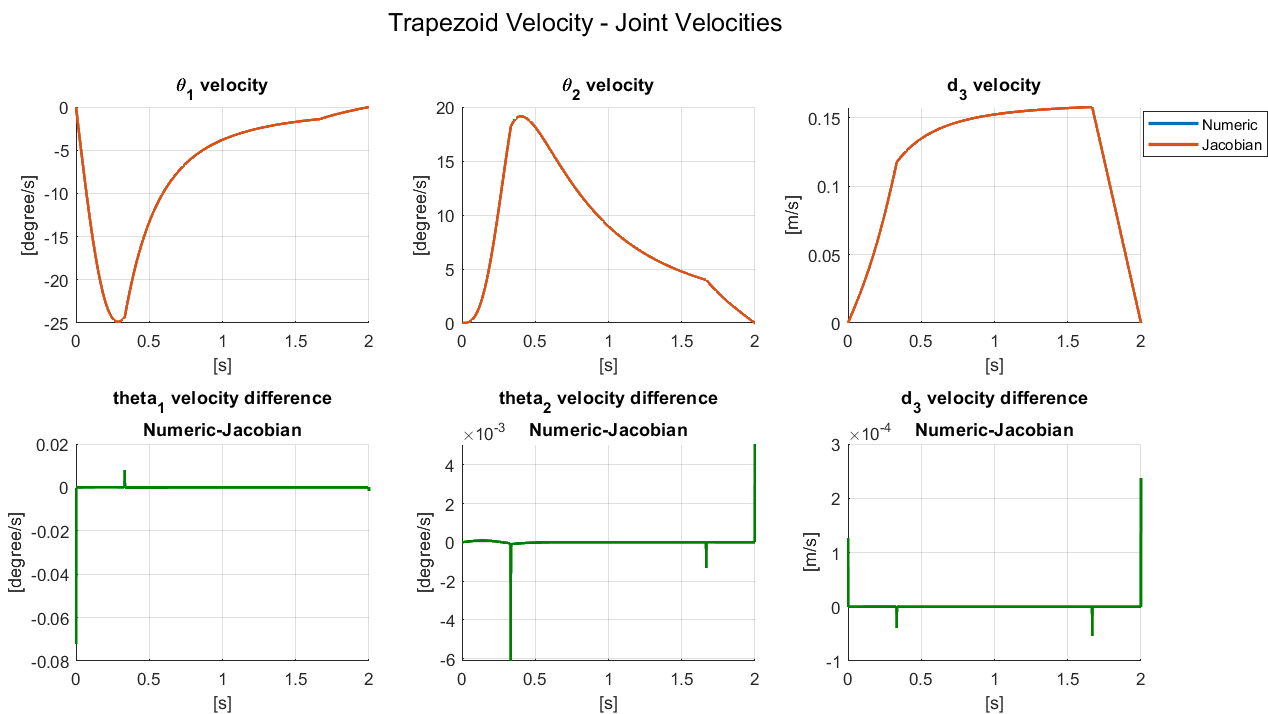


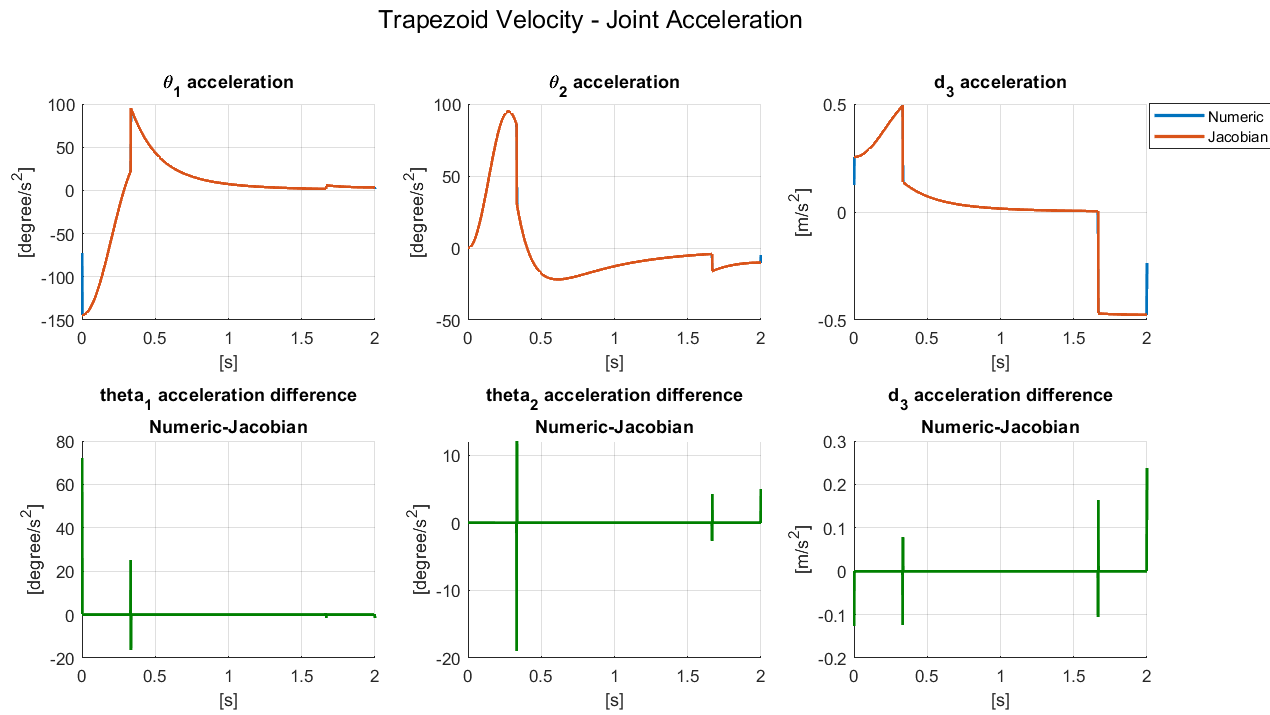
In our assignment .

### Graphic Results





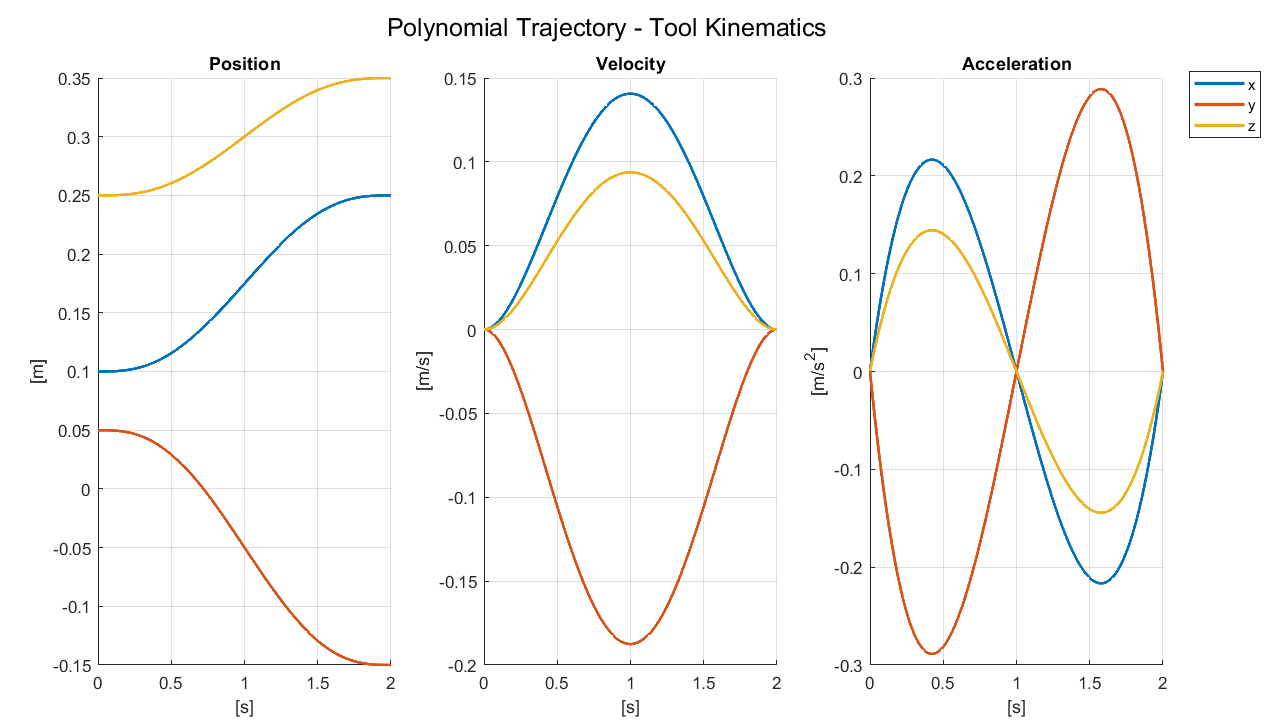


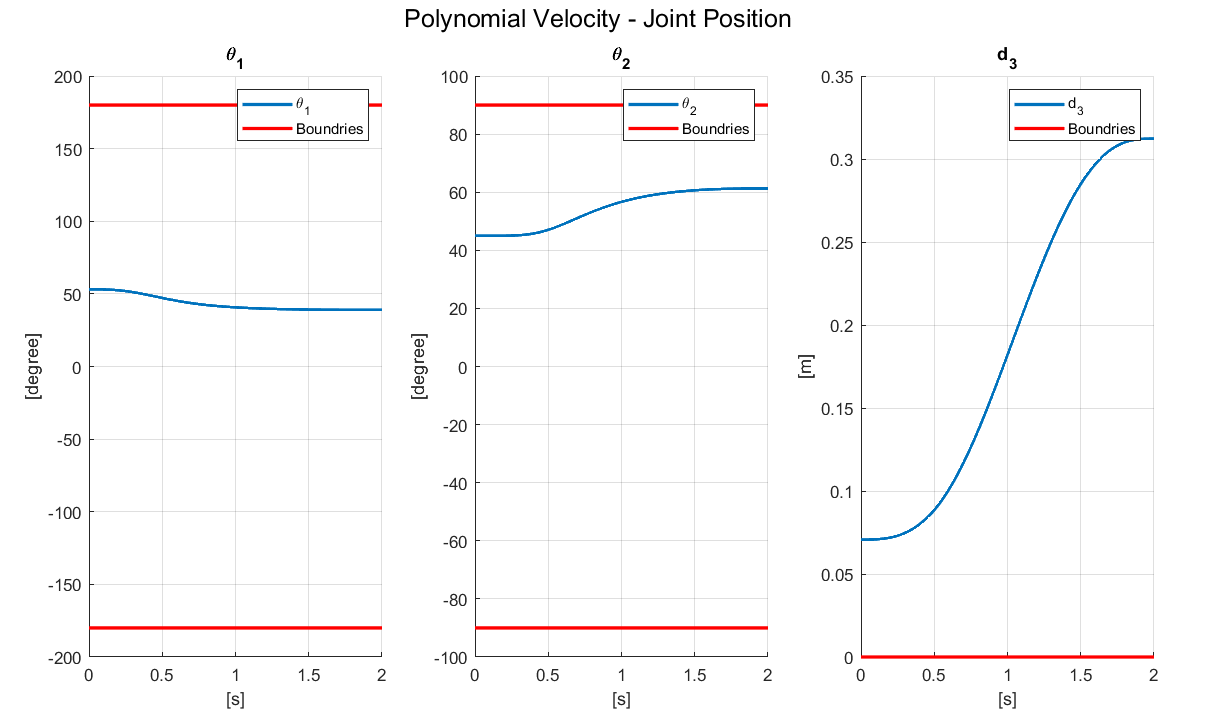


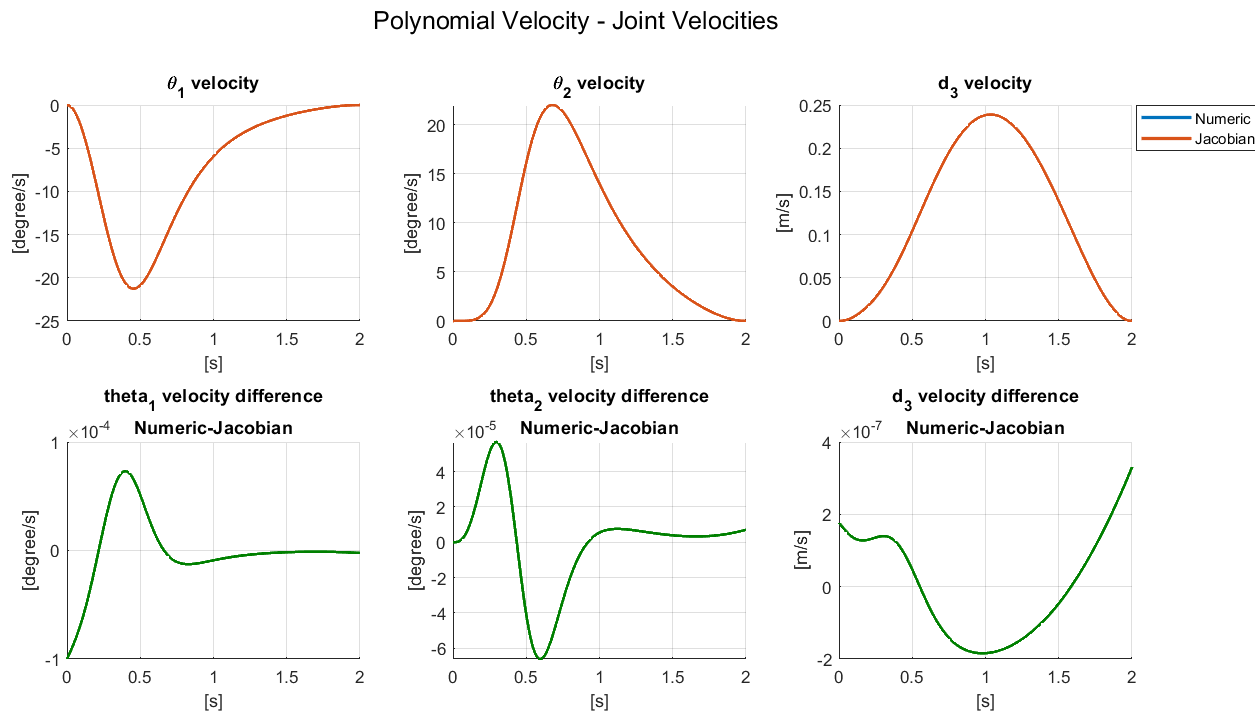
## Polynomial Profile - Minimum Jerk

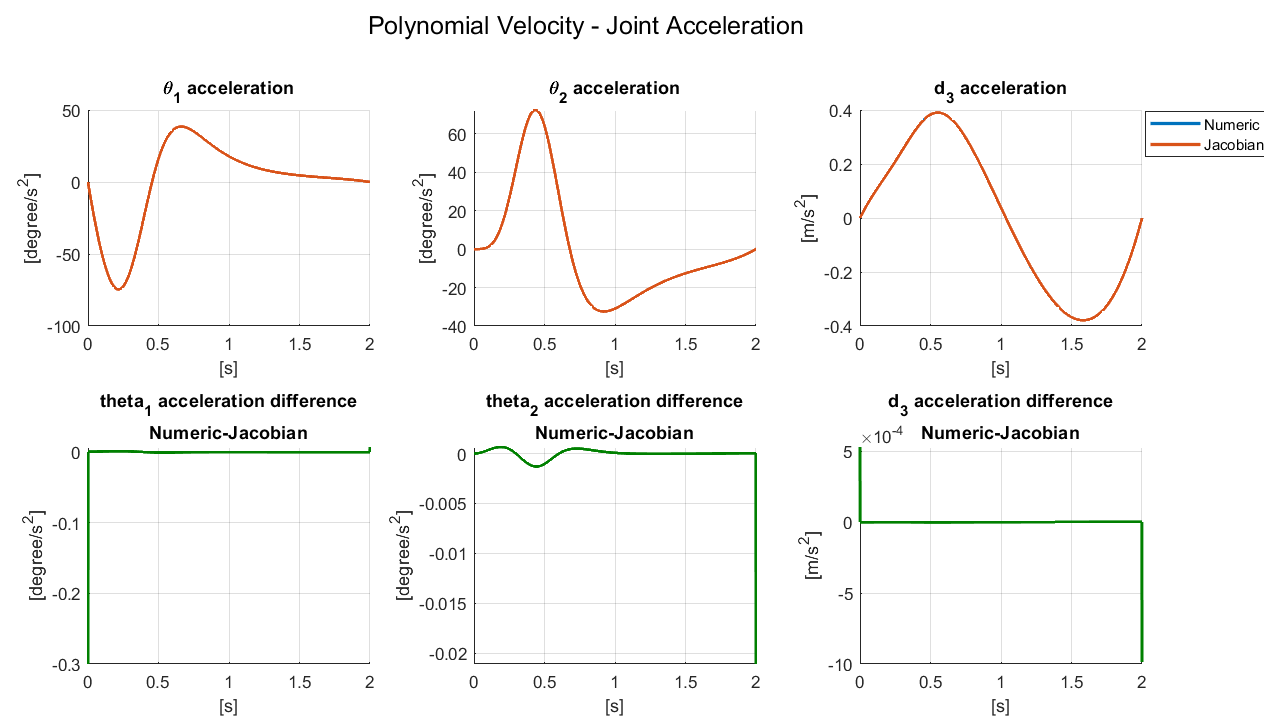
In their paper ‘A Minimum-Jerk Trajectory’ Kyriakopoulos et al. derivate a well-known 1-D polynomial trajectory which minimizes the motion’s jerk [1]. This trajectory also offers zero velocity and acceleration at the end points.

### Graphic Results

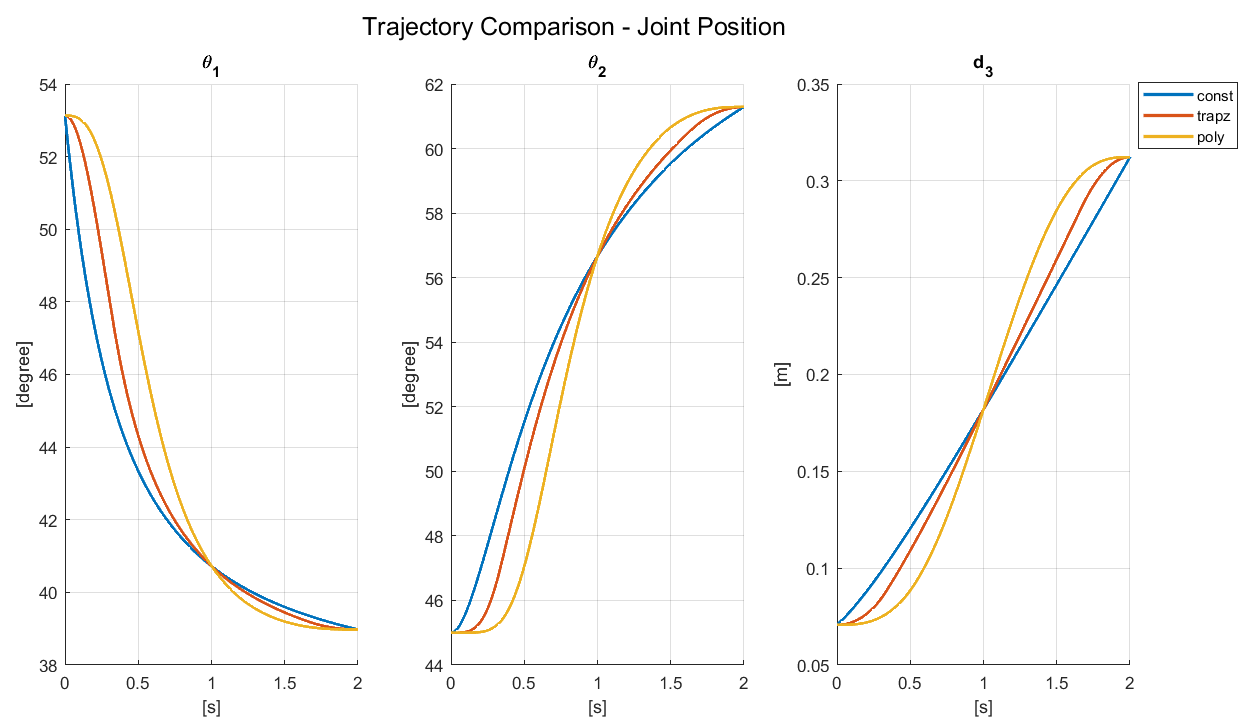


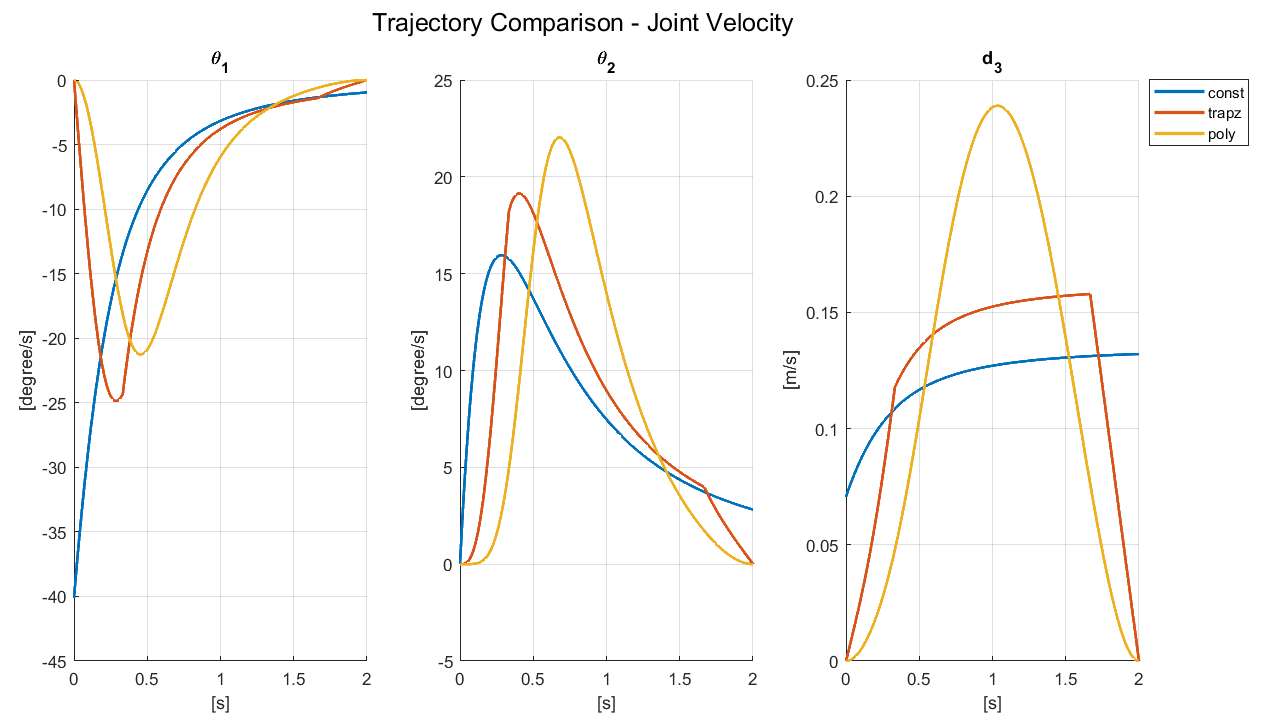


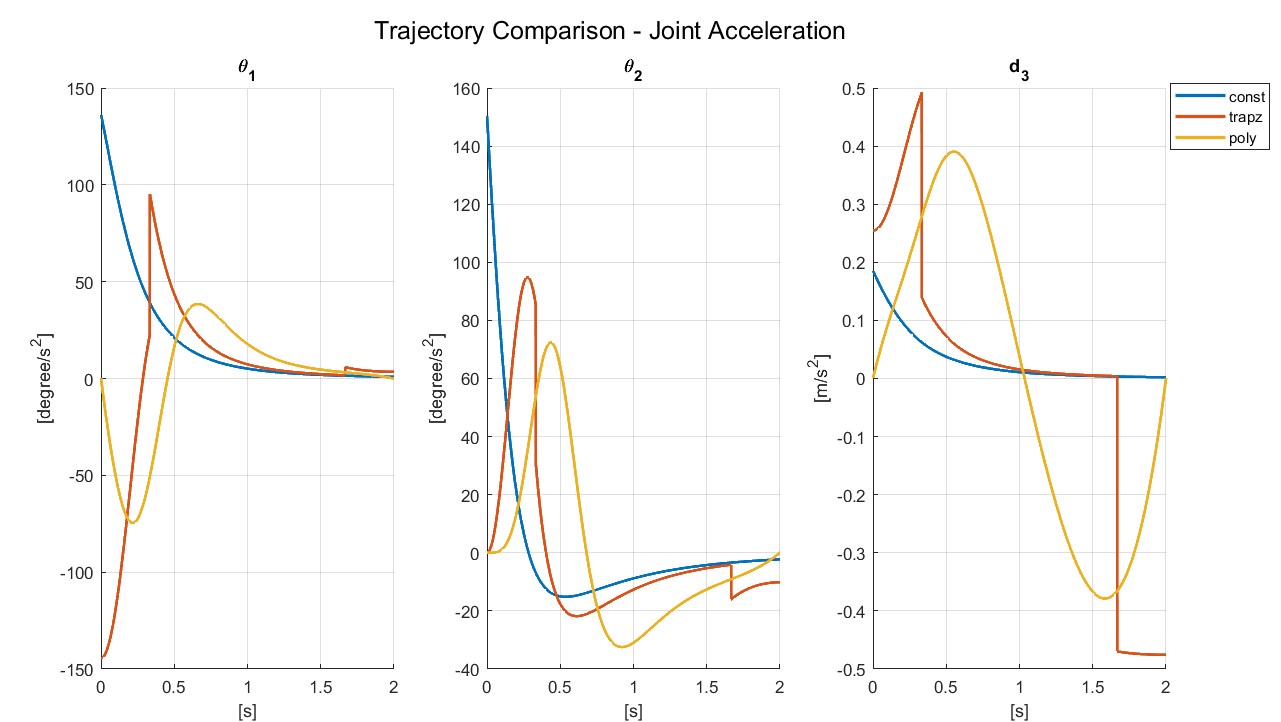




## Graphic Comparison Between Trajectories in Joint Space







# References

[1] K. J. Kyriakopoulos and G. N. Saridis, “Minimum jerk trajectory planning for robotic manipulators,” *Coop. Intell. Robot. Sp.*, vol. 1387, p. 159, 1991, doi: 10.1117/12.25421.