MA635P-Scientific Programming Laboratory

Cubic Hermite Interpolation for Trajectory Planning

Panchatcharam Mariappan[1]

[1]Associate Professor
Department of Mathematics and Statistics
IIT Tirupati, Tirupati

Deadline: March 16, 2026

MA635P
SCIENTIFIC COMPUTING LABORATORY

भारतीय प्रौद्योगिकी संस्थान तिरुपति
TIRUPATI

Cubic Hermite Interpolation for Trajectory Planning

Background

- In robotics and autonomous systems, smooth trajectory generation is essential.
- Trajectories must satisfy position and velocity constraints at waypoints.
- Cubic Hermite interpolation uses both function values and derivative values.
- Ensures C^1 continuity, suitable for motion planning.

Mathematical Formulation

Given two points x_0, x_1 with:

$$f(x_0) = y_0, \quad f'(x_0) = m_0$$
$$f(x_1) = y_1, \quad f'(x_1) = m_1$$

The cubic Hermite interpolant is:

$$H(x) = h_{00}(t)y_0 + h_{10}(t)(x_1 - x_0)m_0 + h_{01}(t)y_1 + h_{11}(t)(x_1 - x_0)m_1$$

where $t = \frac{x - x_0}{x_1 - x_0}$ and

$$h_{00} = 2t^3 - 3t^2 + 1, \quad h_{10} = t^3 - 2t^2 + t$$
$$h_{01} = -2t^3 + 3t^2, \quad h_{11} = t^3 - t^2$$

Objective

- To implement cubic Hermite interpolation for trajectory generation.
- To incorporate velocity constraints at waypoints.
- To ensure smooth motion between multiple segments.
- To compare with linear and cubic spline interpolation.

Steps to Solve

1. Code
   1.1 Implement cubic Hermite basis functions.
   1.2 Generate multi-segment trajectory.
2. Mathematical Report
   2.1 Derivation of cubic Hermite polynomial.
   2.2 Proof of C^1 continuity.
   2.3 Error analysis.
3. Comparison
   3.1 Compare with linear interpolation.
   3.2 Compare with cubic spline interpolation.

Advanced Investigation

1. Acceleration and Jerk Analysis
   1.1 Compute $H''(x)$ and analyze smoothness.
   1.2 Study motion comfort (jerk minimization).
2. Time Scaling
   2.1 Introduce time parameter t.
   2.2 Study effect of varying time intervals.
3. Extension to 2D/3D Trajectories

$$\mathbf{r}(t) = (x(t), y(t), z(t))$$

Apply Hermite interpolation component-wise.

Applications

1. Robot arm trajectory planning.
2. Autonomous vehicle path generation.
3. Drone motion planning.
4. Animation and computer graphics.

Deliverable

1. Python implementation of trajectory planner.
2. A report including:
   - Mathematical derivation
   - Continuity and smoothness analysis
   - Acceleration and jerk study
   - Comparison with spline interpolation
3. Visualization of trajectory, velocity and acceleration profiles.

Problem Statement: Robot Arm Trajectory Planning

Consider a planar 2-DOF (or 3-DOF) robotic arm required to move its end-effector through a sequence of waypoints:

$$\mathbf{P}_0, \mathbf{P}_1, \ldots, \mathbf{P}_n$$

within specified time intervals.

- At each waypoint, desired position and velocity are prescribed.
- The trajectory must be smooth, physically feasible, and suitable for real-time execution.

Problem Statement: Drone Motion Planning

An autonomous drone is required to travel through a sequence of 3D waypoints:
$\mathbf{P}_0, \mathbf{P}_1, \ldots, \mathbf{P}_n$
where $\mathbf{P}_i = (x_i, y_i, z_i)$, within specified time intervals.

- Desired position and velocity vectors are prescribed at each waypoint.
- The trajectory must ensure smooth flight, stability, and physical feasibility.
- Sudden changes in acceleration (jerk) must be minimized.

1. Construct piecewise cubic Hermite interpolants for:

$$\mathbf{r}(t) = (x(t), y(t), z(t))$$

   satisfying:

$$\mathbf{r}(t_i) = \mathbf{P}_i, \quad \dot{\mathbf{r}}(t_i) = \mathbf{v}_i$$

2. Compute velocity and acceleration profiles.
3. Analyze trajectory smoothness and compare with linear interpolation.

Thanks

Doubts and Suggestions

panch.m@iittp.ac.in

MA635P-Scientific Programming Laboratory

Cubic Hermite Interpolation for Trajectory Planning

Panchatcharam Mariappan[1]

[1]Associate Professor
Department of Mathematics and Statistics
IIT Tirupati, Tirupati

Deadline: March 16, 2026

भारतीय प्रौद्योगिकी संस्थान तिरुपति
TIRUPATI