Constrained Trajectories

MEAM 620

Minimum Jerk Segment

Given an n'th order system $x^{(n)} = u$, minimizing $\int u^2 dt$ led to (2n-1)'th order polynomials.

Example: For a 3rd order system, the 'minimum jerk' polynomials are quintic.

$$p(t) = t^5 c_5 + t^4 c_4 + t^3 c_3 + t^2 c_2 + t^1 c_1 + c_0$$

At any time t, we can think of the position as the product of a row vector of time-coefficients [T(t)] and column vector of polynomial coefficients [X].

$$[T(t)] = \begin{bmatrix} t^5 & t^4 & t^3 & t^2 & t^1 & 1 \end{bmatrix}$$

$$[X] = \begin{bmatrix} c_5 & c_4 & c_3 & c_2 & c_1 & c_0 \end{bmatrix}^T$$

$$p(t) = [T(t)][X]$$

$$p(t) = \begin{bmatrix} t^5 & t^4 & t^3 & t^2 & t^1 & 1 \end{bmatrix} \begin{bmatrix} c_5 \\ c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix}$$

Position Boundary Constraints

Enforcing the position at time t=0 and time t=1 yields two position constraints.

We write this as two rows in a constraint matrix A.

$$[T(t)] = \begin{bmatrix} t^5 & t^4 & t^3 & t^2 & t^1 & 1 \end{bmatrix}$$

$$[X] = \begin{bmatrix} c_5 & c_4 & c_3 & c_2 & c_1 & c_0 \end{bmatrix}^T$$

$$p(0) = [T(0)][X]$$

$$p(t_1) = [T(t_1)][X]$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ t_1^5 & t_1^4 & t_1^3 & t_1^2 & t_1^1 & 1 \end{bmatrix} \begin{bmatrix} c_5 \\ c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix} = \begin{bmatrix} p(0) \\ p(t_1) \end{bmatrix}$$

$$[A][X] = b$$

No unique solution; we have 2 constraints and 6 unknowns.

Full Boundary Constraints

The state of our third order system $x^{(3)} = u$ consists of {x, $\dot{x}$, $\ddot{x}$}. Typically all must satisfy boundary conditions.

The derivatives of p are also polynomials, so we can write constraints on the derivatives in the same way.

$$p(t) = t^5 c_5 + t^4 c_4 + t^3 c_3 + t^2 c_2 + t^1 c_1 + c_0$$
$$\dot{p}(t) = 5t^4 c_5 + 4t^3 c_4 + 3t^2 c_3 + 2t^1 c_2 + c_1$$
$$\ddot{p}(t) = 20t^3 c_5 + 12t^2 c_4 + 6t^1 c_3 + 2c_2$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ t_1^5 & t_1^4 & t_1^3 & t_1^2 & t_1^1 & 1 \\ 5t_1^4 & 4t_1^3 & 3t_1^2 & 2t_1^1 & 1 & 0 \\ 20t_1^3 & 12t_1^2 & 6t_1^1 & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} c_5 \\ c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix} = \begin{bmatrix} p(0) \\ \dot{p}(0) \\ \ddot{p}(0) \\ p(t_1) \\ \dot{p}(t_1) \\ \ddot{p}(t_1) \end{bmatrix}$$

$$[A][X] = b$$

At each end point get n=3 constraint equations.

Unique solution: 6 constraints and 6 unknowns.

Minimum Jerk Spline

With m segments, we have 6m unknowns.

$$p_1(t) = t^5 c_{1,5} + t^4 c_{1,4} + t^3 c_{1,3} + t^2 c_{1,2} + t^1 c_{1,1} + c_{1,0}$$

$$\dots$$

$$p_m(t) = t^5 c_{m,5} + t^4 c_{m,4} + t^3 c_{m,3} + t^2 c_{m,2} + t^1 c_{m,1} + c_{m,0}$$

If segment k has duration t_k, it is convenient to define each segment on the interval $[0, t_k]$.

The boundary conditions still only give 6 constraints (first point, last point)

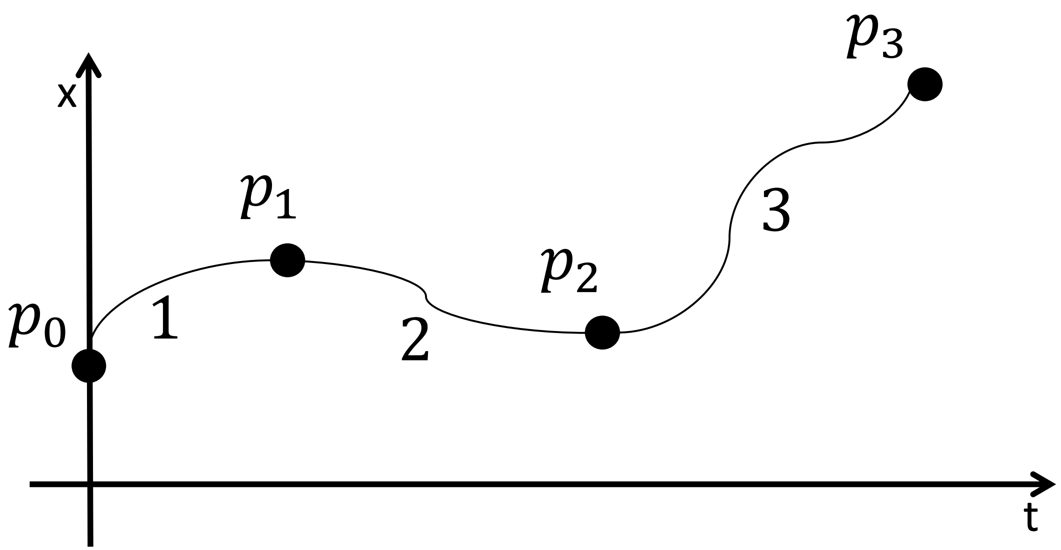
$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & t_2^5 & t_2^4 & t_2^3 & t_2^2 & t_2^1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5t_2^4 & 4t_2^3 & 3t_2^2 & 2t_2^1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 20t_2^3 & 12t_2^2 & 6t_2^1 & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} c_{1,5} \\ c_{1,4} \\ c_{1,3} \\ c_{1,2} \\ c_{1,1} \\ c_{1,0} \\ c_{2,5} \\ c_{2,4} \\ c_{2,3} \\ c_{2,2} \\ c_{2,1} \\ c_{2,0} \end{bmatrix} = \begin{bmatrix} p_1(0) \\ \dot{p}_1(0) \\ \ddot{p}_1(0) \\ p_2(t_2) \\ \dot{p}_2(t_2) \\ \ddot{p}_2(t_2) \end{bmatrix}$$

Waypoint Position Constraints

Forcing the spline to go through (m-1) waypoints adds 2(m-1) constraints.

"Segment k ends at position p_k at time t_k."

"Segment $k + 1$ starts at position p_k at time 0."



$$\begin{bmatrix} t_1^5 & t_1^4 & t_1^3 & t_1^2 & t_1^1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_{1,5} \\ c_{1,4} \\ c_{1,3} \\ c_{1,2} \\ c_{1,1} \\ c_{1,0} \\ c_{2,5} \\ c_{2,4} \\ c_{2,3} \\ c_{2,2} \\ c_{2,1} \\ c_{2,0} \end{bmatrix} = \begin{bmatrix} p_1 \\ p_1 \end{bmatrix}$$

Necessary Continuity Constraints

For a physical system $x^{(n)} = u$, the (n-1)'th derivative must be continuous everywhere.

"The derivative at the end of segment k ***equals*** the derivative at the start of segment k+1."

Adds another m-1 constraints for each required derivative.

$$\begin{bmatrix} 5t_1^4 & 4t_1^3 & 3t_1^2 & 2t_1^1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 20t_1^3 & 12t_1^2 & 6t_1^1 & 2 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 \end{bmatrix} \begin{bmatrix} c_{1,5} \\ c_{1,4} \\ c_{1,3} \\ c_{1,2} \\ c_{1,1} \\ c_{1,0} \\ c_{2,5} \\ c_{2,4} \\ c_{2,3} \\ c_{2,2} \\ c_{2,1} \\ c_{2,0} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Summary so far (5'th order):

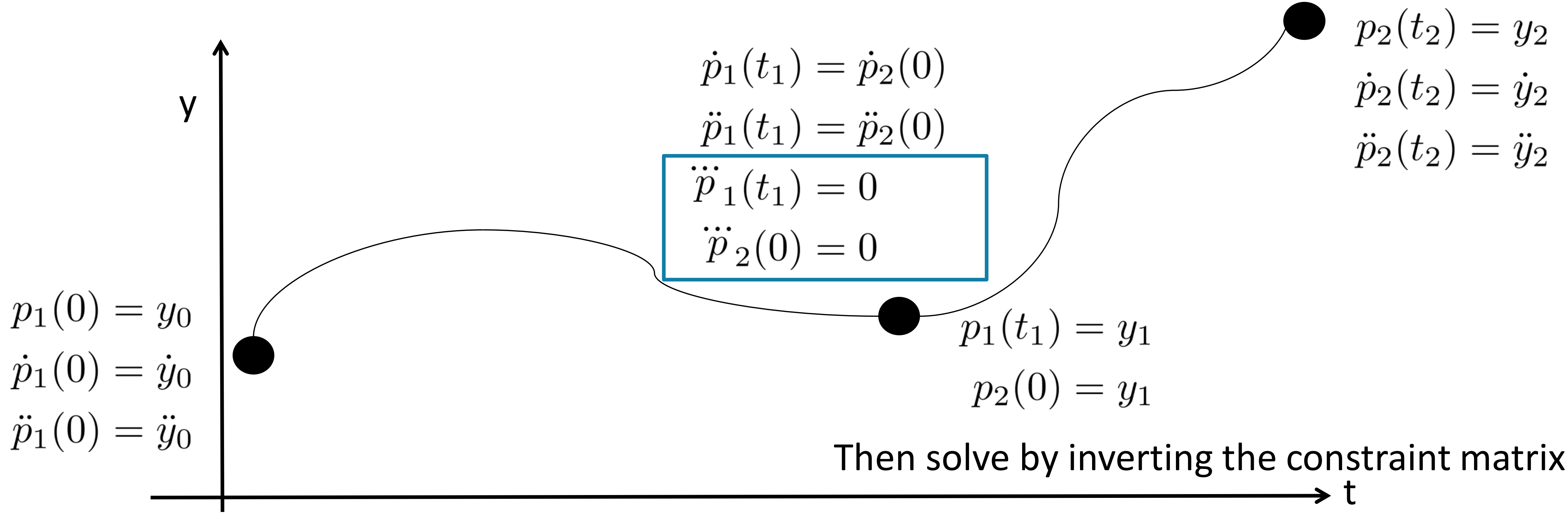
- 6m unknowns
- 2m position constraints (two for each segment)
- 2(m-1) derivative continuity constraints (two at each midpoint)
- 4 derivative boundary constraints (two at each endpoint) -> *2(m-1) remaining*

Solution #0

Boundary and continuity constraints up to acceleration don't fully constrain the solution.

To fully constrain the solution, we need another 2 constraints per midpoint.

Idea: Arbitrarily assign the jerk to 0 at each midpoint.

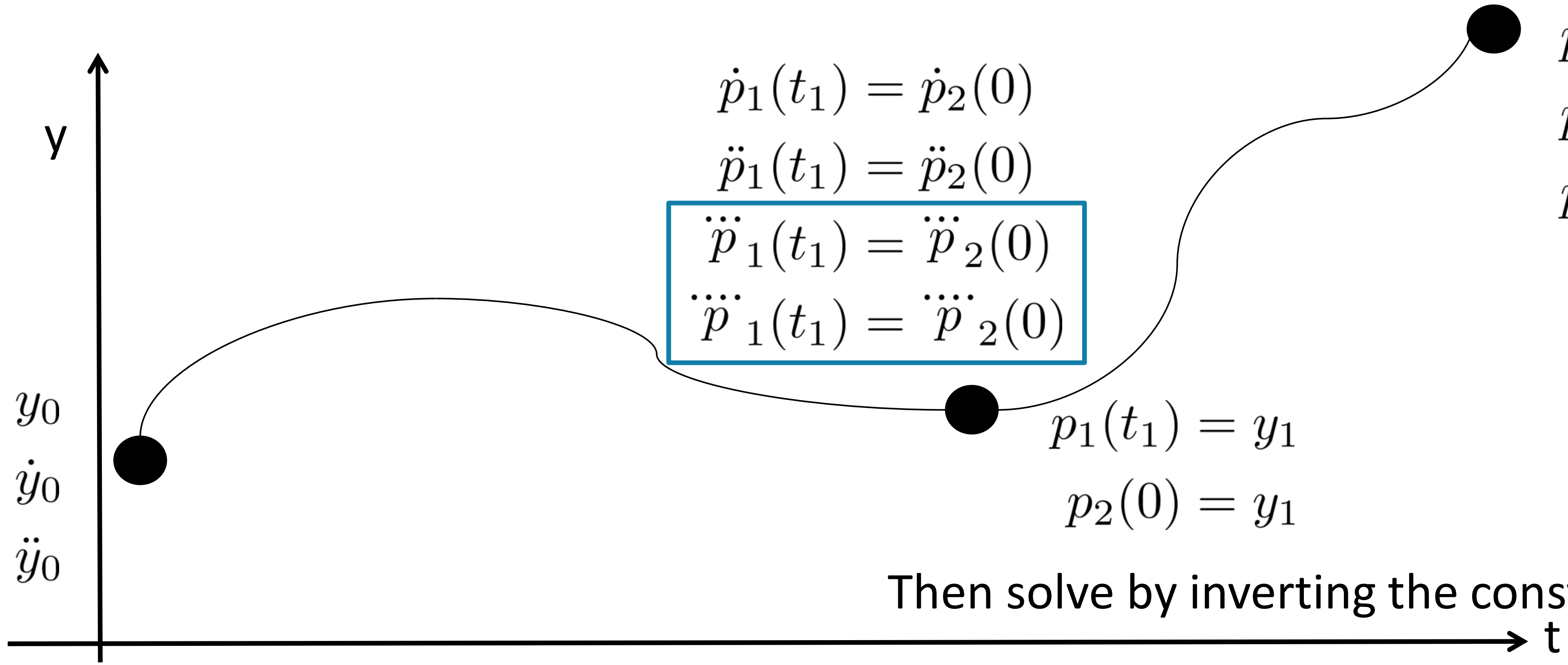


Then solve by inverting the constraint matrix.

Solution #1: Fully Constrained Spline

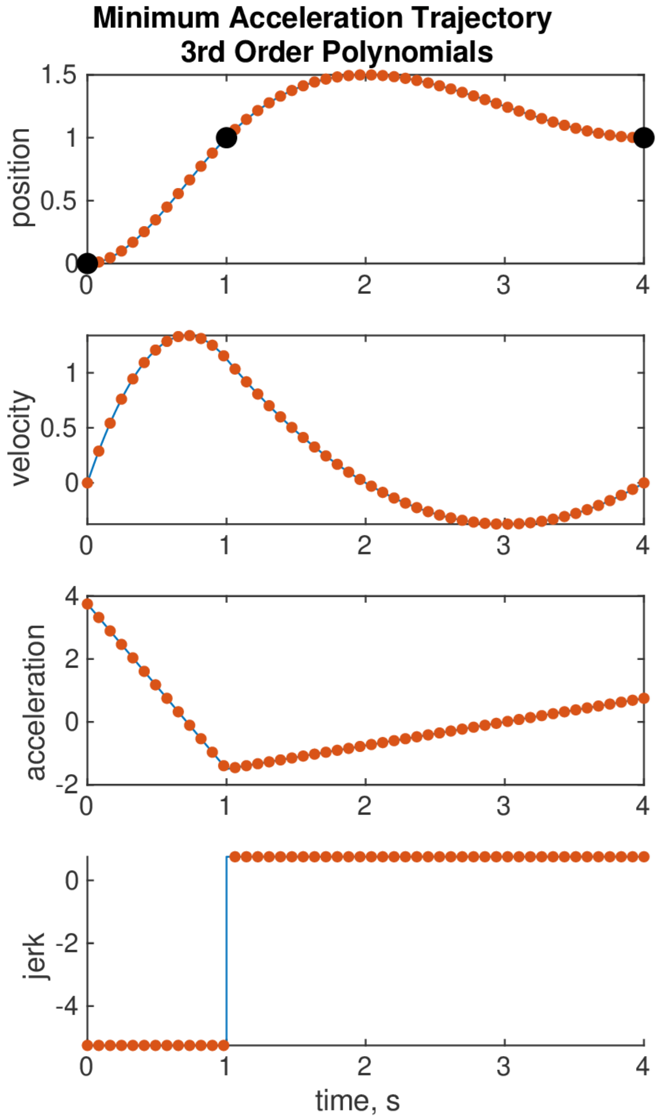
That was pretty arbitrary, we don’t really need zero jerk at the midpoints.

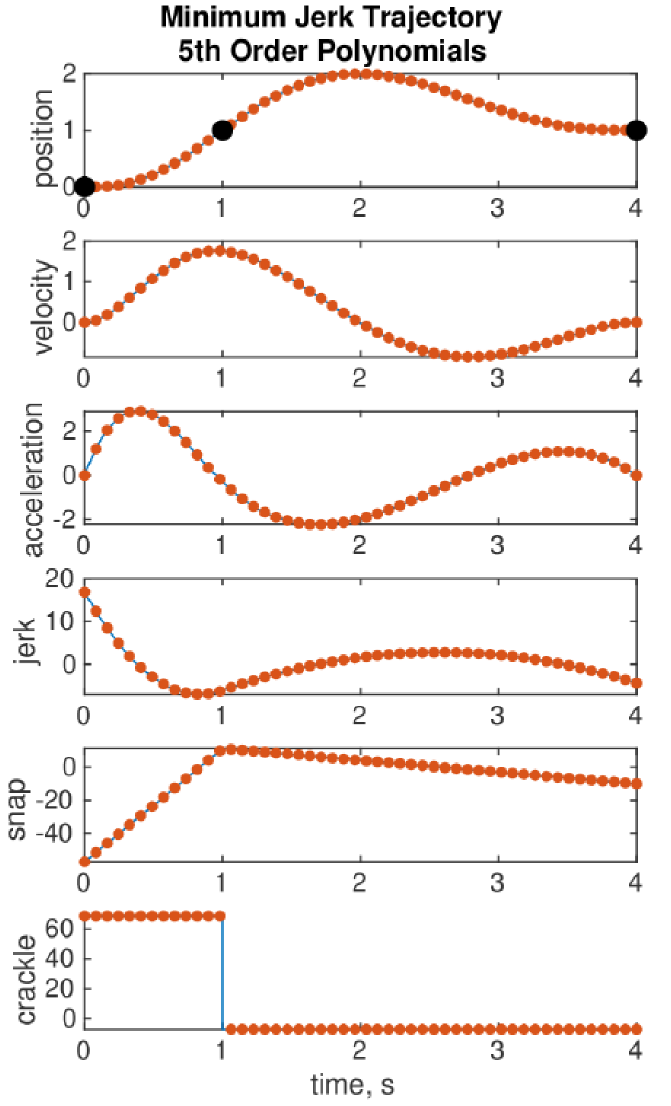
Idea: Add enough additional *continuity* constraints to force a unique solution.



Then solve by inverting the constraint matrix.

Example with Fully Constrained Spline





Solution #2: Optimize Cost

We don’t really need those extra degrees of continuity either.

Idea: Find the polynomial that minimizes the original cost (the integral of u^2).

We already have the polynomial for the jerk.

$$p_1(t) = t^5 c_{1,5} + t^4 c_{1,4} + t^3 c_{1,3} + t^2 c_{1,2} + t^1 c_{1,1} + c_{1,0}$$
$$\dddot{p}_1(t) = 60t^2 c_5 + 24t^1 c_4 + 6c_3$$

If we square $\dddot{p}_1(t)$, integrate, and evaluate from t=[0,t_1] we get the explicit cost.

$$\text{cost} = \int_0^{t_1} (\dddot{p}_1(t))^2 dt = \int_0^{t_1} (60t^2 c_5 + 24t^1 c_4 + 6c_3)^2 dt$$

The cost is a function of the coefficients, and can be put into a quadratic form.

$$\text{cost} = c^T H c + f^T c$$

Know matrix H and vector f, don’t know polynomial coefficients c.

Quadratic Program

The problem of optimizing this cost subject to the constraints we've already found is called a "Quadratic program."

Find "decision variables" X that

$$\text{minimize } X^T H X + f^T X$$

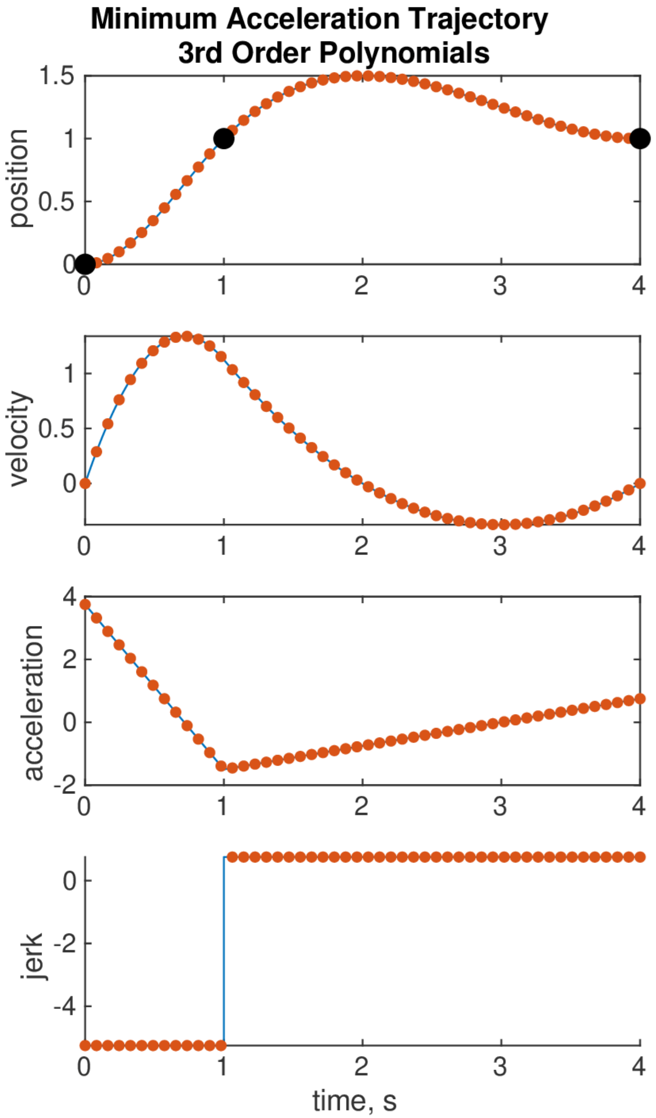
minimize this cost, and

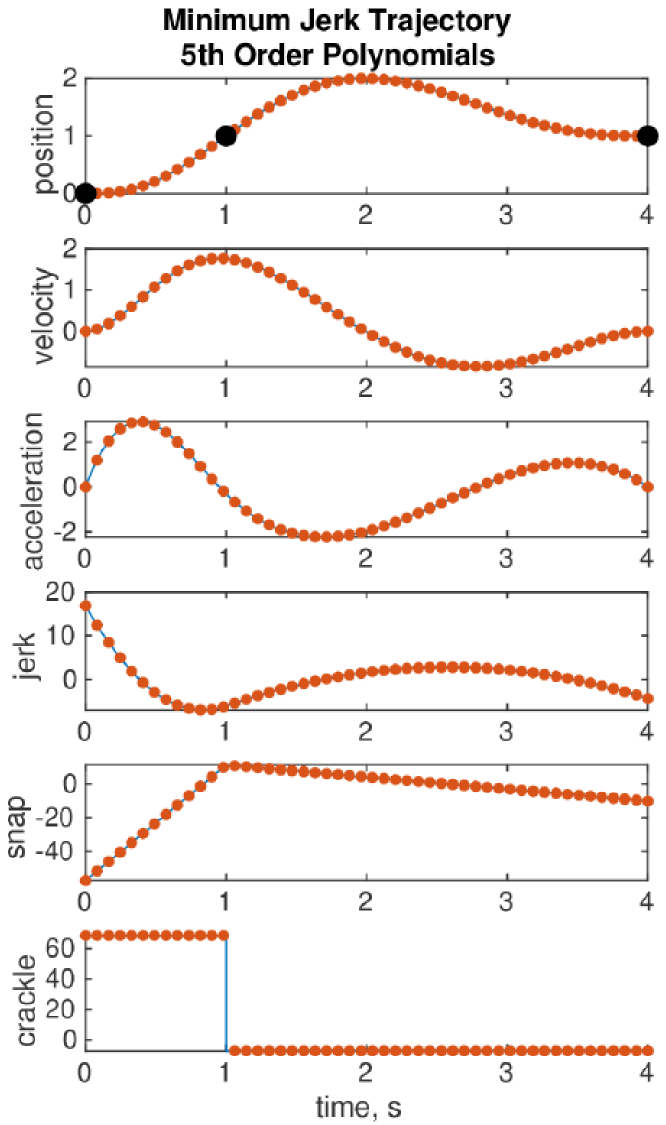
$$\text{s.t } AX = b$$

satisfy these constraints

Quadratic programs are a special class of convex optimization problems that are relatively fast to solve with specialized solvers like CVXOPT.

Example with Optimized Cost





Solution #3: Inequality Coordinate Constraints

We can use extra degrees of freedom to satisfy other constraints.

Idea: Inequality constraints such as ‘position below 5 m’ or ‘velocity under 5 m/s’ can be enforced at additional sample points.

Choose (lots) of sample times s, and for each sample time add a constraint.

$$p(t) = t^5 c_5 + t^4 c_4 + t^3 c_3 + t^2 c_2 + t^1 c_1 + c_0$$

$$p(s) \leq 10$$ *“stay under the 10 m ceiling”*

$$\dot{p}(s) \leq 3$$ *“keep velocity below 3 m/s”*

$$-\dot{p}(s) \leq 3$$ *“keep velocity above -3 m/s”*

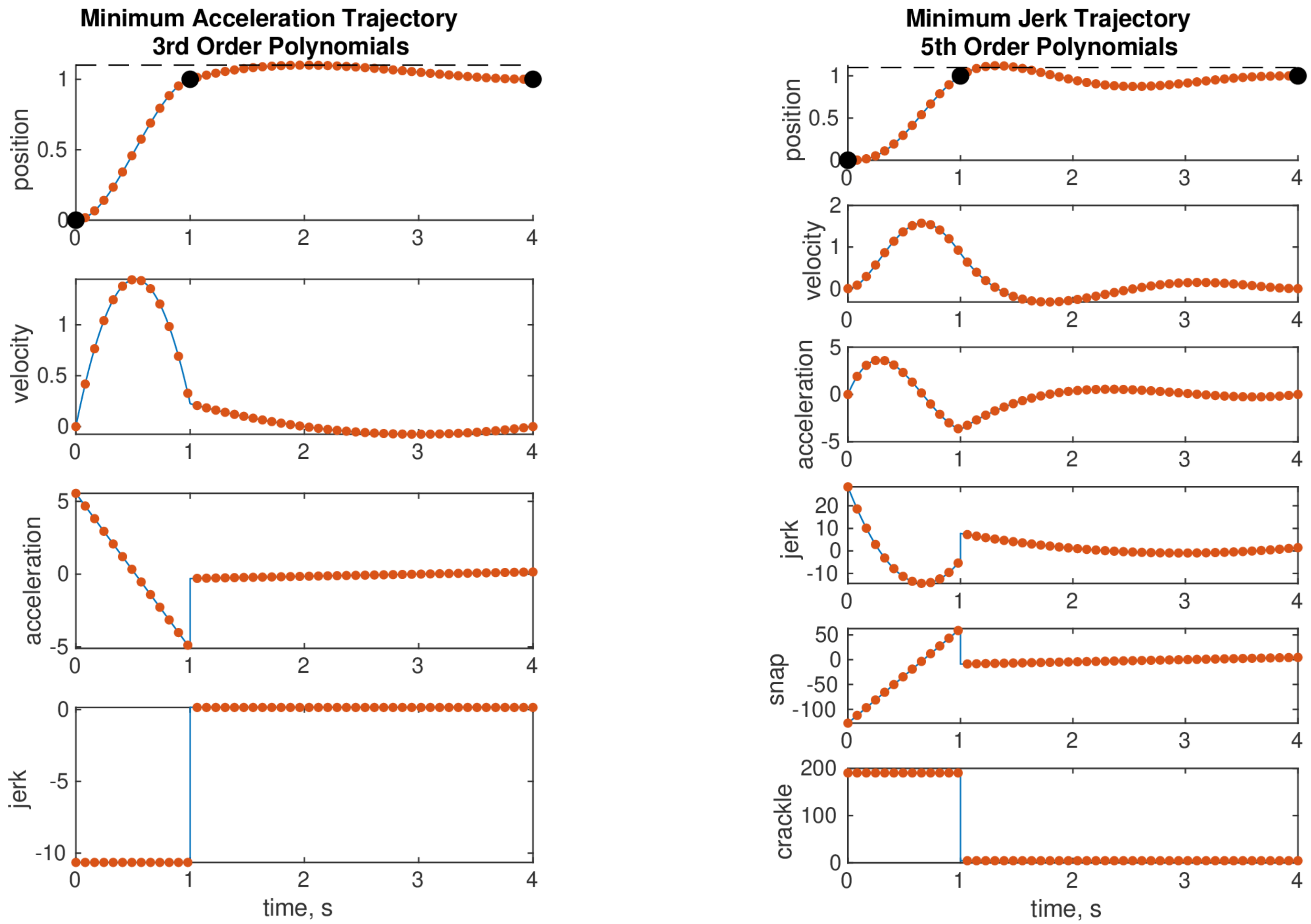
$$\text{minimize } X^T H X + f^T X$$

$$\text{s.t } A_{\text{eq}} X = b_{\text{eq}}$$

$$AX \leq b$$

Can express these with a constraint matrix, just like all the previous examples.

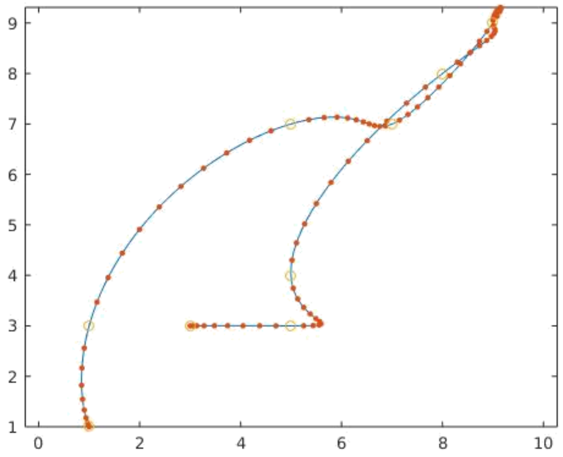
Example with a Constraint

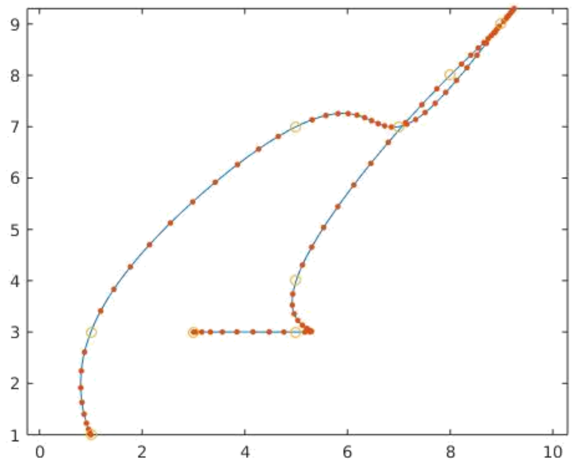


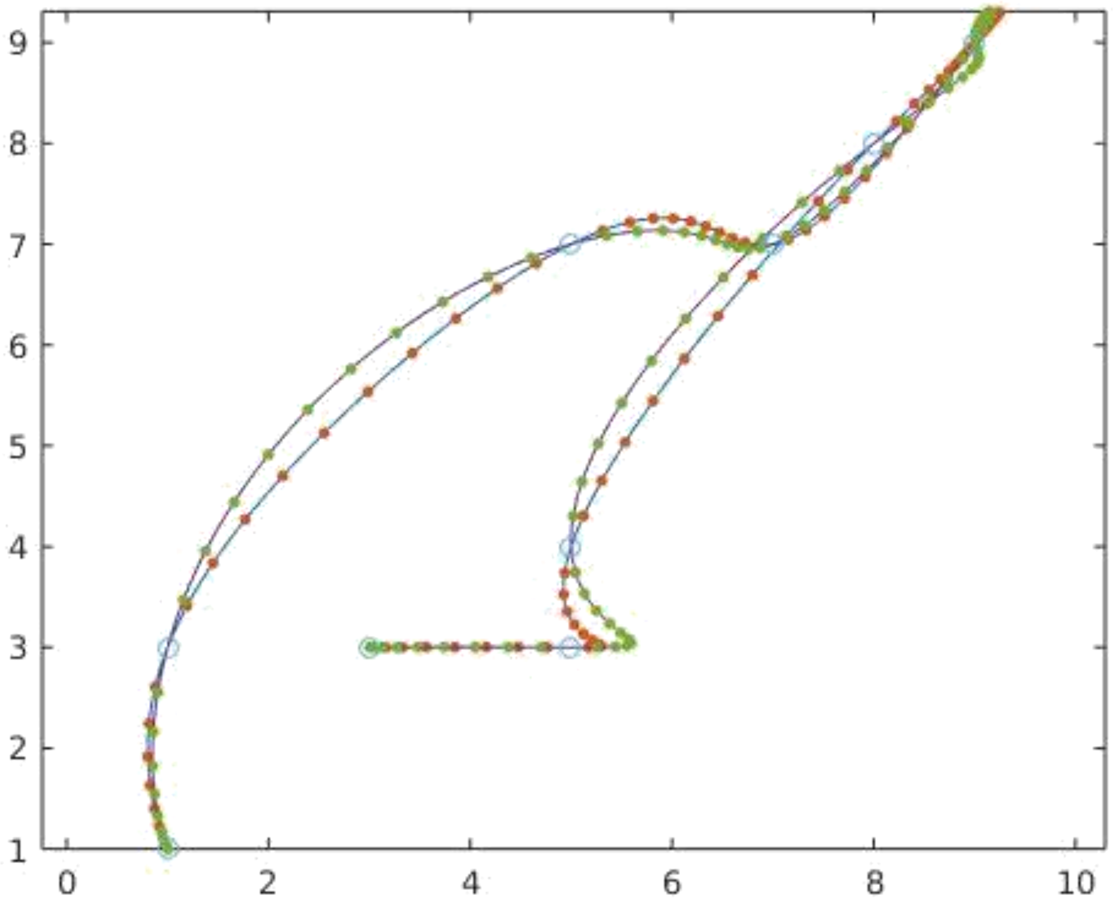
Corridor Constraints

A common requirement is to stay 'close' to the original path.

Idea: Add corridor constraints that relate x,y positions to stay near the original line.







For details see Mellinger and Kumar 2011, "[Minimum Snap Trajectory Generation and Control for Quadrotors]"

Implementation Notes

In big multi-segment problems, numerical stability becomes important.

Consider the intermediate values required to compute a 7th order spline

$$p(t) = t^7 c_7 + t^6 c_6 + t^5 c_5 + t^4 c_4 + t^3 c_3 + t^2 c_2 + t^1 c_1 + c_0$$

When defining constraints:

- Scale the time by α for each segment to run from [0, 1]
- Scale the displacement by β for each segment to run from [0, 1]
- Solve the nondimensional problem, then scale the solution back to real units.

Special cases can be scaled and re-used, eg. zero derivative boundary conditions.

- https://www.youtube.com/watch?v=geqip_0Vjec (hula hoop example exploits scaling for speed)

For details see Mellinger and Kumar 2011, "Minimum Snap Trajectory Generation and Control for Quadrotors"