Creating a trajectory: Interpolation from A to B

By Jonathan Woolfrey

There are two ways to approach moving a robot arm: joint space, or end-effector/task space. In specifying end-effector motion, we still have to resolve the necessary joint velocities to achieve the task. This can get complicated. For now, we'll look at how to smoothly transition a joint from one position to another.

We don't want our joint motors to be working too hard, so if the joint positions transition smoothly, then the joint velocities will also be smooth.

We can define an intermediate joint position $q(s)$ between two joint angles q_A and q_B as a function of a smooth scalar s.

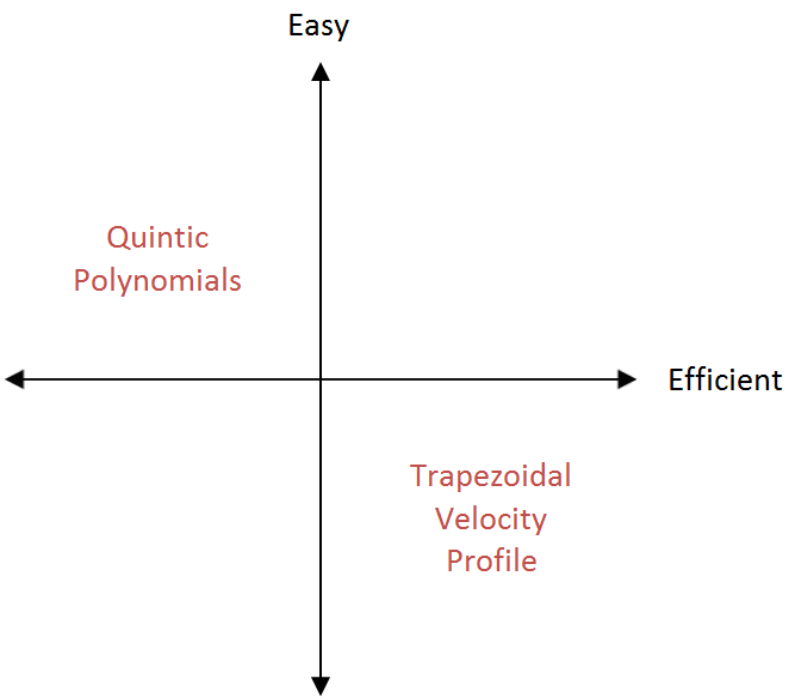
$$q(s) = (1-s)q_A + sq_B \qquad s = [0,1] \tag{1}$$

It can be seen then that:

$$q(s) = \begin{cases} q_A & \text{when s} = 0 \\ q_B & \text{when s} = 1 \end{cases}$$

There are two main approaches for achieving this:

1. Quintic Polynomial: an easy, but inefficient method.
2. Trapezoidal Velocity Profile: a more complex, but much more efficient method.



1. Quintic Polynomial

Let s be a 5th order polynomial function of time:

$$s = At^5 + Bt^4 + Ct^3 + Dt^2 + Et + F \qquad ❷$$
$$\dot{s} = 5At^4 + 4Bt^3 + 3Ct^2 + 2Dt + E$$
$$\ddot{s} = 20At^3 + 12Bt^2 + 6Ct + 2D$$

When $t = 0$

$$s_0 = F$$
$$\dot{s}_0 = E$$
$$\ddot{s}_0 = 2D$$

And at some final time T

$$s_f = AT^5 + BT^4 + CT^3 + DT^2 + ET + F$$
$$\dot{s}_f = 5AT^4 + 4BT^3 + 3CT^2 + 2DT + E$$
$$\ddot{s}_f = 20AT^3 + 12BT^2 + 6CT + 2D$$

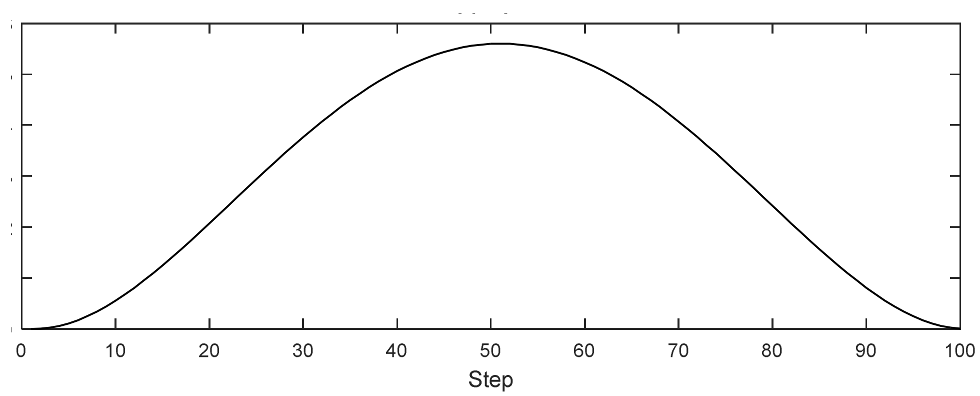
We pack these in a matrix for easier handling: $$\begin{bmatrix} s_0 \\ \dot{s}_0 \\ \ddot{s}_0 \\ s_f \\ \dot{s}_f \\ \ddot{s}_f \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ T^5 & T^4 & T^3 & T^2 & T & 1 \\ 5T^4 & 4T^3 & 3T^2 & 2T & 1 & 0 \\ 20T^3 & 12T^2 & 6T & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \\ E \\ F \end{bmatrix}$$

Then solve for the polynomial coefficients: $$\begin{bmatrix} A \\ B \\ C \\ D \\ E \\ F \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ T^5 & T^4 & T^3 & T^2 & T & 1 \\ 5T^4 & 4T^3 & 3T^2 & 2T & 1 & 0 \\ 20T^3 & 12T^2 & 6T & 2 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} s_0 \\ \dot{s}_0 \\ \ddot{s}_0 \\ s_f \\ \dot{s}_f \\ \ddot{s}_f \end{bmatrix} \qquad ❸$$

Then plug ❸ back in to ❷, and ❷ back in to ❶.

Fortunately, Robot Toolbox can solve this for us using, `jtraj(q1,q2,stepCount)`.

The joint velocity profiles will look something like this:

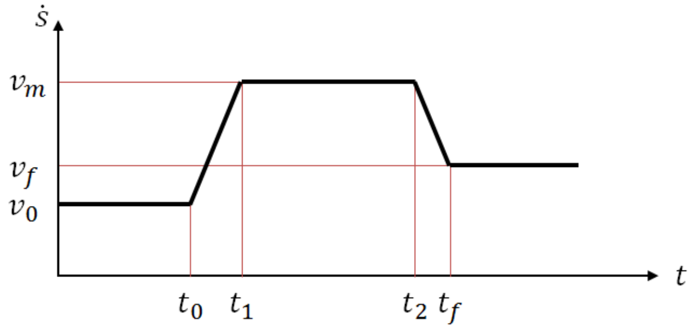


2. Trapezoidal Velocity Profile

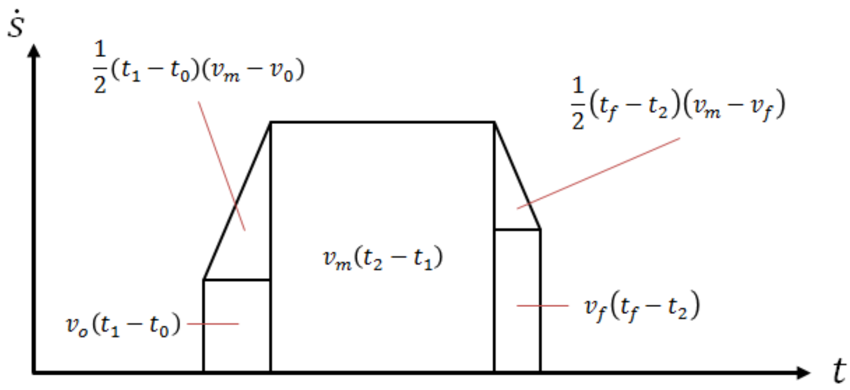
The velocity profile using Quintic Polynomials peaks for a brief instance in time which leads to two problems:

1. Not making full use of the motor's speed capabilities
2. The motor has to use a lot of power accelerating and decelerating. This is not energy efficient.

Instead, if we explicitly define the velocity $\dot{s}$ to follow a trapezoidal shape, as shown:



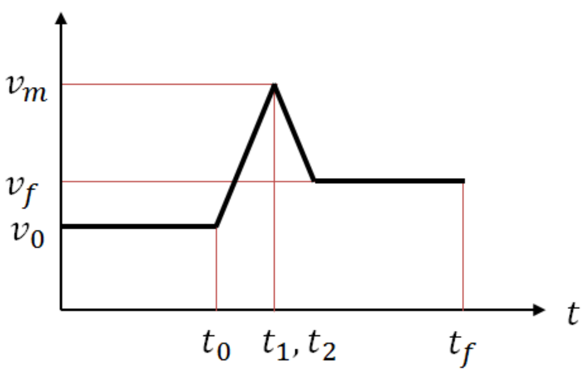
The area under the trapezoid defines the total distance s we have travelled in that time period.



Using some geometry and algebra we can define:

$$
s(t) \begin{cases} 0 & \text{for } t = t_0 \\ (v_o(t - t_0) + ½a(t - t_0)^2)s_f^{-1} & \text{for } t_0 < t < t_1 \\ \left(v_o(t_1 - t_0) + ½a(t_1 - t_0)^2 + v_m(t - t_1)\right)s_f^{-1} & \text{for } t_1 \leq t < t_2 \\ (v_o(t_1 - t_0) + ½a(t_1 - t_0)^2 + v_m(t_2 - t_1) + v_m(t - t_2) - ½a(t - t_2)^2)s_f^{-1} & \text{for } t_2 \leq t < t_f \\ 1 & \text{for t} \geq t_f \end{cases}
$$

We also have to be careful when selecting the parameters of the velocity profile, or we could end up in a situation like this:



Robot Toolbox can also help us generate trapezoidal profiles, albeit with a little extra work.

```
s = lspb(0,1,steps);
qMatrix = nan(steps,6);
for i = 1:steps
    qMatrix(i,:) = (1-s(i))*q1 + s(i)*q2;
end
```