Homework Assignment 4: Trajectory Planning

Dynamics Of Non Linear Robotic Systems

Ilia Sevostianov

November 25, 2019

Contents

Code

The code with comments you can find here.

Task 1

Compute the joint trajectory from $q(0) = 1$ to $q(2) = 4$ with null initial and final velocities:

1. Use polynomial position profile
2. Use trapezoidal velocity profile.
3. Use velocity profile of the type $\dot{q}(t) = a(b + sin(ct))$.

The problem of trajectory planning is to find a trajectory that connects an initial to a final configuration while satisfying other specified constraints at the endpoints (e.g., velocity and/or acceleration constraints).

Polynomial Position Profile

We have four constraints to satisfy:

- $q(0) = 1$
- $q(2) = 4$
- $\dot{q}(0) = 0$
- $\dot{q}(2) = 0$

So, we require a polynomial with four independent coefficients that can be chosen to satisfy these constraints. Thus we consider a cubic trajectory of the form:

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \tag{1}$$

Then, velocity $\dot{q}(t)$ and accelerations $\ddot{q}(t)$ equations are:

$$\dot{q}(t) = a_1 + 2a_2 t + 3a_3 t^2 \tag{2}$$

$$\ddot{q}(t) = 2a_2 t + 6a_3 t \tag{3}$$

Combining equations 1 and 2 with the four constraints yields four equations in four unknowns:

$$q(0) = a_0 + a_1 t_0 + a_2 t_0^2 + a_3 t_0^3 \tag{4}$$

$$\dot{q}(0) = a_1 + 2a_2t_0 + 3a_3t_0^2 \tag{5}$$

$$q(2) = a_0 + a_1t_f + a_2t_f^2 + a_3t_f^3 \tag{6}$$

$$\dot{q}(2) = a_1 + 2a_2t_f + 3a_3t_f^2 \tag{7}$$

These four equations can be combined into a single matrix equation

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 \\ 0 & 1 & 2t_0 & 3t_0^2 \\ 1 & t_f & t_f^2 & t_f^3 \\ 0 & 1 & 2t_f & 3t_f^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} q(0) \\ \dot{q}(0) \\ q(2) \\ \dot{q}(2) \end{bmatrix} \tag{8}$$

The equation 8 can be written as:

$$Ma = b \tag{9}$$

Thus, we can find the coefficients of polynomial trajectory.

$$a = M^{-1}b \tag{10}$$

That's the main idea how to solve such trajectory planning tasks. And the polynomial equation's power depends on amount of constraints. All the coefficients can be found through equation 10.

So, polynomial position profile for given constraints is:

$$q(t) = -\frac{3t^3}{4} + \frac{9t^2}{4} + 1 \tag{11}$$

graphs of trajectory you can see in the figure 1

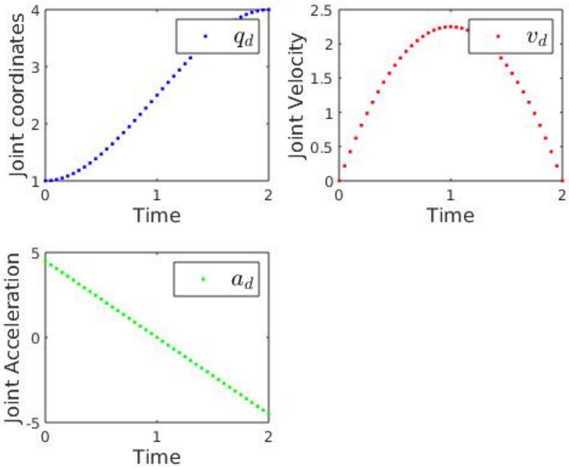

Figure 1: the joint trajectory from $q(0) = 1$ to $q(2) = 4$ (polynomial)

Trapezoidal Velocity Profile

This type of trajectory is appropriate when a constant velocity is desired along a portion of the path. The LSPB trajectory is such that the velocity is initially "ramped up" to its desired value and then "ramped down" when it approaches the goal position.

The first part from time t_0 to time t_b is a quadratic polynomial. This results in a linear "ramp" velocity. At time t_b , called the **blend time**, the trajectory switches to a linear function. This corresponds to a constant velocity. Finally, at time $t_f - t_b$ the trajectory switches once again, this time to a quadratic polynomial so that the velocity is linear.

Let's say, we have the graph for LSPB trajectory (velocity graph, figure 2).

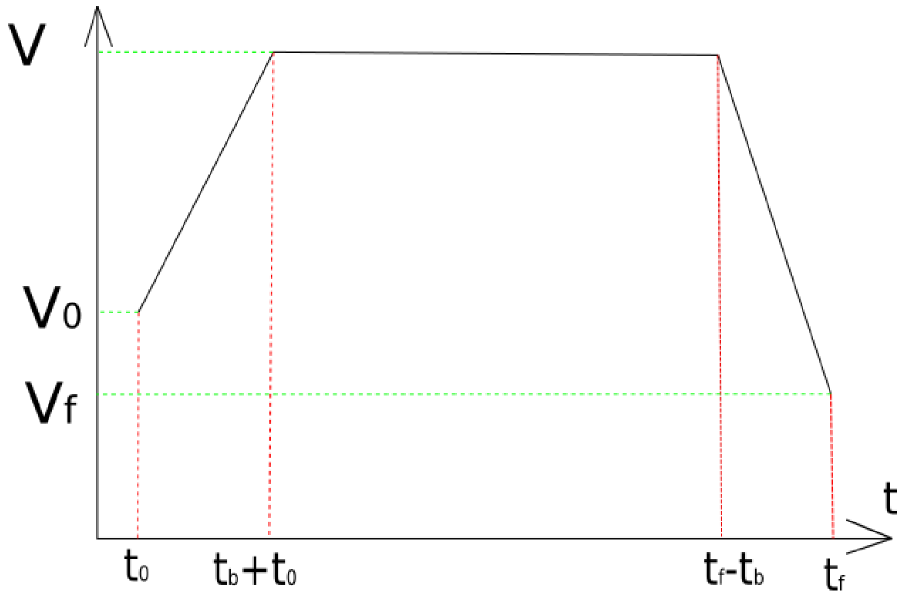

Figure 2: Trapezoidal velocity profile

So, we have:

$$q = \begin{cases} a_{10} + a_{11}t + a_{12}t^2 & t_0 \leqslant t \leqslant t_0 + t_b \\ Vt + C_1 & t_0 + t_b < t \leqslant t_f - t_b \\ a_{20} + a_{21}t + a_{22}t^2 & t_f - t_b < t \leqslant t_f \end{cases} \tag{12}$$

What interval do we have to choose t_b?

- $min_{t_b} = t_0$
- $max_{t_b} = \frac{t_f + t_0}{2}$
- So, $t_b \in [t_0; \frac{t_f+t_0}{2}]$

I decided to take the mean of min_{t_b} and max_{t_b} to set equal to t_b.

After that, we should calculate V for that t_b. To do that, we should at first calculate the area S below the line in the figure 2.

$$S = \frac{V + V_0}{2} t_b + V(t_f - t_0 - 2t_b) + \frac{V + V_f}{2} t_b \tag{13}$$

and

$$S = q_f - q_0 \tag{14}$$

Thus, we can calculate V:

$$V = \frac{q_0 - q_f + \frac{V_0 t_b}{2} + \frac{V_f t_b}{2}}{t_0 + t_b - t_f} \tag{15}$$

So, now it's pretty understandable how to implement this type of profile.
I implemented it on our constraints and got the next results:

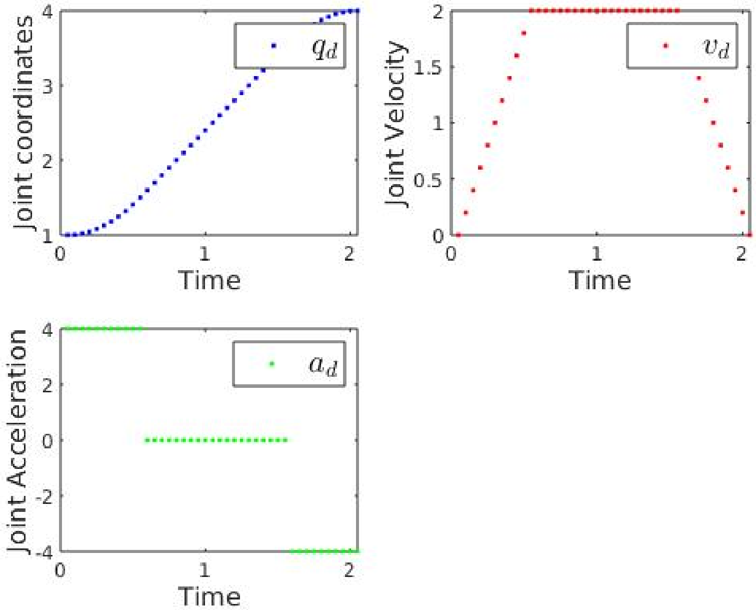

Figure 3: The joint trajectory from $q(0) = 1$ to $q(2) = 4$ (trapezoidal)

Velocity profile of the type $\dot{q}(t) = a(b + sin(ct))$

To solve this task I needed to calculate differential and integral of $\dot{q}(t)$ to obtain $\ddot{q}(t)$ and $q(t)$.

- $\dot{q}(t) = a(b + sin(ct))$
- $\ddot{q}(t) = ca\cos(ct))$
- $q(t) = a(bt - \frac{cos(ct)}{c}) + C_0$

I used function **solve()** in MATLAB to solve the system of equations including all the constraints.

The result you can see in the figure 4.

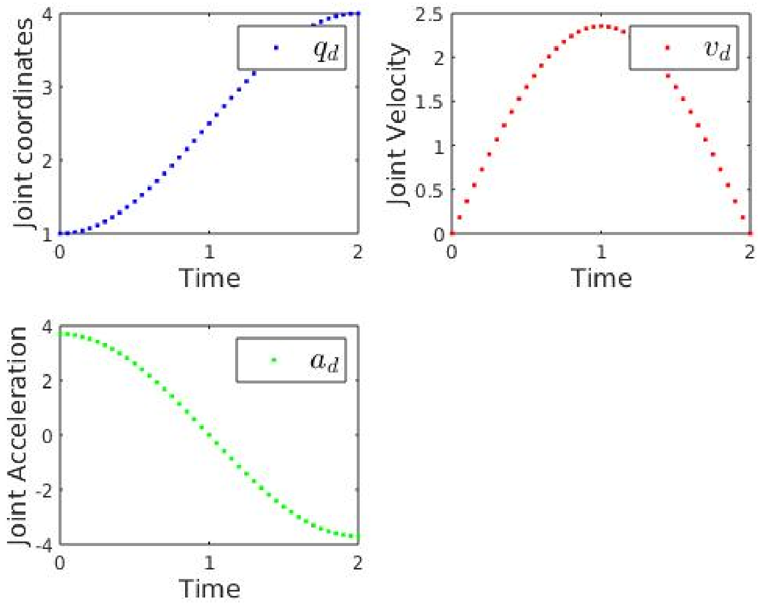

Figure 4: The joint trajectory from $q(0) = 1$ to $q(2) = 4$ ($\dot{q(t)} = a(b + sin(ct))$)

Task 2

Compute the joint trajectory through points $q(0) = 1$, $q(2) = 2$, $q(4) = 0$ with null initial and final velocities and accelerations.

1. Use polynomial position profile
2. Use trapezoidal velocity profile.
3. Use splines

Polynomial Position Profile

I have here 7 constraints, so, I need to use polynomial of power 6. And the idea of calculating the coefficients is the same as in subsection Polynomial Position Profile.

The result you can see in the figure 5.

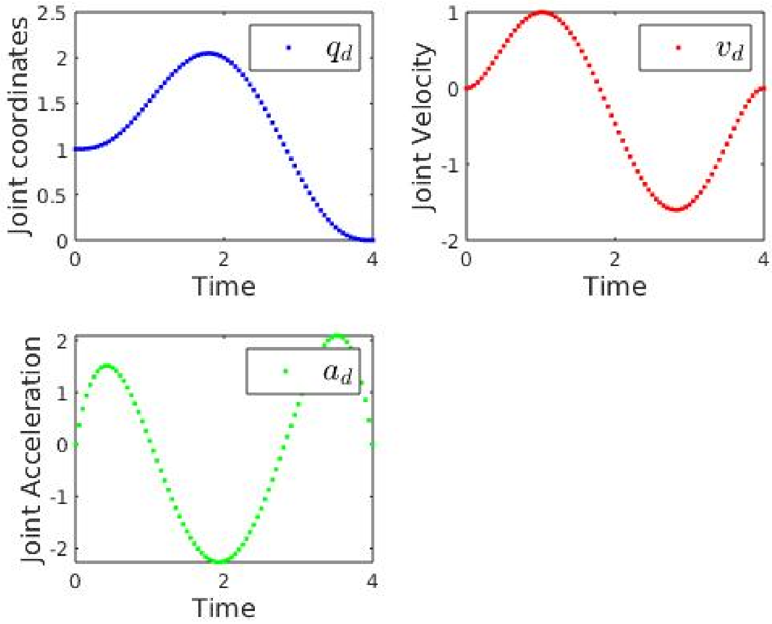

Figure 5: joint trajectory through points $q(0) = 1$, $q(2) = 2$, $q(4) = 0$ (polynomial)

Trapezoidal Velocity Profile

Here I had 7 constraints, so I needed to divide the interval into 2 subintervals and make 2 trapezoidal velocity profiles there to satisfy the constraints. The idea is the same as in subsection Trapezoidal Velocity Profile So, equations for my trajectory are:

$$q = \begin{cases} a_{10} + a_{11}t + a_{12}t^2 & t_0 \leqslant t \leqslant t_0 + t_b \\ V_1 t + C_1 & t_0 + t_b < t \leqslant t_1 - t_b \\ a_{20} + a_{21}t + a_{22}t^2 & t_1 - t_b \leqslant t \leqslant t_1 \\ a_{30} + a_{31}t + a_{32}t^2 & t_1 \leqslant t \leqslant t_1 + t_b \\ V_2 t + C_2 & t_1 + t_b < t \leqslant t_f - t_b \\ a_{40} + a_{41}t + a_{42}t^2 & t_f - t_b \leqslant t \leqslant t_f \end{cases} \tag{16}$$

The results you can see in the figure 6.

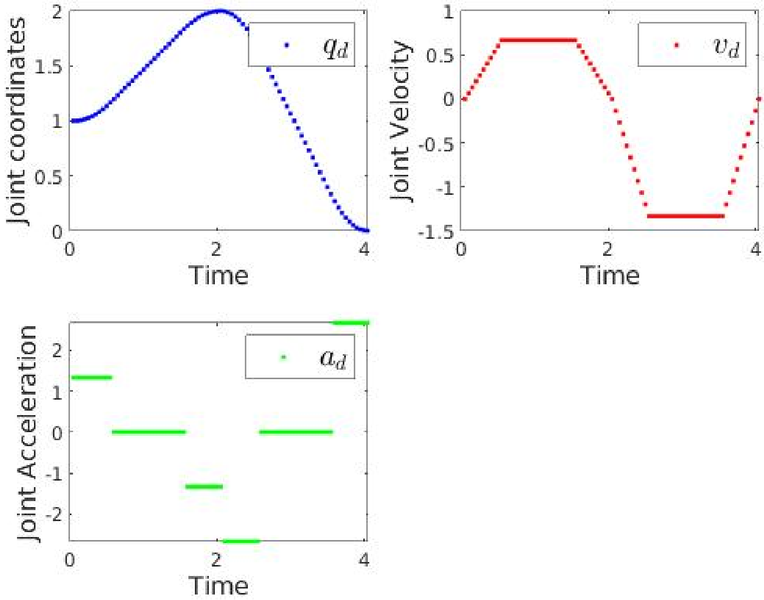

Figure 6: joint trajectory through points $q(0) = 1$, $q(2) = 2$, $q(4) = 0$ (trapezoidal)

Splines

An alternative to using a single high order polynomial for the entire trajectory is to use low order polynomials for trajectory segments between adjacent via points. These polynomials sometimes refered to as interpolating polyno- mials or blending polynomials. With this approach, we must take care that continuity constraints (e.g., in velocity and acceleration) are satisfied at the via points, where we switch from one polynomial to another.

For example, given initial and final times, t_0 and t_f, respectively, with

$$q^d(t_0) = q_0; q^d(t_f) = q_1 \dot{q}^d(t_0) = q_0'; \dot{q}^d(t_f) = q_1' \tag{17}$$

the required cubic polynomial q d (t) can be computed from

$$q^d(t_0) = a_0 + a_1(t - t_0) + a_2(t - t_0)^2 + a_3(t - t_0)^3 \tag{18}$$

where

$$a_2 = \frac{3(q_1 - q_0) - (2q_0' + q'1)(t_f - t_0)}{(t_f - t_0)^2} \tag{19}$$

$$a_3 = \frac{2(q_0 - q_1) + (q_0' + q'1)(t_f - t_0)}{(t_f - t_0)^3} \tag{20}$$

I had also constraints about acceleration, so I needed to use polynomials of power equal to three and five.

$$q = \begin{cases} a_{00} + a_{01}(t) + a_{02}(t)^2 + a_{03}(t)^3 & t_0 \leqslant t \leqslant t_1 \\ a_{10} + a_{11}(t) + a_{12}(t)^2 + a_{13}(t)^3 + a_{14}(t)^4 + +a_{15}(t)^5 & t_1 \leqslant t \leqslant t_f \end{cases} \tag{21}$$

The results you can see in the figure 7.

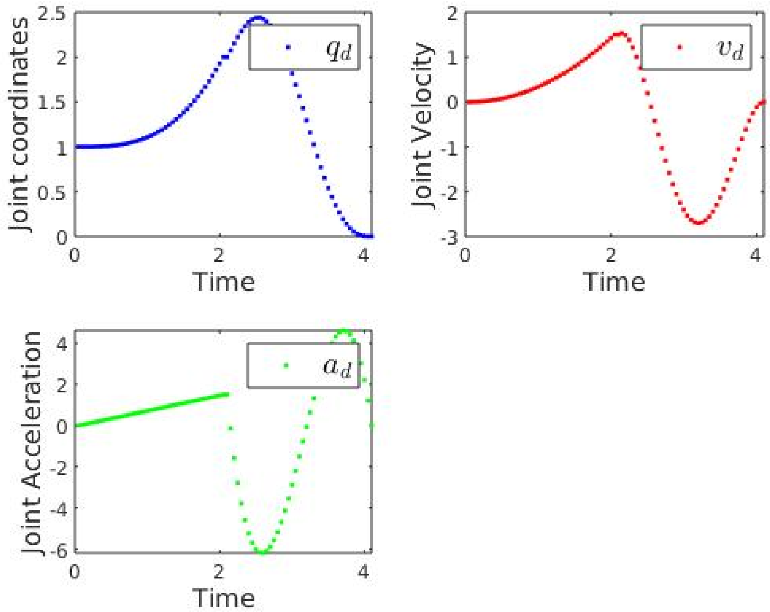

Figure 7: joint trajectory through points $q(0) = 1$, $q(2) = 2$, $q(4) = 0$ (Splines)

Results

Let me compare different approaches that I used in both tasks.

Table of Max, Min $\dot{q}(t)$, $\ddot{q}(t)$				
Method	$\dot{q}_m in(t)$	$\dot{q}_m ax(t)$	$\ddot{q}_m in(t)$	$\ddot{q}_m ax(t)$
$POLYNOMIAL_1$	0	2.2	-4.5	4.5
$TRAPEZOIDAL_1$	0	2	-4	4
$\dot{q}(t) = a(b + sin(ct))$	0	2.3	-3.7	3.7
$POLYNOMIAL_2$	-1.6	1	-2.3	2.1
$TRAPEZOIDAL_2$	-1.3	0.6	-2.6	2.6
$SPLINE$	-2.7	1.5	-6.2	4.6

Table 1: Methods Comparison

As I concluded, with bigger amount of constraints spline is more unsufficient and enegry expensive than other methods (see 1). Sine law also makes robot move too fast what can affect on energy efficiency. Polynomial ones give smooth enough result and low energy consumption. Also, spline has fast changing for jerk, that can damage the construction.