ROBOTICS, VISION AND CONTROL

Trajectory Planning. Multipoint

Riccardo Muradore

UNIVERSITÀ di VERONA
Dipartimento di INFORMATICA

ALTAIR

Outline

Problem statement

Joint Space Trajectories
Sequence of Points

PROJECT

Problem statement

Multipoint Trajectories = Trajectories through a Sequence of Points

Functions suitable for the interpolation (or approximation) of a set of given points (t_k, q_k), $k = 0, \ldots, n$.

The problem is discussed now in the case of a single axis of motion ($q_k \in \mathbb{R}$).

Possible approaches:

- polynomial functions of proper degree $\star$
- orthogonal polynomials
- trigonometric polynomials
- cubic spline functions $\star$
- B-spline functions
- non-linear filters

Joint Space Trajectories
Sequence of Points

Given $n+1$ pairs (q_k, t_k), design a trajectory such that the end-effector passes by each point q_k (*path points, via points*) at a specific instant of time t_k.

Given $n+1$ pairs (q_k, t_k), design a trajectory such that the end-effector passes by each point q_k (*path points, via points*) at a specific instant of time t_k.

Solutions:

1. a high-order polynomial (n) to consider all the constraints

$$q(t) = a_n t^n + a_{n-1} t^{n-1} + \ldots + a_2 t^2 + a_1 t + a_0, \qquad t \in \{t_0, t_1, \ldots, t_{n-1}, t_n\}$$

The solution of the interpolation problem of $n+1$ points can be obtained by solving a linear system of $n+1$ equations in $n+1$ unknowns

$$\boldsymbol{q} = \begin{bmatrix} q_0 \\ q_1 \\ \vdots \\ q_{n-1} \\ q_n \end{bmatrix} = \begin{bmatrix} 1 & t_0 & t_0^2 & \ldots & t_0^n \\ 1 & t_1 & t_1^2 & \ldots & t_1^n \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ 1 & t_{n-1} & t_{n-1}^2 & \ldots & t_{n-1}^n \\ 1 & t_n & t_n^2 & \ldots & t_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \\ a_n \end{bmatrix} = V\,\boldsymbol{a}$$

V is a Vandermonde matrix which is non singular if $t_{k+1} > t_k, \forall k$

The unknowns can be computed by $\quad \boldsymbol{a} = V^{-1}\,\boldsymbol{q}$

Advantages:

- The trajectory defined in this way crosses all the given points
- Only $n+1$ coefficients are needed
- The derivatives (of any order) of the function $q(t)$ are continuous in $[t_0, t_n]$
- The interpolating trajectory is unique

Disadvantages:

- inefficient from the computational point of view and may produce numerical errors for large values of n; the condition number of the matrix V ($\kappa = \frac{\sigma_{max}}{\sigma_{min}}$) is proportional to n
- The degree of the polynomial depends on the number of points ($\Rightarrow$ big V matrix)
- The variation of a single point, or the insertion of a new point, implies that all the coefficients of the polynomial must be recomputed.
- The resulting trajectories are usually characterized by pronounced ‘oscillations’ that are usually unacceptable.

Remark. The Vandermonde matrix V can be enhanced to consider initial and final velocities and accelerations (four more lines)

2. a suitable number of low-order interpolating polynomials (*motion primitives*) continuous at the path points

The solution #2 can exploit cubic polynomials

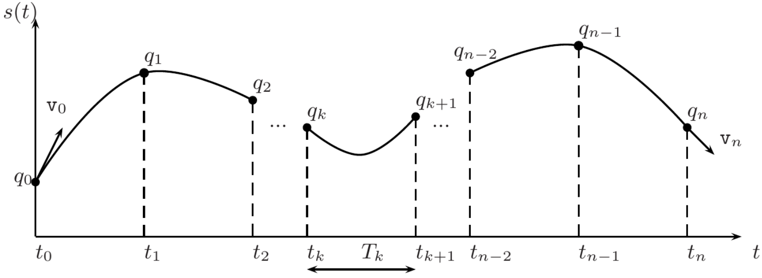
$$q(t) = a_3 t^3 + a_2 t^2 + a_1 t + a_0$$

that we used in the Point-to-Point motion. They can guarantee continuity of positions and velocities at the n path points.

▷ n cubic polynomials $\Pi_k(t)$, for $k = 0, \ldots, n-1$

▷ $\Pi_k(t)$ interpolates the point q_k at t_k and q_{k+1} at t_{k+1}

N.B. $q_i = q_0$ $(t_i = t_0)$, $q_f = q_n$ $(t_f = t_n)$



$$s(t) = \{\Pi_k(t),\ t \in [t_k, t_{k+1}],\ k = 0, \ldots, n-1\}$$

where $T_k = t_{k+1} - t_k$ and

$$\Pi_k(t) = a_3^k(t - t_k)^3 + a_2^k(t - t_k)^2 + a_1^k(t - t_k) + a_0^k$$

Since n polynomials are necessary for the definition of a trajectory through $n + 1$ points, the total number of coefficients to be determined is $4n$.

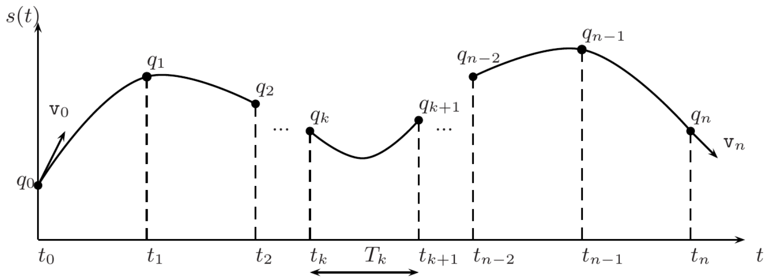
Joint Space Trajectories – Sequence of Points

Conditions for solving the problems:

- $2n$ conditions for the interpolation of the given points (t_k, q_k), since each cubic function must cross the points at its extremities.
- $n-1$ conditions for the continuity of the velocities at the transition points.
- $n-1$ conditions for the continuity of the accelerations at the transition points.

There are $2n + 2(n-1) = 4n - 2$ conditions; the remaining degrees of freedom are $4n - (4n - 2) = 2$.

We need to impose two additional constraints.



Different pairs of conditions

1. The initial and final velocity $\dot{q}_0$, $\dot{q}_n$
2. The initial and final accelerations $\ddot{q}_0$, $\ddot{q}_n$
3. Periodic trajectory $\dot{q}_0 = \dot{q}_n$, $\ddot{q}_0 = \ddot{q}_n$

Observations:

- The degree of the polynomials used to construct the spline does not depend on the number n of data points
- The function $s(t)$ has continuous derivatives up to the order $(p-1)$.
- The jerk $\dddot{s}(t)$ of a cubic spline ($p=3$) is piecewise constant between t_k and t_{k+1}, $k=0,\ldots,n-1$
- The function $f(t)$ which minimizes the functional

$$J=\int_{t_0}^{t_n}\left(\frac{d^2 f(t)}{dt^2}\right)^2 dt$$

(proportional to the curvature of $f(t)$) and for which

$$E=\int_{t_0}^{t_n}\left(\ddot{f}(t)-\ddot{s}(t)\right)^2 dt$$

is equal to zero (i.e. $f(t)=s(t)$) is the cubic spline with zero conditions on the initial and final acceleration. This spline is called *natural spline*.

Joint Space Trajectories – Sequence of Points

Interpolating polynomials with imposed velocities at path | initial | final points

Let

$$
\begin{aligned}
s(t) &= \{\Pi_k(t),\ t \in [t_k, t_{k+1}],\ k = 0, \ldots, n-1\} \\
\Pi_k(t) &= a_3^k t^3 + a_2^k t^2 + a_1^k t + a_0^k
\end{aligned}
$$

We have to solve the system

$$
\begin{aligned}
\Pi_k(t_k) &= q_k, & k &= 0, \ldots, n-1 \\
\dot{\Pi}_{k+1}(t_{k+1}) &= \dot{q}_{k+1}, & k &= 0, \ldots, n-2 \\
\Pi_k(t_{k+1}) &= q_{k+1}, & k &= 0, \ldots, n-1 \\
\dot{\Pi}_k(t_{k+1}) &= \dot{q}_{k+1}, & k &= 0, \ldots, n-2
\end{aligned}
$$

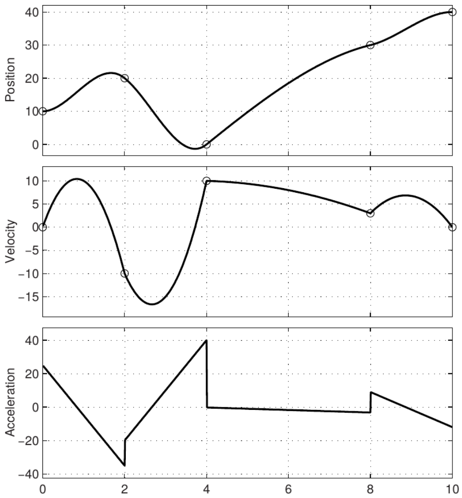
in the unknowns $\{a_3^k, a_2^k, a_1^k, a_0^k\}$, with the constraints (continuity of $\dot{q}(t)$)

$$\dot{\Pi}_k(t_{k+1}) = \dot{\Pi}_{k+1}(t_{k+1})$$

for $k = 0, \ldots, n-1$.

The function $\ddot{q}(t)$ is discontinuous.

Joint Space Trajectories – Sequence of Points



Interpolating polynomials with imposed velocities at path | initial | final points (II version)

Let

$$\Pi_k(t) = a_3^k(t-t_k)^3 + a_2^k(t-t_k)^2 + a_1^k(t-t_k) + a_0^k$$

$k = 1, \ldots, n-1$, the previous conditions on positions and velocity will bring to

$$\begin{array}{rcl} \Pi_k(t_k) & = & a_0^k = q_k \\ \dot{\Pi}_k(t_k) & = & a_1^k = \dot{q}_k \\ \Pi_k(t_{k+1}) & = & a_3^k T_k^3 + a_2^k T_k^2 + a_1^k T_k + a_0^k = q_{k+1} \\ \dot{\Pi}_k(t_{k+1}) & = & 3a_3^k T_k^2 + 2a_2^k T_k + a_1^k = \dot{q}_{k+1} \end{array}$$

where $T_k = t_{k+1} - t_k$

The solution is

$$
\begin{aligned}
a_0^k &= q_k \\
a_1^k &= \dot{q}_k \\
a_2^k &= \frac{1}{T_k}\left[\frac{3(q_{k+1}-q_k)}{T_k} - 2\dot{q}_k - \dot{q}_{k+1}\right] \\
a_3^k &= \frac{1}{T_k^2}\left[\frac{2(q_k-q_{k+1})}{T_k} + \dot{q}_k + \dot{q}_{k+1}\right]
\end{aligned}
$$

Joint Space Trajectories – Sequence of Points

Interpolating polynomials with computed velocities at path points and imposed velocity at initial / final points

The $n+1$ pairs (q_k, t_k) already have a sort of information about the velocity.

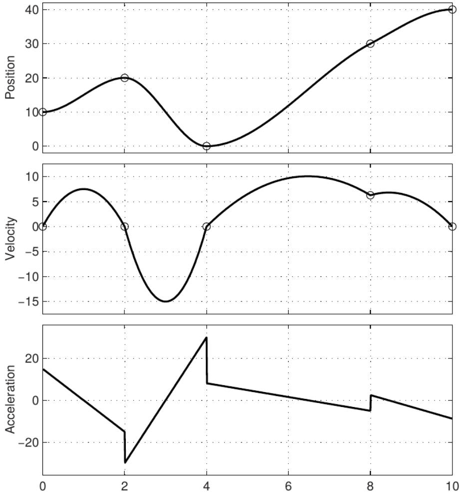
For $k = 1, \ldots, n-1$, we compute

$$v_k := \frac{q_k - q_{k-1}}{t_k - t_{k-1}}, \qquad (\sim \text{Euler approximation})$$

The velocity in each path point is computed as

$$\begin{aligned} \dot{q}_0 &= 0 \\ \dot{q}_k &= \begin{cases} 0, & \text{if } \operatorname{sgn}(v_k) \neq \operatorname{sgn}(v_{k+1}) \\ \frac{v_k + v_{k+1}}{2}, & \text{if } \operatorname{sgn}(v_k) = \operatorname{sgn}(v_{k+1}) \end{cases} \\ \dot{q}_n &= 0 \end{aligned}$$

Joint Space Trajectories – Sequence of Points



Interpolating polynomials with continuous accelerations at path points and imposed velocity at initial | final points

To have $q(t) \in \mathcal{C}^2$ two position constraints for each of the adjacent cubic polynomials and two constraints guaranteeing continuity of velocity and acceleration

$$
\begin{aligned}
\Pi_{k-1}(t_k) &= q_k && \text{(position constraint)} \\
\Pi_{k-1}(t_k) &= \Pi_k(t_k) && \text{(position constraint)} \\
\dot{\Pi}_{k-1}(t_k) &= \dot{\Pi}_k(t_k) && \text{(velocity constraint)} \\
\ddot{\Pi}_{k-1}(t_k) &= \ddot{\Pi}_k(t_k) && \text{(acceleration constraint)}
\end{aligned}
$$

which correspond to $4n - 2$ equations in $4(n - 1)$ unknowns.

Two missing equations !!! $\Pi_0(t_1)$ and $\Pi_n(t_n)$ cannot be included.

Let re-write $\ddot{\Pi}_{k-1}(t_k) = \ddot{\Pi}_k(t_k)$ as

$$\begin{aligned} \ddot{\Pi}_k(t_{k+1}) &= \ddot{\Pi}_{k+1}(t_{k+1}) \\ 6a_3^k T_k + 2a_2^k &= 2a_2^{k+1}, \qquad\qquad k = 0, \ldots, n-2 \end{aligned}$$

Using the previous expression for a_3^k and a_2^k, a_2^{k+1}, we end up with

$$6\frac{1}{T_k^2}\left[\frac{2(q_k - q_{k+1})}{T_k} + \dot{q}_k + \dot{q}_{k+1}\right]T_k + 2\frac{1}{T_k}\left[\frac{3(q_{k+1} - q_k)}{T_k} - 2\dot{q}_k - \dot{q}_{k+1}\right] =$$

$$= 2\frac{1}{T_{k+1}}\left[\frac{3(q_{k+2} - q_{k+1})}{T_{k+1}} - 2\dot{q}_{k+1} - \dot{q}_{k+2}\right]$$

from where we need to "isolate" the unknown velocities $\dot{q}_k, \dot{q}_{k+1}, \dot{q}_{k+2}$ as a function of q_k, q_{k+1}, q_{k+2} and T_k, T_{k+1}

For $k = 0, \ldots, n-2$

$$T_{k+1}\dot{q}_k + 2(T_k + T_{k+1})\dot{q}_{k+1} + T_k\dot{q}_{k+2} = 3\frac{T_{k+1}}{T_k}(q_{k+1} - q_k) + 3\frac{T_k}{T_{k+1}}(q_{k+2} - q_{k+1})$$

$$\begin{bmatrix} T_{k+1} & 2(T_k + T_{k+1}) & T_k \end{bmatrix} \begin{bmatrix} \dot{q}_k \\ \dot{q}_{k+1} \\ \dot{q}_{k+2} \end{bmatrix} = c_k$$

and finally

$$\begin{bmatrix} T_1 & 2(T_0 + T_1) & T_0 & 0 & & & & & \\ 0 & T_2 & 2(T_1 + T_2) & T_1 & & & & & \\ & & \ddots & \ddots & \ddots & & & & \\ & & & T_{k+1} & 2(T_k + T_{k+1}) & T_k & & & \\ & & & & \ddots & \ddots & \ddots & & \\ & & & & & T_{n-1} & 2(T_{n-2} + T_{n-1}) & T_{n-2} \end{bmatrix} \begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \vdots \\ \dot{q}_k \\ \vdots \\ \dot{q}_n \end{bmatrix} = \begin{bmatrix} c_0 \\ \vdots \\ c_k \\ \vdots \\ c_{n-2} \end{bmatrix}$$

Since $\dot{q}_0$ and $\dot{q}_n$ are known, we have

$$\begin{bmatrix} 2(T_0+T_1) & T_0 & 0 & & & & \\ T_2 & 2(T_1+T_2) & T_1 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & T_{k+1} & 2(T_k+T_{k+1}) & T_k & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & T_{n-1} & 2(T_{n-2}+T_{n-1}) \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_k \\ \vdots \\ \dot{q}_{n-1} \end{bmatrix} = \begin{bmatrix} c_0 - T_1\dot{q}_0 \\ \vdots \\ c_k \\ \vdots \\ c_{n-2} - T_{n-2}\dot{q}_n \end{bmatrix}$$

or in compact form

$$\boldsymbol{A}\dot{\boldsymbol{q}} = \boldsymbol{c}$$

with $\boldsymbol{A} \in \mathbb{R}^{(n-1)\times(n-1)}$, $\dot{\boldsymbol{q}} \in \mathbb{R}^{n-1}$, and $\boldsymbol{c} \in \mathbb{R}^{n-1}$

Since $\boldsymbol{A}$ has a “tri-diagonal” structure, it is non singular if $T_k > 0$.

Then $\dot{\boldsymbol{q}} = \boldsymbol{A}^{-1}\boldsymbol{c}$, and it is possible to use the previous equations for computing a_i^k

Assumption: $b_1 \neq 0$ (*what can we do if* $b_1 = 0$ *?*)

$$\begin{bmatrix} b_1 & c_1 & 0 & & & & \\ a_2 & b_2 & c_2 & 0 & & & \\ 0 & \ddots & \ddots & \ddots & 0 & & \\ & 0 & a_k & b_k & c_k & 0 & \\ & & 0 & \ddots & \ddots & \ddots & 0 \\ & & & 0 & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & & 0 & a_n & b_n \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_k \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ \vdots \\ d_k \\ \vdots \\ d_n \end{bmatrix}$$

Thomas algorithm

Forward elimination

for $k = 2 : 1 : n$ **do**

$\quad m = \frac{a_k}{b_{k-1}}$

$\quad b_k = b_k - m c_{k-1}$

$\quad d_k = d_k - m d_{k-1}$

end

Backward substitution

$x_n = \frac{d_n}{b_n}$

for $k = n-1 : -1 : 1$ **do**

$\quad x_k = \frac{d_k - c_k x_{k+1}}{b_k}$

end


Project

To do

- Interpolating polynomials with computed velocities at path points and imposed velocity at initial | final points
- Interpolating polynomials with continuous accelerations at path points and imposed velocity at initial | final points (+ Thomas algorithm)