ROBOTICS, VISION AND CONTROL

Trajectory Planning. Multipoint

Riccardo Muradore

UNIVERSITÀ di VERONA
Dipartimento di INFORMATICA

Outline

Cubic splines with assigned initial and final velocities: computation based on the accelerations

Smoothing cubic splines

PROJECT

Cubic splines with assigned initial and final velocities: computation based on the accelerations

Given

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

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

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

the coefficient $\{a_0^k, a_1^k, a_2^k, a_3^k\}$ of the k-th cubic polynomial $\Pi_k(t)$ can be expressed in terms of positions q_k, q_{k+1} and velocities $\dot{q}_k, \dot{q}_{k+1}$ (as we saw in the previous slides) or in terms of positions q_k, q_{k+1} and accelerations $\ddot{q}_k, \ddot{q}_{k+1}$

$$\begin{aligned}
\Pi_k(t_k) &= a_0^k = q_k \\
\ddot{\Pi}_k(t_k) &= 2a_2^k = \ddot{q}_k \\
\Pi_k(t_{k+1}) &= a_3^k T_k^3 + a_2^k T_k^2 + a_1^k T_k + a_0^k = q_{k+1} \\
\ddot{\Pi}_k(t_{k+1}) &= 6a_3^k T_k + 2a_2^k = \ddot{q}_{k+1}
\end{aligned}$$

Joint Space Trajectories – Sequence of Points

Solving the systems we have

$$
\begin{aligned}
a_0^k &= q_k \\
a_1^k &= \frac{q_{k+1} - q_k}{T_k} - \frac{\ddot{q}_{k+1} + 2\ddot{q}_k}{6} T_k \\
a_2^k &= \frac{\ddot{q}_k}{2} \\
a_3^k &= \frac{\ddot{q}_{k+1} - \ddot{q}_k}{6 T_k}
\end{aligned}
$$

and so

$$
\Pi_k(t) = \left(\frac{\ddot{q}_{k+1} - \ddot{q}_k}{6 T_k} \right) (t - t_k)^3 + \left(\frac{\ddot{q}_k}{2} \right) (t - t_k)^2 + \left(\frac{q_{k+1} - q_k}{T_k} - \frac{\ddot{q}_{k+1} + 2\ddot{q}_k}{6} T_k \right) (t - t_k) + q_k
$$

Cubic splines with assigned initial and final velocities: computation based on the accelerations

The k-th cubic polynomial $\Pi_k(t)$ of the spline is expressed as a function of the accelerations at its endpoints, i.e. $\ddot{q}(t_k)$, $k = 0, \ldots, n$, instead of the velocities $\dot{q}(t_k)$.

$$\Pi_k(t) = \left(\frac{q_{k+1}}{T_k} - \frac{T_k \ddot{q}_{k+1}}{6}\right)(t - t_k) + \left(\frac{q_k}{T_k} - \frac{T_k \ddot{q}_k}{6}\right)(t_{k+1} - t) + \frac{\ddot{q}_k}{6T_k}(t_{k+1} - t)^3 + \frac{\ddot{q}_{k+1}}{6T_k}(t - t_k)^3,$$

$$t \in [t_k, t_{k+1}]$$

Computing the velocity and acceleration, we get

$$\begin{aligned}
\dot{\Pi}_k(t) &= \left(\frac{q_{k+1}}{T_k} - \frac{T_k \ddot{q}_{k+1}}{6}\right) - \left(\frac{q_k}{T_k} - \frac{T_k \ddot{q}_k}{6}\right) - \frac{\ddot{q}_k}{2T_k}(t_{k+1} - t)^2 + \frac{\ddot{q}_{k+1}}{2T_k}(t - t_k)^2 \\
&= \frac{q_{k+1} - q_k}{T_k} - \frac{T_k(\ddot{q}_{k+1} - \ddot{q}_k)}{6} + \frac{\ddot{q}_k}{2T_k}(t_{k+1} - t)^2 + \frac{\ddot{q}_{k+1}}{2T_k}(t - t_k)^2
\end{aligned}$$

$$\ddot{\Pi}_k(t) = \frac{\ddot{q}_k}{T_k}(t_{k+1} - t) + \frac{\ddot{q}_{k+1}}{T_k}(t - t_k)$$

Continuity of accelerations in the intermediate points

$$\ddot{\Pi}_{k-1}(t_k) = \ddot{\Pi}_k(t_k) = \ddot{q}_k \tag{1}$$

Continuity of velocities in the intermediate points

$$\dot{\Pi}_{k-1}(t_k) = \dot{\Pi}_k(t_k) \tag{2}$$

we have

$$\begin{aligned}
\dot{\Pi}_{k-1}(t_k) &= \left(\frac{q_k}{T_{k-1}} - \frac{T_{k-1}\ddot{q}_k}{6}\right) - \left(\frac{q_{k-1}}{T_{k-1}} - \frac{T_{k-1}\ddot{q}_{k-1}}{6}\right) - \frac{\ddot{q}_{k-1}}{2T_{k-1}}(t_k - t_k)^2 + \frac{\ddot{q}_k}{2T_{k-1}}(t_k - t_{k-1})^2 \\
&= \left(\frac{q_{k+1}}{T_k} - \frac{T_k\ddot{q}_{k+1}}{6}\right) - \left(\frac{q_k}{T_k} - \frac{T_k\ddot{q}_k}{6}\right) + \frac{\ddot{q}_k}{2T_k}(t_{k+1} - t_k)^2 + \frac{\ddot{q}_{k+1}}{2T_k}(t_k - t_k)^2 \\
&= \dot{\Pi}_k(t_k)
\end{aligned}$$

$$
\begin{aligned}
\dot{\Pi}_{k-1}(t_k) &= \left(\frac{q_k}{T_{k-1}} - \frac{T_{k-1}\ddot{q}_k}{6}\right) - \left(\frac{q_{k-1}}{T_{k-1}} - \frac{T_{k-1}\ddot{q}_{k-1}}{6}\right) - \frac{\ddot{q}_{k-1}}{2T_{k-1}}(t_k - t_k)^2 + \frac{\ddot{q}_k}{2T_{k-1}}(t_k - t_{k-1})^2 \\
&= \left(\frac{q_{k+1}}{T_k} - \frac{T_k\ddot{q}_{k+1}}{6}\right) - \left(\frac{q_k}{T_k} - \frac{T_k\ddot{q}_k}{6}\right) + \frac{\ddot{q}_k}{2T_k}(t_{k+1} - t_k)^2 + \frac{\ddot{q}_{k+1}}{2T_k}(t_k - t_k)^2 \\
&= \dot{\Pi}_k(t_k)
\end{aligned}
$$

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

Remembering that $T_k = t_{k+1} - t_k$ we end up with

$$
\begin{aligned}
\left(\frac{q_k}{T_{k-1}} - \frac{T_{k-1}\ddot{q}_k}{6}\right) - \left(\frac{q_{k-1}}{T_{k-1}} - \frac{T_{k-1}\ddot{q}_{k-1}}{6}\right) + \frac{\ddot{q}_k}{2T_{k-1}}T_{k-1}^2 = \\
= \left(\frac{q_{k+1}}{T_k} - \frac{T_k\ddot{q}_{k+1}}{6}\right) - \left(\frac{q_k}{T_k} - \frac{T_k\ddot{q}_k}{6}\right) + \frac{\ddot{q}_k}{2T_k}T_k^2
\end{aligned}
$$

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

Collecting the unknown accelerations $\ddot{q}_{k-1}, \ddot{q}_k, \ddot{q}_{k+1}$ on the left side and the known values on the right side, we have

$$
\begin{aligned}
T_{k-1}\ddot{q}_{k-1} + 2(T_{k-1} + T_k)\ddot{q}_k + T_k\ddot{q}_{k+1} = \\
= 6\left(\frac{q_{k+1} - q_k}{T_k}\right) - 6\left(\frac{q_k - q_{k-1}}{T_{k-1}}\right)
\end{aligned}
$$

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

Since we know the *initial and final velocities* $\dot{q}_0$, $\dot{q}_n$, using

$$
\dot{\Pi}_k(t) = \frac{q_{k+1} - q_k}{T_k} - \frac{T_k(\ddot{q}_{k+1} - \ddot{q}_k)}{6} - \frac{\ddot{q}_k}{2T_k}(t_{k+1} - t)^2 + \frac{\ddot{q}_{k+1}}{2T_k}(t - t_k)^2
$$

we have the two further equations

$$
\begin{aligned}
\dot{\Pi}_0(t_0) &= \dot{q}_0 \\
\dot{\Pi}_{n-1}(t_n) &= \dot{q}_n
\end{aligned}
$$

Joint Space Trajectories – Sequence of Points

$$
\begin{aligned}
\dot{\Pi}_0(t_0) &= \dot{q}_0 \\
&= \frac{q_1 - q_0}{T_0} - \frac{T_0(\ddot{q}_1 - \ddot{q}_0)}{6} - \frac{\ddot{q}_0}{2T_0}(t_1 - t_0)^2 + \frac{\ddot{q}_1}{2T_0}(t_0 - t_0)^2 \\
&= \frac{q_1 - q_0}{T_0} - \frac{T_0(\ddot{q}_1 - \ddot{q}_0)}{6} - \frac{\ddot{q}_0}{2T_0} T_0^2 \\
\dot{\Pi}_{n-1}(t_n) &= \dot{q}_n \\
&= \frac{q_n - q_{n-1}}{T_{n-1}} - \frac{T_{n-1}(\ddot{q}_n - \ddot{q}_{n-1})}{6} - + \frac{\ddot{q}_{n-1}}{2T_{n-1}}(t_n - t_n)^2 + \frac{\ddot{q}_n}{2T_{n-1}}(t_n - t_{n-1})^2 \\
&= \frac{q_n - q_{n-1}}{T_{n-1}} - \frac{T_{n-1}(\ddot{q}_n - \ddot{q}_{n-1})}{6} + \frac{\ddot{q}_n}{2T_{n-1}} T_{n-1}^2
\end{aligned}
$$

Hence

$$
\begin{aligned}
\frac{T_0^2}{3}\ddot{q}_0 + \frac{T_0^2}{6}\ddot{q}_1 &= q_1 - q_0 - T_0\dot{q}_0 \\
\frac{T_{n-1}^2}{3}\ddot{q}_n + \frac{T_{n-1}^2}{6}\ddot{q}_{n-1} &= q_{n-1} - q_n + T_{n-1}\dot{q}_n
\end{aligned}
$$

Collecting the equations in a linear system, $\boldsymbol{A}\ddot{\boldsymbol{q}} = \boldsymbol{c}$, ($\boldsymbol{A} \in \mathbb{R}^{(n+1)\times(n+1)}$) we have

$$
\begin{bmatrix}
2T_0 & T_0 & 0 & & & & & \\
T_0 & 2(T_0+T_1) & T_1 & & & & & \\
 & \ddots & \ddots & \ddots & & & & \\
 & & T_k & 2(T_k+T_{k+1}) & T_{k+1} & & & \\
 & & & \ddots & \ddots & \ddots & & \\
 & & & & T_{n-2} & 2(T_{n-2}+T_{n-1}) & T_{n-1} \\
 & & & & & T_{n-1} & 2T_{n-1}
\end{bmatrix}
\begin{bmatrix}
\ddot{q}_0 \\ \ddot{q}_1 \\ \vdots \\ \ddot{q}_k \\ \vdots \\ \ddot{q}_{n-1} \\ \ddot{q}_n
\end{bmatrix}
=
\begin{bmatrix}
6\left(\frac{q_1-q_0}{T_0} - \dot{q}_0\right) \\
6\left(\frac{q_2-q_1}{T_1} - \frac{q_1-q_0}{T_0}\right) \\
\vdots \\
6\left(\frac{q_n-q_{n-1}}{T_{n-1}} - \frac{q_{n-1}-q_{n-2}}{T_{n-2}}\right) \\
6\left(\dot{q}_n - \frac{q_n-q_{n-1}}{T_{n-1}}\right)
\end{bmatrix}
$$

Joint Space Trajectories – Sequence of Points

A cubic spline is a function continuous up to the second derivative, but in general it is not possible to assign at the same time both initial and final velocities and accelerations.

$\Rightarrow$ *at its extremities the spline is characterized by a discontinuity on the velocities or on the accelerations.*

Possible solutions:

1. Resort to polynomial functions of degree 5 on the first and last tract,

   Drawbacks: larger overshoot in these segments; increasing the computational burden
2. Add two free extra points in the first and last segment: their values are computed by imposing the desired initial and final values of both velocity and acceleration

Smoothing cubic splines

Smoothing cubic splines

Smoothing cubic splines are designed to *approximate* and not interpolate a set of given data points

$$\begin{aligned} \boldsymbol{q} &= \begin{bmatrix} q_0 & q_1 & q_2 & \dots & q_{n-1} & q_n \end{bmatrix}^T \\ \boldsymbol{t} &= \begin{bmatrix} t_0 & t_1 & t_2 & \dots & t_{n-1} & t_n \end{bmatrix}^T \end{aligned}$$

The approximated trajectory $s(t)$, $t \in [t_0, t_n]$ is the solution of a minimization problem which metric L is a trade-off between two apposite goals ($\mu \in [0, 1]$)

$$L = \mu \underbrace{\sum_{k=0}^{n} w_k (s(t_k) - q_k)^2}_{(*)} + (1 - \mu) \underbrace{\int_{t_0}^{t_n} \ddot{s}(t)^2 dt}_{(**)}$$

where

- $(*)$ fitting of the given via-points (t_k, q_k), $k = 0, \dots, n$
- $(**)$ smoothness of the trajectory $s(t)$, i.e. with curvature/acceleration as small as possible.

Smoothing cubic splines

$$L = \mu \underbrace{\sum_{k=0}^{n} w_k (s(t_k) - q_k)^2}_{\text{fitting}} + (1-\mu) \underbrace{\int_{t_0}^{t_n} \ddot{s}(t)^2 dt}_{\text{smoothness}}$$

- $\mu \in [0, 1]$ weights the trade-off between the two conflicting goals
- w_k are parameters which can be arbitrarily chosen in order to modify the weight of the k-th quadratic error on the global optimization problem

  $\Rightarrow$ reduce the approximation error only in some points of interest
- both in the fitting term and in the smoothness term, we have the power of 2

Smoothing cubic splines

We will focus on cubic spline

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

defined as a function of the accelerations (see previous slides *where now we use s_k instead of q_k and $\ddot{s}_k$ instead of $\ddot{q}_k$*)

$$\Gamma_k(t) = \left(\frac{s_{k+1}}{T_k} - \frac{T_k \ddot{s}_{k+1}}{6}\right)(t - t_k) + \left(\frac{s_k}{T_k} - \frac{T_k \ddot{s}_k}{6}\right)(t_{k+1} - t) + \frac{\ddot{s}_k}{6T_k}(t_{k+1} - t)^3 + \frac{\ddot{s}_{k+1}}{6T_k}(t - t_k)^3,$$

$$t \in [t_k, t_{k+1}], \qquad T_k = t_{k+1} - t_k$$

with acceleration

$$\ddot{\Gamma}_k(t) = \frac{\ddot{s}_k}{T_k}(t_{k+1} - t) + \frac{\ddot{s}_{k+1}}{T_k}(t - t_k)$$

The smoothness term is

$$\underbrace{\int_{t_0}^{t_n} \ddot{s}(t)^2 dt}_{\text{smoothness}} = \sum_{k=0}^{n-1} \int_{t_k}^{t_{k+1}} \ddot{\Gamma}_k(t)^2 dt$$

Smoothing cubic splines

The acceleration can be re-written as

$$
\begin{aligned}
\ddot{\Gamma}_k(t) &= \frac{\ddot{s}_k}{T_k}(t_{k+1} - t) + \frac{\ddot{s}_{k+1}}{T_k}(t - t_k) \\
&= \frac{\ddot{s}_k}{T_k}(t_{k+1} - t_k + t_k - t) + \frac{\ddot{s}_{k+1}}{T_k}(t - t_k) \\
&= \ddot{s}_k - \frac{\ddot{s}_k}{T_k}(t - t_k) + \frac{\ddot{s}_{k+1}}{T_k}(t - t_k)
\end{aligned}
$$

and the smoothness term as

$$
\begin{aligned}
\int_{t_0}^{t_n} \ddot{s}(t)^2 dt = \sum_{k=0}^{n-1} \int_{t_k}^{t_{k+1}} \ddot{\Gamma}_k(t)^2 dt &= \sum_{k=0}^{n-1} \int_{t_k}^{t_{k+1}} \left(\ddot{s}_k + \frac{\ddot{s}_{k+1} - \ddot{s}_k}{T_k}(t - t_k) \right)^2 dt \\
&= \sum_{k=0}^{n-1} \int_{0}^{T_k} \left(\ddot{s}_k + \frac{\ddot{s}_{k+1} - \ddot{s}_k}{T_k}\tau \right)^2 d\tau \\
&= \sum_{k=0}^{n-1} \frac{1}{3} T_k \left(\ddot{s}_k^2 + \ddot{s}_k \ddot{s}_{k+1} + \ddot{s}_{k+1}^2 \right)
\end{aligned}
$$

Smoothing cubic splines

The overall cost function is

$$
\begin{aligned}
L &= \underbrace{\mu \sum_{k=0}^{n} w_k (s(t_k) - q_k)^2}_{\text{fitting}} + (1-\mu) \underbrace{\int_{t_0}^{t_n} \ddot{s}(t)^2 dt}_{\text{smoothness}} \\
&= \mu \sum_{k=0}^{n} w_k (s(t_k) - q_k)^2 + (1-\mu) \sum_{k=0}^{n-1} \frac{1}{3} T_k \left(\ddot{s}_k^2 + \ddot{s}_k \ddot{s}_{k+1} + \ddot{s}_{k+1}^2\right)
\end{aligned}
$$

or equivalently (we just divide both term by $\mu > 0$ and so the minimum point does not change)

$$
L = \sum_{k=0}^{n} w_k (s(t_k) - q_k)^2 + \underbrace{\frac{1-\mu}{6\mu}}_{\triangleq \lambda} \sum_{k=0}^{n-1} 2 T_k \left(\ddot{s}_k^2 + \ddot{s}_k \ddot{s}_{k+1} + \ddot{s}_{k+1}^2\right)
$$

Smoothing cubic splines

Let

$$
\begin{aligned}
\boldsymbol{q} &= \begin{bmatrix} q_0 & q_1 & q_2 & \dots & q_{n-1} & q_n \end{bmatrix}^T \\
\boldsymbol{s} &= \begin{bmatrix} s(t_0) & s(t_1) & s(t_2) & \dots & s(t_{n-1}) & s(t_n) \end{bmatrix}^T \\
&= \begin{bmatrix} s_0 & s_1 & s_2 & \dots & s_{n-1} & s_n \end{bmatrix}^T \\
\ddot{\boldsymbol{s}} &= \begin{bmatrix} \ddot{s}_0 & \ddot{s}_1 & \ddot{s}_2 & \dots & \ddot{s}_{n-1} & \ddot{s}_n \end{bmatrix}^T
\end{aligned}
$$

and

$$
\boldsymbol{A} = \begin{bmatrix}
2T_0 & T_0 & 0 & & & & \\
T_0 & 2(T_0+T_1) & T_1 & & & & \\
 & \ddots & \ddots & \ddots & & & \\
 & & T_k & 2(T_k+T_{k+1}) & T_{k+1} & & \\
 & & & \ddots & \ddots & \ddots & \\
 & & & & T_{n-2} & 2(T_{n-2}+T_{n-1}) & T_{n-1} \\
 & & & & & T_{n-1} & 2T_{n-1}
\end{bmatrix}
$$

$$
\boldsymbol{W} = \operatorname{diag}\{w_0, w_1, \dots, w_{n-1}, w_n\}
$$

Smoothing cubic splines

The cost function L has the following matrix expression

$$L = (\boldsymbol{q} - \boldsymbol{s})^T \boldsymbol{W} (\boldsymbol{q} - \boldsymbol{s}) + \lambda \ddot{\boldsymbol{s}}^T \boldsymbol{A} \ddot{\boldsymbol{s}}$$

in the unknowns $\boldsymbol{s}$ and $\ddot{\boldsymbol{s}}$.

ASSUMPTION: the initial and final velocities are equal to zero, i.e. $\dot{q}_0 = 0$, $\dot{q}_n = 0$ (*clamped spline*)

We know from the “Interpolating polynomials with continuous accelerations at path points and imposed velocity at initial | final points” that there exists a linear relationship between (unknown) accelerations and (known) positions

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

$$\begin{bmatrix} 2T_0 & T_0 & 0 & & & & \\ T_0 & 2(T_0+T_1) & T_1 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & T_k & 2(T_k+T_{k+1}) & T_{k+1} & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & T_{n-2} & 2(T_{n-2}+T_{n-1}) & T_{n-1} \\ & & & & & T_{n-1} & 2T_{n-1} \end{bmatrix} \begin{bmatrix} \ddot{q}_0 \\ \ddot{q}_1 \\ \vdots \\ \ddot{q}_k \\ \vdots \\ \ddot{q}_{n-1} \\ \ddot{q}_n \end{bmatrix} = \begin{bmatrix} 6\left(\frac{q_1-q_0}{T_0} - \dot{q}_0\right) \\ 6\left(\frac{q_2-q_1}{T_1} - \frac{q_1-q_0}{T_0}\right) \\ \vdots \\ 6\left(\frac{q_n-q_{n-1}}{T_{n-1}} - \frac{q_{n-1}-q_{n-2}}{T_{n-2}}\right) \\ 6\left(\dot{q}_n - \frac{q_n-q_{n-1}}{T_{n-1}}\right) \end{bmatrix}$$

Setting $\dot{q}_0 = 0$, $\dot{q}_n = 0$, and remembering that in the approximation problem the positions on the right side of the equation are $s_k = s(t_k)$, we get

$$\begin{bmatrix} 2T_0 & T_0 & 0 & & & & \\ T_0 & 2(T_0+T_1) & T_1 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & T_k & 2(T_k+T_{k+1}) & T_{k+1} & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & T_{n-2} & 2(T_{n-2}+T_{n-1}) & T_{n-1} \\ & & & & & T_{n-1} & 2T_{n-1} \end{bmatrix} \begin{bmatrix} \ddot{s}_0 \\ \ddot{s}_1 \\ \vdots \\ \ddot{s}_k \\ \vdots \\ \ddot{s}_{n-1} \\ \ddot{s}_n \end{bmatrix} = \begin{bmatrix} 6\left(\frac{s_1-s_0}{T_0}\right) \\ 6\left(\frac{s_2-s_1}{T_1} - \frac{s_1-s_0}{T_0}\right) \\ \vdots \\ 6\left(\frac{s_n-s_{n-1}}{T_{n-1}} - \frac{s_{n-1}-s_{n-2}}{T_{n-2}}\right) \\ 6\left(-\frac{s_n-s_{n-1}}{T_{n-1}}\right) \end{bmatrix}$$

The right side can be re-written as

$$
= \begin{bmatrix}
-\frac{6}{T_0} & \frac{6}{T_0} & 0 & & & & & \\
\frac{6}{T_0} & -\left(\frac{6}{T_0}+\frac{6}{T_1}\right) & \frac{6}{T_1} & & & & & \\
& \ddots & \ddots & \ddots & & & & \\
& & \frac{6}{T_k} & -\left(\frac{6}{T_k}+\frac{6}{T_{k+1}}\right) & \frac{6}{T_{k+1}} & & & \\
& & & \ddots & \ddots & \ddots & & \\
& & & & \frac{6}{T_{n-2}} & -\left(\frac{6}{T_{n-2}}+\frac{6}{T_{n-1}}\right) & \frac{6}{T_{n-1}} \\
& & & & & \frac{6}{T_{n-1}} & -\frac{6}{T_{n-1}}
\end{bmatrix}
\begin{bmatrix} s_0 \\ s_1 \\ \vdots \\ s_k \\ \vdots \\ s_{n-1} \\ s_n \end{bmatrix}
$$

In compact form

$$
\boldsymbol{A\ddot{s}} = \boldsymbol{Cs}
$$

Finally

$$\begin{aligned} L &= (\boldsymbol{q}-\boldsymbol{s})^T \boldsymbol{W}(\boldsymbol{q}-\boldsymbol{s}) + \lambda \ddot{\boldsymbol{s}}^T \boldsymbol{A} \ddot{\boldsymbol{s}} \\ &= (\boldsymbol{q}-\boldsymbol{s})^T \boldsymbol{W}(\boldsymbol{q}-\boldsymbol{s}) + \lambda \boldsymbol{s}^T \boldsymbol{C}^T \boldsymbol{A}^{-1} \boldsymbol{C} \boldsymbol{s} \end{aligned}$$

The optimal solution $\boldsymbol{s}^\circ$ is

$$\boldsymbol{s}^\circ = \arg\min_{\boldsymbol{s}} (\boldsymbol{q}-\boldsymbol{s})^T \boldsymbol{W}(\boldsymbol{q}-\boldsymbol{s}) + \lambda \boldsymbol{s}^T \boldsymbol{C}^T \boldsymbol{A}^{-1} \boldsymbol{C} \boldsymbol{s}$$

Computing the first derivative (gradient) w.r.t. $\boldsymbol{s}$ we have

$$\nabla_{\boldsymbol{s}} L(\boldsymbol{s}) = -\boldsymbol{W}(\boldsymbol{q}-\boldsymbol{s}) + \lambda \boldsymbol{C}^T \boldsymbol{A}^{-1} \boldsymbol{C} \boldsymbol{s}$$

and the necessary condition $\nabla_{\boldsymbol{s}} L(\boldsymbol{s}) = 0$ gives

$$\boldsymbol{s} = (\boldsymbol{W} + \lambda \boldsymbol{C}^T \boldsymbol{A}^{-1} \boldsymbol{C})^{-1} \boldsymbol{W} \boldsymbol{q}$$

Computing the second derivative it is possible to prove that this extreme is actually a minimum

$$\nabla_{\boldsymbol{ss}} L(\boldsymbol{s}) = \boldsymbol{W} + \lambda \boldsymbol{C}^T \boldsymbol{A}^{-1} \boldsymbol{C} \succ 0$$

The optimal solution

$$\boldsymbol{s} = (\boldsymbol{W} + \lambda \boldsymbol{C}^T \boldsymbol{A}^{-1} \boldsymbol{C})^{-1} \boldsymbol{W} \boldsymbol{q}$$

can be written in a different way exploiting the matrix inversion lemma

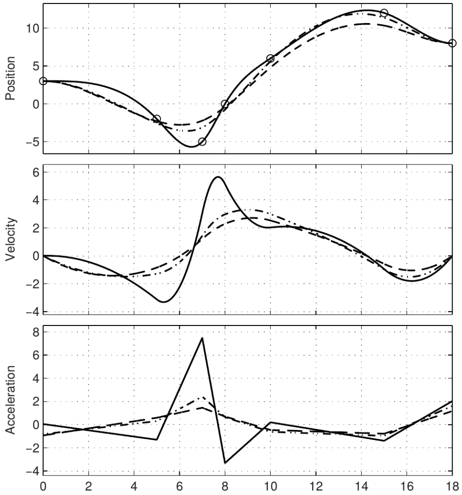
$$\boldsymbol{s} = \boldsymbol{q} - \lambda \boldsymbol{W}^{-1} \boldsymbol{C}^T \left(\boldsymbol{A} + \lambda \boldsymbol{C} \boldsymbol{W}^{-1} \boldsymbol{C}^T \right)^{-1} \boldsymbol{C} \boldsymbol{q}$$

Observation. The unknown accelerations $\ddot{s}$ can be computed as an intermediate step

$$\begin{aligned} \left(\boldsymbol{A} + \lambda \boldsymbol{C} \boldsymbol{W}^{-1} \boldsymbol{C}^T \right) \ddot{\boldsymbol{s}} &= \boldsymbol{C} \boldsymbol{q}. \\ \boldsymbol{s} &= \boldsymbol{q} - \lambda \boldsymbol{W}^{-1} \boldsymbol{C}^T \ddot{\boldsymbol{s}} \end{aligned}$$

This vector allows to define the cubic splines Γ_k, $k = 0, \ldots, n-1$.

Smoothing cubic splines



$\mu \in \{0.3, 0.6, 1\}$ $\boldsymbol{W}^{-1} = \text{diag}\,\{0, 1, \ldots, 1, 0\}$

Remarks.

- Since $\boldsymbol{W} = \text{diag}\{w_0, w_1, \ldots, w_{n-1}, w_n\}$ is diagonal, its inverse is

$$\boldsymbol{W}^{-1} = \text{diag}\left\{\frac{1}{w_0}, \frac{1}{w_1}, \ldots, \frac{1}{w_{n-1}}, \frac{1}{w_n}\right\}$$

- if the curve $s(t)$ has to exactly pass through q_k then we have to set the k-th diagonal element to 0 instead of $\frac{1}{w_k}$
- the approximation errors are larger for those points in which the acceleration (i.e. curvature) is higher.

  To reduce these errors in particular points it is necessary to change the corresponding weights in $\boldsymbol{W}$.
- By recursively applying the algorithm for the computation of smoothing splines, it is possible to find the value of the coefficient μ_δ which guarantees that the maximum approximation error ($\varepsilon_k \triangleq \max_k |s(t_k) - q_k|$) is smaller than a given threshold δ


Project

PROJECT – Assignment # 4

To do

- Compute cubic splines based on the accelerations with assigned initial and final velocities
- Compute the smoothing cubic splines