Trajectory Planning for Robot Manipulators

Claudio Melchiorri

Dipartimento di Ingegneria dell'Energia Elettrica e dell'Informazione (DEI)

Università di Bologna

email: claudio.melchiorri@unibo.it

Summary

Kinematics: geometrical relationships in terms of position/velocity between the joint- and work-space.

Dynamics: relationships between the torques applied to the joints and the consequent movements of the links.

Control: computation of the control actions (joint torques) necessary to execute a desired motion.

Trajectory planning: planning of the *desired* movements of the manipulator.

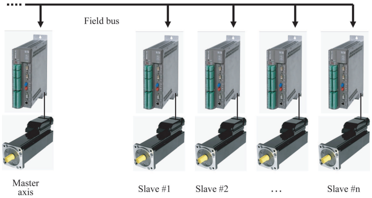
Usually, the user is requested to define some points and general features of the trajectory (e.g. initial/final points, duration, maximum velocity, etc.), and the real computation of the trajectory is demanded to the control system.

Trajectory planning

Trajectory planning: IMPORTANT aspect in robotics, VERY IMPORTANT for the dimensioning, control, and use of electric motors in automatic machines (e.g. packaging).

Origin of the interest for the control area was the substitution of mechanical cams with *electric cams* in the design of automatic machines.

$\Longrightarrow$

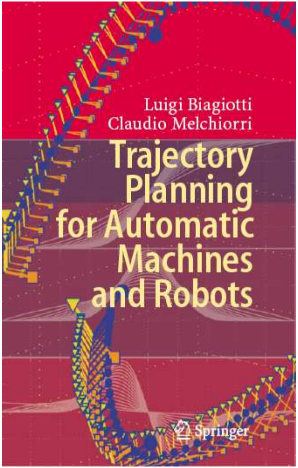


Trajectory planning

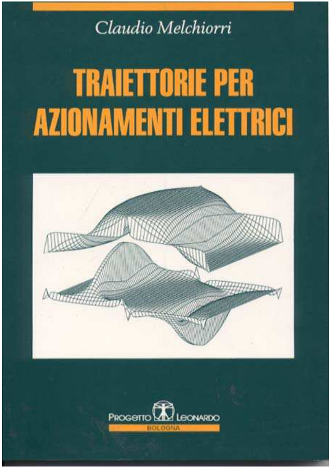
Some suggested references:

- C. Melchiorri, *Traiettorie per azionamenti elettrici*, Progetto Leonardo, Esculapio Ed., Bologna, Feb. 2000;
- G. Canini, C. Fantuzzi, *Controllo del moto per macchine automatiche*, Pitagora Ed., Bo, 2003;
- G. Legnani, M. Tiboni, R. Adamini, *Meccanica degli azionamenti: Vol. 1 - Azionamenti elettrici*, Progetto Leonardo, Esculapio Ed., Bologna, Feb. 2002.
- L. Biagiotti, C. Melchiorri, *Trajectory Planning for Automatic Machines and Robots*, Springer, 2008.

Trajectory planning



Springer, 2008



Esculapio, 2000

Trajectory planning

The planning modalities for trajectories may be quite different:

- point-to-point
- with pre-defined path

Or:

- in the joint space;
- in the work space, either defining some points of interest (initial and final points, *via points*) or the whole geometric path $\mathbf{x} = \mathbf{x}(t)$.

For planning a desired trajectory, it is necessary to specify two aspects:

- *geometric path*
- *motion law*

with constraints on the continuity (smoothness) of the trajectory and on its time-derivatives up to a given degree.

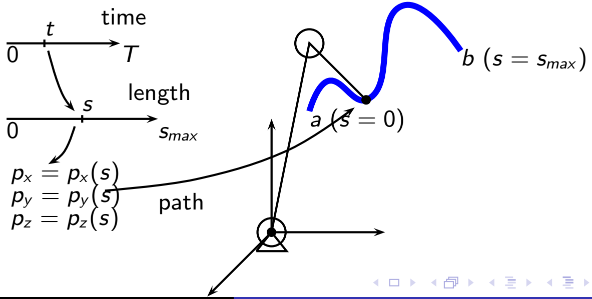
Geometric path and motion law

The geometric path can be defined in the work-space or in the joint-space. Usually, it is expressed in a parametric form as

$$p = p(s) \qquad \text{work-space}$$

$$q = q(\sigma) \qquad \text{joint-space}$$

The parameter s (σ) is defined as a function of time, and in this manner the motion law $s = s(t)$ ($\sigma = \sigma(t)$) is obtained.



Geometric path and motion law

Examples of geometric paths: (in the work space) linear, circular or parabolic segments or, more in general, tracts of analytical functions.

In the joint space, geometric paths are obtained by assigning initial/final (and, in case, also intermediate) values for the joint variables, along with the desired motion law.

Concerning the motion law, it is necessary to specify continuos functions up to a given order of derivations (often at least first and second order, i.e. velocity and acceleration).

Usually, polynomial functions a of proper degree n are employed:

$$s(t) = a_0 + a_1 t + a_2 t^2 + \ldots + a_n t^n$$

In this manner, a "smooth" interpolation of the points defining the geometric path is achieved.

Trajectory planning

Input data to an algorithm for trajectory planning are:

- data defining on the path (points),
- geometrical constraints on the path (e.g. obstacles),
- constraints on the mechanical dynamics
- constraints due to the actuation system

Output data is:

- the trajectory in the joint- or work-space, given as a sequence (in time) of the acceleration, velocity and position values:

$$a(kT), \qquad v(kT), \qquad p(kT) \qquad k = 0, \dots, N$$

being T a proper time interval defining the instants in which the trajectory is computed (and converted in the joint space) and sent to each actuator.

Trajectory planning

Usually, the user has to specify only a minimum amount of information about the trajectory, such as initial and final points, duration of the motion, maximum velocity, and so on.

- *Work-space trajectories* allow to consider directly possible constraints on the path (obstacles, path geometry, . . .) that are more difficult to take into consideration in the joint space (because of the non linear kineamtics)
- *Joint space trajectories* are computationally simpler and allow to consider problems due to singular configurations, actuation redundancy, velocity/acceleration constraints.

Joint-space trajectories

Trajectories are specified by defining some characteristic points:

- directly assigned by some specifications
- assigned by defining desired configurations $\mathbf{x}$ in the work-space, which are then converted in the joint space using the inverse kinematic model.

The algorithm that computes a function $\mathbf{q}(t)$ *interpolating* the given points is characterized by the following features:

- trajectories must be computationally efficient
- the position and velocity profiles (at least) must be continuos functions of time
- undesired effects (such as non regular curvatures) must be minimized or completely avoided.

In the following discussion, a single joint is considered.

If more joints are present, a coordinated motion must be planned, e.g. considering for each of them the same initial and final time instant, or evaluating the most stressed joint (with the largest displacement) and then scaling suitably the motion of the remaining ones.

Polynomial trajectories

In the most simple cases, (a segment of) a trajectory is specified by assigning initial and final conditions on: time (duration), position, velocity, acceleration, ... Then, the problem is to determine a function

$$q = q(t) \qquad \text{or} \qquad q = q(\sigma), \qquad \sigma = \sigma(t)$$

so that those conditions are satisfied.

This is a boundary condition problem, that can be easily solved by considering polynomial functions such as:

$$q(t) = a_0 + a_1 t + a_2 t^2 + \ldots + a_n t^n$$

The degree n (3, 5, ...) of the polynomial depends on the number of boundary conditions that must be verified and on the desired "smoothness" of the trajectory.

Polynomial trajectories

In general, besides the initial and final values, other constraints could be specified on the values of some time-derivatives (velocity, acceleration, jerk, ...) in generic instants t_j. In other terms, one could be interested in defining a polynomial function $q(t)$ whose k-th derivative has a specified value $q^k(t_j)$ at a given istant t_j.

Mathematically, these conditions may be expressed as:

$$k!a_k + (k+1)!a_{k+1}t_j + \ldots + \frac{n!}{(n-k)!}a_n t_j^{n-k} = q^k(t_j)$$

or, in matrix form:

$$\mathbf{M}\ \mathbf{a} = \mathbf{b}$$

where:

- $\mathbf{M}$ is a known $(n+1) \times (n+1)$ matrix,
- $\mathbf{b}$ is the vector with the $n+1$ constraints on the trajectory (known data),
- $\mathbf{a} = [a_0, a_1, \ldots, a_n]^T$ contains the unknown parameters to be computed

$$\mathbf{a} = \mathbf{M}^{-1}\mathbf{b}$$

Third-order polynomial trajectories

Given an initial and a final instant t_i, t_f, a (segment of a) trajectory may be specified by assigning initial and final conditions:

- initial position and velocity q_i, $\dot{q}_i$;
- final position and velocity q_f, $\dot{q}_f$

There are four boundary conditions, and therefore a polynomial of degree 3 (at least) must be considered

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

where the four parameters a_0, a_1, a_2, a_3 must be defined so that the boundary conditions are satisfied.From the boundary conditions, it follows that

$$\begin{array}{rcrcl} q(t_i) &=& a_0 + a_1 t_i + a_2 t_i^2 + a_3 t_i^3 &=& q_i \\ \dot{q}(t_i) &=& a_1 + 2a_2 t_i + 3a_3 t_i^2 &=& \dot{q}_i \\ q(t_f) &=& a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 &=& q_f \\ \dot{q}(t_f) &=& a_1 + 2a_2 t_f + 3a_3 t_f^2 &=& \dot{q}_f \end{array} \tag{2}$$

Third-order polynomial trajectories

In order to solve these equations, let us assume for the moment that $t_i = 0$. Therefore:

$$a_0 = q_i \tag{3}$$

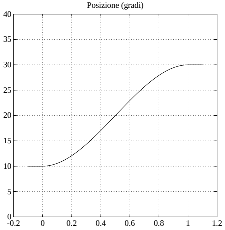
$$a_1 = \dot{q}_i \tag{4}$$

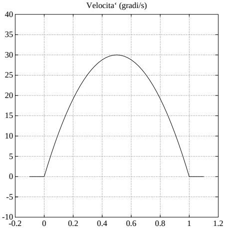
$$a_2 = \frac{-3(q_i - q_f) - (2\dot{q}_i + \dot{q}_f)t_f}{t_f^2} \tag{5}$$

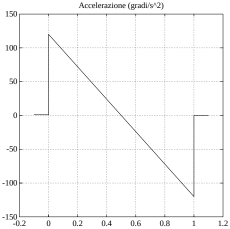
$$a_3 = \frac{2(q_i - q_f) + (\dot{q}_i + \dot{q}_f)t_f}{t_f^3} \tag{6}$$

Third-order polynomial trajectories

Position, velocity and acceleration profiles obtained with a cubic polynomial and boundary conditions: $q_i = 10^o, q_f = 30^o$, $\dot{q}_i = \dot{q}_f = 0\ ^o/s$, $\quad t_i = 0, t_f = 1s$:





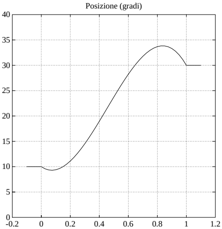


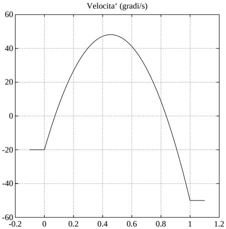
Obviously:

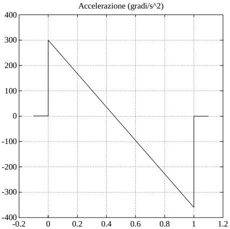
position	$\rightarrow$	cubic function
velocity	$\rightarrow$	parabolic function
acceleration	$\rightarrow$	linear function

Third-order polynomial trajectories

Non null initial velocity: $q_i = 10^o, q_f = 30^o$, $\dot{q}_i = -20\ ^o/s, \dot{q}_f = -50\ ^o/s$, $t_i = 0, t_f = 1s$.







Third-order polynomial trajectories

The results obtained with the polynomial (1) and the coefficients (3)-(6) can be generalized to the case in which $t_i \neq 0$. One obtains:

$$q(t) = a_0 + a_1(t - t_i) + a_2(t - t_i)^2 + a_3(t - t_i)^3 \qquad t_i \leq t \leq t_f$$

with coefficients

$$\begin{aligned} a_0 &= q_i \\ a_1 &= \dot{q}_i \\ a_2 &= \frac{-3(q_i - q_f) - (2\dot{q}_i + \dot{q}_f)(t_f - t_i)}{(t_f - t_i)^2} \\ a_3 &= \frac{2(q_i - q_f) + (\dot{q}_i + \dot{q}_f)(t_f - t_i)}{(t_f - t_i)^3} \end{aligned}$$

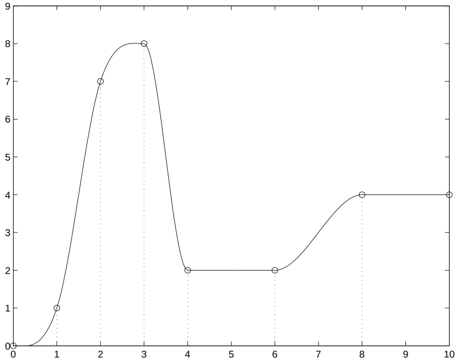
In this manner, it is very simple to plan a trajectory passing through a sequence of intermediate points.

Third-order polynomial trajectories

The trajectory is divided in n segments, each of them defined by:

- initial and final point q_k e q_{k+1}
- initial and final instant t_k, t_{k+1} $\qquad k = 0, \ldots, n-1.$
- initial and final velocity $\dot{q}_k, \dot{q}_{k+1}$

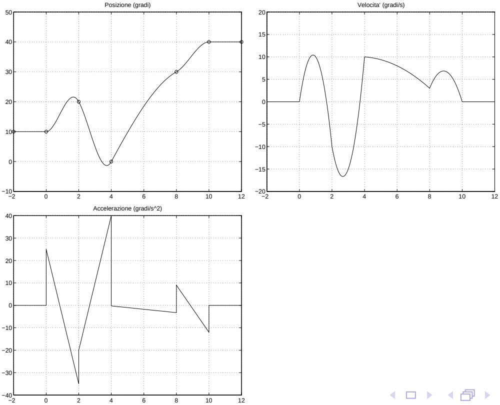
The above relationships are then adopted for each of these segments.



Third-order polynomial trajectories

Position, velocity and acceleration profiles with:

$t_0 = 0$	$t_1 = 2$	$t_2 = 4$	$t_3 = 8$	$t_4 = 10$
$q_0 = 10^o$	$q_1 = 20^o$	$q_2 = 0^o$	$q_3 = 30^o$	$q_4 = 40^o$
$\dot{q}_0 = 0^o/s$	$\dot{q}_1 = -10^o/s$	$\dot{q}_2 = 20^o/s$	$\dot{q}_3 = 3^o/s$	$\dot{q}_4 = 0^o/s$



Third-order polynomial trajectories

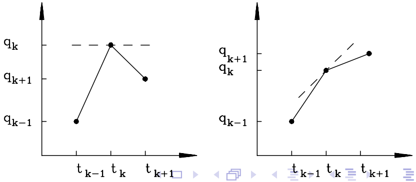
Often, a trajectory is assigned by specifying a sequence of desired points (*via-points*) without indication on the velocity in these points.
In these cases, the "most suitable" values for the velocities must be automatically computed.
This assignment is quite simple with heuristic rules such as:

$$\begin{aligned} \dot{q}_1 &= 0; \\ \dot{q}_k &= \begin{cases} 0 & \text{sign}(v_k) \neq \text{sign}(v_{k+1}) \\ \frac{1}{2}(v_k + v_{k+1}) & \text{sign}(v_k) = \text{sign}(v_{k+1}) \end{cases} \\ \dot{q}_n &= 0 \end{aligned}$$

being

$$v_k = \frac{q_k - q_{k-1}}{t_k - t_{k-1}}$$

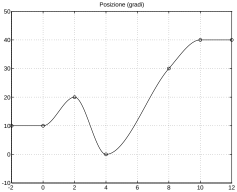
the 'slope' of the tract $[t_{k-1} - t_k]$.

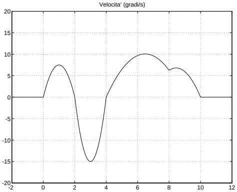


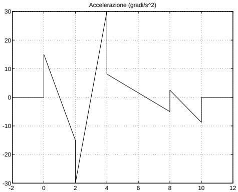
Third-order polynomial trajectories

Automatic computation of the intermediate velocities (data as in the previous example)

$t_0 = 0$	$t_1 = 2$	$t_2 = 4$	$t_3 = 8$	$t_4 = 10$
$q_0 = 10^o$	$q_1 = 20^o$	$q_2 = 0^o$	$q_3 = 30^o$	$q_4 = 40^o$







Fifth-order polynomial trajectories

From the above examples, it may be noticed that both the position and velocity profiles are continuous functions of time.

This is not true for the acceleration, that presents therefore discontinuities among different segments. Moreover, it is not possible to specify for this signal suitable initial/final values in each segment.

In many applications, these aspects do not constitute a problem, being the trajectories "smooth" enough.

On the other hand, if it is requested to specify initial and final values for the acceleration (e.g. for obtaining acceleration profiles), then (at least) fifth-order polynomial functions should be considered

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5$$

with the six boundary conditions:

$$\begin{aligned} q(t_i) &= q_i & \qquad q(t_f) &= q_f \\ \dot{q}(t_i) &= \dot{q}_i & \qquad \dot{q}(t_f) &= \dot{q}_f \\ \ddot{q}(t_i) &= \ddot{q}_i & \qquad \ddot{q}(t_f) &= \ddot{q}_f \end{aligned}$$

Fifth-order polynomial trajectories

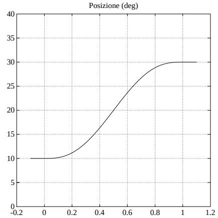
In this case, (if $T = t_f - t_i$) the coefficients of the polynomial are

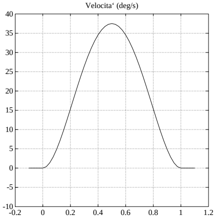
$$
\begin{aligned}
a_0 &= q_i \\
a_1 &= \dot{q}_i \\
a_2 &= \frac{1}{2}\ddot{q}_i \\
a_3 &= \frac{1}{2T^3}[20(q_f - q_i) - (8\dot{q}_f + 12\dot{q}_i)T - (3\ddot{q}_f - \ddot{q}_i)T^2] \\
a_4 &= \frac{1}{2T^4}[30(q_i - q_f) + (14\dot{q}_f + 16\dot{q}_i)T + (3\ddot{q}_f - 2\ddot{q}_i)T^2] \\
a_5 &= \frac{1}{2T^5}[12(q_f - q_i) - 6(\dot{q}_f + \dot{q}_i)T - (\ddot{q}_f - \ddot{q}_i)T^2]
\end{aligned}
$$

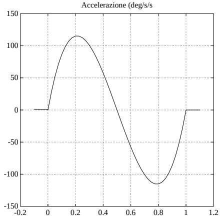
If a sequence of points is given, the same considerations made for third-order polynomials trajectories can be made in computing the intermediate velocity values.

Fifth-order polynomial trajectories

Fifth-order trajectory with the boundary conditions:
$q_i = 10^o, q_f = 30^o, \dot{q}_i = \dot{q}_f = 0\ ^o/s, \ddot{q}_i = \ddot{q}_f = 0\ ^o/s^2, t_i = 0s, t_f = 1s.$





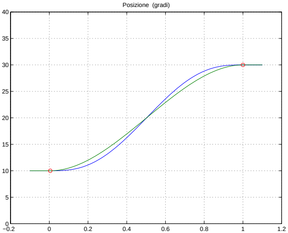


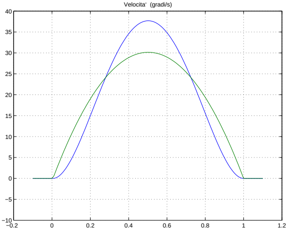
Obviously:

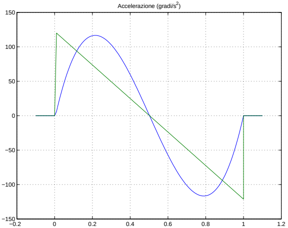
position	$\rightarrow$	5-th order function
velocity	$\rightarrow$	4-th order function
acceleration	$\rightarrow$	3-rd order function

Fifth-order polynomial trajectories

Comparison of fifth- and third-order trajectories with the boundary conditions: $q_i = 10^o, q_f = 30^o, \dot{q}_i = \dot{q}_f = 0\ ^o/s, \ddot{q}_i = \ddot{q}_f = 0\ ^o/s^2, t_i = 0s, t_f = 1s$.



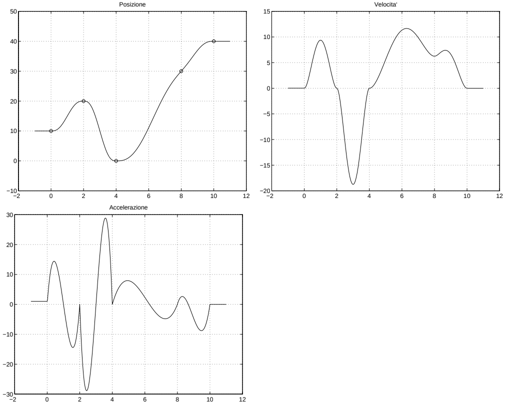




blue → fifth-order
green → third-order

Fifth-order polynomial trajectories

Position, velocity, acceleration profiles with automatic assignment of the intermediate velocities and null accelerations.



Note that the resulting motion has smoother profiles.

Trapezoidal trajectories

A different approach for planning a trajectory is to compute linear segments joined with parabolic blends.

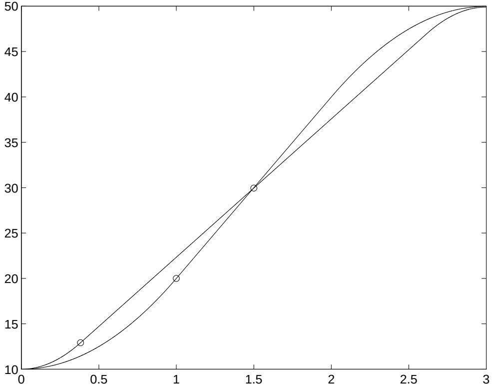
In the linear tract, the velocity is constant while, in the parabolic blends, it is a linear function of time: trapezoidal profiles, typical of this type of trajectory, are then obtained.

In trapezoidal trajectories, the duration is divided into three parts:

1. in the first part, a constant acceleration is applied, then the velocity is linear and the position a parabolic function of time
2. in the second, the acceleration is null, the velocity is constant and the position is linear in time
3. in the last part a (negative) acceleration is applied, then the velocity is a negative ramp and the position a parabolic function.

Trapezoidal trajectories

Usually, the acceleration and the deceleration phases have the same duration ($t_a = t_d$). Therefore, symmetric profiles, with respect to a central instant $(t_f - t_i)/2$, are obtained.



Trapezoidal trajectories

The trajectory is computed according to the following equations.

1) **Acceleration phase, $t \in [0 \div t_a]$.**
The position, velocity and acceleration are described by

$$
\begin{aligned}
q(t) &= a_0 + a_1 t + a_2 t^2 \\
\dot{q}(t) &= a_1 + 2 a_2 t \\
\ddot{q}(t) &= 2 a_2
\end{aligned}
$$

The parameters are defined by constraints on the initial position q_i and the velocity $\dot{q}_i$, and on the desired constant velocity $\dot{q}_v$ that must be obtained at the end of the acceleration period. Assuming a null initial velocity and considering $t_i = 0$ one obtains

$$
\begin{aligned}
a_0 &= q_i \\
a_1 &= 0 \\
a_2 &= \frac{\dot{q}_v}{2 t_a}
\end{aligned}
$$

In this phase, the acceleration is constant and equal to $\dot{q}_v / t_a$.

Trapezoidal trajectories

2) **Constant velocity phase, $t \in [t_a \div t_f - t_a]$.**
Position, velocity and acceleration are now defined as

$$\begin{aligned} q(t) &= b_0 + b_1 t \\ \dot{q}(t) &= b_1 \\ \ddot{q}(t) &= 0 \end{aligned}$$

where, because of continuity,

$$b_1 = \dot{q}_v$$

Moreover, the following equation must hold

$$q(t_a) = q_i + \dot{q}_v \frac{t_a}{2} = b_0 + \dot{q}_v t_a$$

and then

$$b_0 = q_i - \dot{q}_v \frac{t_a}{2}$$

Trapezoidal trajectories

3) **Deceleration phase, $t \in [t_f - t_a \div t_f]$.**

The position, velocity and acceleration are given by

$$
\begin{aligned}
q(t) &= c_0 + c_1 t + c_2 t^2 \\
\dot{q}(t) &= c_1 + 2 c_2 t \\
\ddot{q}(t) &= 2 c_2
\end{aligned}
$$

The parameters are now defined with constrains on the final position q_f and velocity $\dot{q}_f$, and on the velocity $\dot{q}_v$ at the beginning of the deceleration period.

If the final velocity is null, then:

$$
\begin{aligned}
c_0 &= q_f - \frac{\dot{q}_v}{2} \frac{t_f^2}{t_a} \\
c_1 &= \dot{q}_v \frac{t_f}{t_a} \\
c_2 &= -\frac{\dot{q}_v}{2 t_a}
\end{aligned}
$$

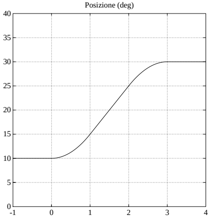
Trapezoidal trajectories

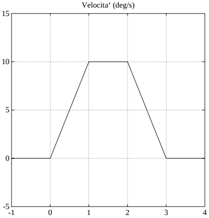
Summarizing, the trajectory is computed as

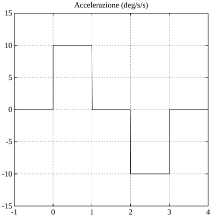
$$q(t) = \begin{cases} q_i + \dfrac{\dot{q}_v}{2t_a} t^2 & 0 \le t < t_a \\ q_i + \dot{q}_v (t - \frac{t_a}{2}) & t_a \le t < t_f - t_a \\ q_f - \dfrac{\dot{q}_v}{t_a} \dfrac{(t_f - t)^2}{2} & t_f - t_a \le t \le t_f \end{cases}$$

Trapezoidal trajectories

Typical position, velocity and acceleration profiles of a trapezoidal trajectory.







Trapezoidal trajectories

Some additional constraints must be specified in order to solve the previous equations.
A typical constraint concerns the duration of the acceleration/deceleration periods t_a that, for symmetry, must satisfy the condition

$$t_a \leq t_f/2$$

Moreover, the following condition must be verified (for the sake of simplicity, consider $t_i = 0$):

$$\begin{aligned} \ddot{q} t_a &= \frac{q_m - q_a}{t_m - t_a} \\ q_a &= q_i + \frac{1}{2}\ddot{q} t_a^2 \end{aligned} \qquad \left\{ \begin{array}{lcl} q_a &=& q(t_a) \\ q_m &=& (q_i + q_f)/2 \\ t_m &=& t_f/2 \end{array} \right.$$

from which

$$\ddot{q} t_a^2 - \ddot{q} t_f t_a + (q_f - q_i) = 0 \tag{7}$$

Finally:

$$\dot{q}_v = \frac{q_f - q_i}{t_f - t_a}$$

Trapezoidal trajectories

Any pair of values $(\ddot{q}, t_a)$ verifying (7) can be considered.

Given the acceleration $\ddot{q}$ (for example $\ddot{q}_{max}$), then

$$t_a = \frac{t_f}{2} - \frac{\sqrt{\ddot{q}^2 t_f^2 - 4\ddot{q}(q_f - q_i)}}{2\ddot{q}}$$

from which we have also that the minimum value for the acceleration is

$$|\ddot{q}| \geq \frac{4|q_f - q_i|}{t_f^2}$$

if the value $|\ddot{q}| = \frac{4|q_f - q_i|}{t_f^2}$ is assigned, then $t_a = t_f/2$ and the constant velocity tract does not exist.

If the value $t_a = t_f/3$ is specified, the following velocity and acceleration values are obtained

$$\dot{q}_v = \frac{3(q_f - q_i)}{2t_f} \qquad \ddot{q} = \frac{9(q_f - q_i)}{2t_f^2}$$

Trapezoidal trajectories

Another way to compute this type of trajectory is to define a maximum value $\ddot{q}_a$ for the desired acceleration and then compute the relative duration t_a of the acceleration and deceleration periods.

If the maximum values ($\ddot{q}_{max}$ and $\dot{q}_{max}$, known) for the acceleration and velocity must be reached, it is possible to assign

$$\left\{\begin{array}{rcll} t_a &=& \dfrac{\dot{q}_{max}}{\ddot{q}_{max}} & \text{acceleration time} \\ \dot{q}_{max}(T - t_a) &=& q_f - q_i = L & \text{displacement} \\ T &=& \dfrac{L\ddot{q}_{max} + \dot{q}_{max}^2}{\ddot{q}_{max}\dot{q}_{max}} & \text{time duration} \end{array}\right.$$

and then ($t_f = t_i + T$)

$$q(t) = \begin{cases} q_i + \frac{1}{2}\ddot{q}_{max}(t - t_i)^2 & t_i \le t \le t_i + t_a \\ q_i + \ddot{q}_{max} t_a (t - t_i - \frac{t_a}{2}) & t_i + t_a < t \le t_f - t_a \\ q_f - \frac{1}{2}\ddot{q}_{max}(t_f - t - t_i)^2 & t_f - t_a < t \le t_f \end{cases} \tag{8}$$

Trapezoidal trajectories

In this case, the linear tract exists if and only if

$$L \geq \frac{\dot{q}_{max}^2}{\ddot{q}_{max}}$$

Otherwise

$$\begin{cases} t_a = \sqrt{\dfrac{L}{\ddot{q}_{max}}} & \text{acceleration time} \\ T = 2t_a & \text{total time duration} \end{cases}$$

and (still $t_f = t_i + T$)

$$q(t) = \begin{cases} q_i + \frac{1}{2}\ddot{q}_{max}(t - t_i)^2 & t_i \leq t \leq t_i + t_a \\ q_f - \frac{1}{2}\ddot{q}_{max}(t_f - t)^2 & t_f - t_a < t \leq t_f \end{cases} \tag{9}$$

Trapezoidal trajectories

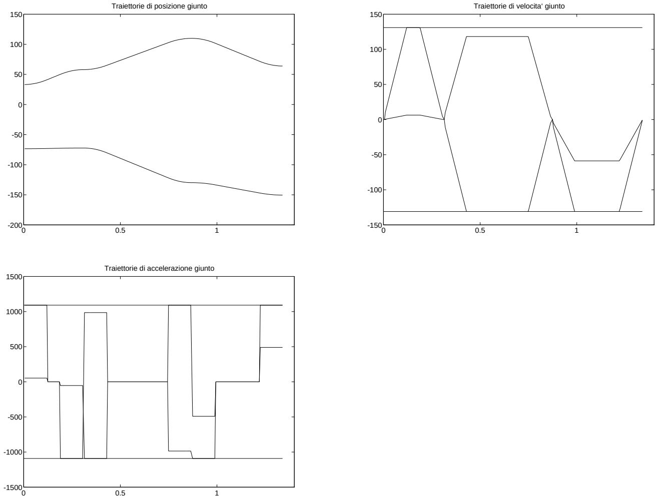
With this modality for computing the trajectory, the time duration of the motion from q_i to q_f is not specified. In fact, the period T is computed on the basis of the maximum acceleration and velocity values.

If more joints have to be co-ordinated with the same constraints on the maximum acceleration and velocity, the joint with the largest displacement must be individuated. For this joint, the maximum value $\ddot{q}_{max}$ for the acceleration is assigned and then the corresponding values t_a and T are computed.

For the remaining joints, the acceleration and velocity values must be computed on the basis of these values of t_a and T, and on the basis of the given displacement L_i:

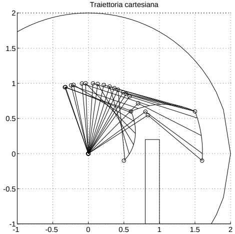
$$\ddot{q}_i = \frac{L_i}{t_a(T - t_a)}, \qquad \dot{q}_i = \frac{L_i}{T - t_a}$$

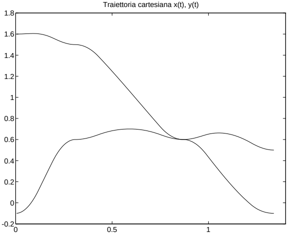
Trapezoidal trajectories

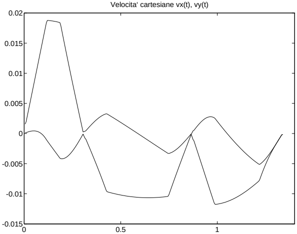


Trapezoidal trajectories

The trajectories in the workspace are:

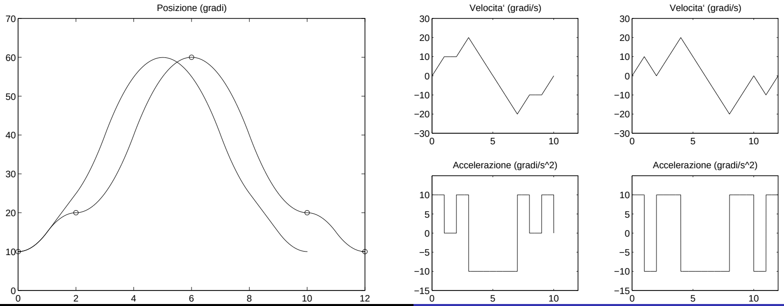






Trapezoidal trajectories

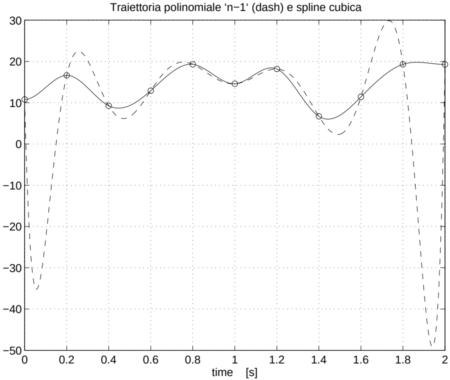
If a trajectory interpolating more consecutive points is computed with the above technique, a motion with null velocities in the via-points is obtained. Since this behavior may be undesirable, it is possible to “anticipate” the actuation of a tract of the trajectory between points q_k and q_{k+q} *before* the motion from q_{k-1} to q_k is terminated. This is possible by adding (starting at an instant $t_k - t'_a$) the velocity and acceleration contributions of the two segments $[q_{k-1} - q_k]$ and $[q_k - q_{k+1}]$.
Obviously, another possibility is to compute the parameters of the functions defining the trapezoidal trajectory in order to have desired boundary conditions (i.e. velocities) for each segment.



Spline

In general, the problem of defining a function interpolating a set of n points can be solved with a polynomial function of degree $n - 1$.

In planning a trajectory, this approach does not give good results since the resulting motions in general present large oscillations.



In general, given:

2 points $\Longrightarrow$ unique line

3 points $\Longrightarrow$ unique quadric

...

n points $\Longrightarrow$ unique polynomial with degree $n - 1$

Spline

The (unique) polynomial $p(x)$ with degree $n-1$ interpolating n points (x_i, y_i) can be computed by the *Lagrange* expression:

$$\begin{aligned} p(x) \quad = \quad & \frac{(x-x_2)(x-x_3)\cdots(x-x_n)}{(x_1-x_2)(x_1-x_3)\cdots(x_1-x_n)} y_1 + \frac{(x-x_1)(x-x_3)\cdots(x-x_n)}{(x_2-x_1)(x_2-x_3)\cdots(x_2-x_n)} y_2 + \cdots \\ & + \cdots + \frac{(x-x_1)(x-x_2)\cdots(x-x_{n-1})}{(x_n-x_1)(x_n-x_2)\cdots(x_n-x_{n-1})} y_n \end{aligned}$$

Other (recursive) expressions have been defined, more efficient from a computational point of view (*Neville formulation*).

Spline

Another (less efficient) approach for the computation of the coefficients of the polynomial $p(x)$ is based on the following procedure:

$$y_i = p(x_i) = a_{n-1} x_i^{n-1} + \cdots + a_1 x_i + a_0 \qquad i = 1, \ldots, n$$

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-1} \\ y_n \end{bmatrix} = \begin{bmatrix} x_1^{n-1} & x_1^{n-2} & \cdots & x_1 & 1 \\ x_2^{n-1} & x_2^{n-2} & \cdots & x_2 & 1 \\ & & \vdots & & \\ x_{n-1}^{n-1} & x_{n-1}^{n-2} & \cdots & x_{n-1} & 1 \\ x_n^{n-1} & x_n^{n-2} & \cdots & x_n & 1 \end{bmatrix} \begin{bmatrix} a_{n-1} \\ a_{n-2} \\ \vdots \\ a_1 \\ a_0 \end{bmatrix} = \mathbf{X}\mathbf{a}$$

and then, by inverting matrix $\mathbf{X}$, the parameters are obtained

$$\mathbf{a} = \mathbf{X}^{-1}\mathbf{y}$$

$\Longrightarrow$ **Numerical problems in computing $\mathbf{X}^{-1}$ for high values of n!!!**

Spline

Given n points, in order to avoid the problem of high 'oscillations' (and also of the numerical precision):

NO: one polynomial of degree $n-1$

$\Longrightarrow$ **YES: $n-1$ polynomials with lower degree p $(p < n-1)$: each polynomial interpolates a segment of the trajectory.**

Usually, the degree p of the $n-1$ polynomials is chosen so that continuity of the velocity and acceleration profile is achieved. In this case, the choice $p = 3$ is made (cubic polynomials):

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

There are 4 coefficients for each polynomial, and therefore it is necessary to compute $4(n-1)$ coefficients.

Obviously, it is possible to choose higher values for p (e.g. $p = 5, 7, \ldots$).

Spline

$4(n-1)$ coefficients

On the other hand, there are:

- $2(n-1)$ conditions on the position (each cubic function interpolates the initial/final points);
- $n-2$ conditions on the continuity of velocity in the intermediate points;
- $n-2$ conditions on the continuity of acceleration in the intermediate points.

Therefore, there are

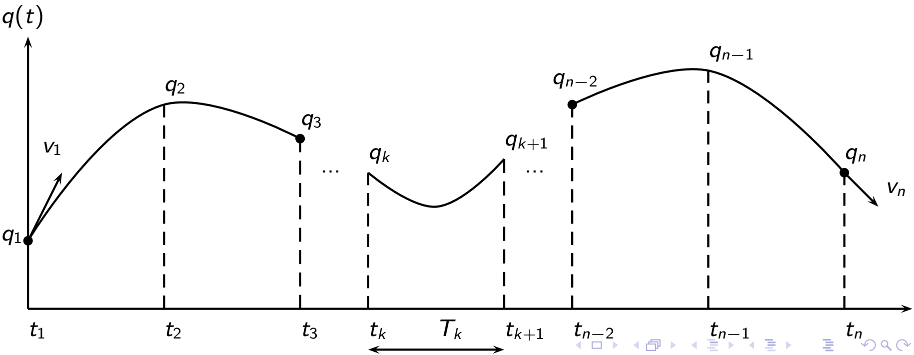
$$4(n-1) - 2(n-1) - 2(n-2) = 2$$

degrees of freedom left, that can be used *for example* for imposing proper conditions on the initial and final velocity.

Spline

The function obtained in this manner is a *spline*.

Among all the interpolating functions of n points with the same degree of continuity of derivation, the spline has the smallest *curvature*.



Spline

Mathematically, it is necessary to compute a function

$$\begin{cases} q(t) = \{q_k(t), \quad t \in [t_k,\ t_{k+1}], \quad k = 1, \ldots, n-1\} \\ q_k(\tau) = a_{k0} + a_{k1}\tau + a_{k2}\tau^2 + a_{k3}\tau^3, \quad \tau \in [0,\ T_k], \\ \qquad (\tau = t - t_k, \quad T_k = t_{k+1} - t_k) \end{cases}$$

with the conditions

$$q_k(0) = q_k, \quad q_k(T_k) = q_{k+1} \qquad k = 1, \ldots, n-1$$

$$\dot{q}_k(T_k) = \dot{q}_{k+1}(0) = v_k \qquad k = 1, \ldots, n-2$$

$$\ddot{q}_k(T_k) = \ddot{q}_{k+1}(0) \qquad k = 1, \ldots, n-2$$

Spline - Computation

The parameters a_{ki} are computed according to the following algorithm.

Let assume that the velocities $v_k, \; k = 2, \ldots, n-1$ in the intermediate points are known.

In this case, we can impose for each cubic polynomial the four boundary conditions on position and velocity:

$$\begin{cases} q_k(0) &= a_{k0} &= q_k \\ \dot{q}_k(0) &= a_{k1} &= v_k \\ q_k(T_k) &= a_{k0} + a_{k1}T + a_{k2}T^2 + a_{k3}T^3 &= q_{k+1} \\ \dot{q}_k(T_k) &= a_{k1} + 2a_{k2}T + 3a_{k3}T^2 &= v_{k+1} \end{cases}$$

and then

$$\begin{cases} a_{k0} = q_k \\ a_{k1} = v_k \\ a_{k2} = \dfrac{1}{T_k}\left[\dfrac{3(q_{k+1} - q_k)}{T_k} - 2v_k - v_{k+1}\right] \\ a_{k3} = \dfrac{1}{T_k^2}\left[\dfrac{2(q_k - q_{k+1})}{T_k} + v_k + v_{k+1}\right] \end{cases}$$

... but the velocities v_k are not known...

Spline - Computation

By using the conditions on continuity of the accelerations in the intermediate points, one obtains

$$\ddot{q}_k(T_k) = 2a_{k2} + 6a_{k3}\,T_k = 2a_{k+1,2} = \ddot{q}_{k+1}(0) \qquad k = 1, \dots, n-2$$

form which, by substituting the expressions of $a_{k2}, a_{k3}, a_{k+1,2}$ and multiplying by $(T_k\,T_{k+1})/2$, one obtains

$$T_{k+1} v_k + 2(T_{k+1} + T_k) v_{k+1} + T_k v_{k+2} = \frac{3}{T_k\,T_{k+1}} \left[T_k^2 (q_{k+2} - q_{k+1}) + T_{k+1}^2 (q_{k+1} - q_k) \right]$$

These equations may be written in matrix form as

$$\begin{bmatrix} T_2 & 2(T_1+T_2) & T_1 & & & & \\ 0 & T_3 & 2(T_2+T_3) & T_2 & & & \\ & & & \ddots & & & \\ & & & T_{n-2} & 2(T_{n-3}+T_{n-2}) & T_{n-3} & 0 \\ & & & & T_{n-1} & 2(T_{n-2}+T_{n-1}) & T_{n-2} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{n-1} \\ v_n \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{n-3} \\ c_{n-2} \end{bmatrix}$$

where the c_k are (known) constant terms depending on the intermediate positions and the duration of each segments.

Spline - Computation

Since the velocities v_1 and v_n are known, the corresponding columns can be eliminated form the left-hand side matrix, and then

$$\begin{bmatrix} 2(T_1+T_2) & T_1 & & & \\ T_3 & 2(T_2+T_3) & T_2 & & \\ & & \vdots & & \\ & & T_{n-2} & 2(T_{n-3}+T_{n-2}) & T_{n-3} \\ & & & T_{n-1} & 2(T_{n-2}+T_{n-1}) \end{bmatrix} \begin{bmatrix} v_2 \\ \vdots \\ v_{n-1} \end{bmatrix} =$$

$$\begin{bmatrix} \frac{3}{T_1 T_2}\left[T_1^2(q_3-q_2)+T_2^2(q_2-q_1)\right] - \mathbf{T_2 v_1} \\ \frac{3}{T_2 T_3}\left[T_2^2(q_4-q_3)+T_3^2(q_3-q_2)\right] \\ \vdots \\ \frac{3}{T_{n-3} T_{n-2}}\left[T_{n-3}^2(q_{n-1}-q_{n-2})+T_{n-2}^2(q_{n-2}-q_{n-3})\right] \\ \frac{3}{T_{n-2} T_{n-1}}\left[T_{n-2}^2(q_n-q_{n-1})+T_{n-1}^2(q_n-q_{n-2})\right] - \mathbf{T_{n-2} v_n} \end{bmatrix}$$

that is

$$\mathbf{A}(\mathbf{T})\,\mathbf{v} = \mathbf{c}(\mathbf{T}, \mathbf{q}, v_1, v_n)$$

or

$$\mathbf{A}\mathbf{v} = \mathbf{c}$$

Spline - Computation

- The matrix $\mathbf{A}$ is tridiagonal, and is always invertible if $T_k > 0$ ($|a_{kk}| > \sum_{j \neq k} |a_{kj}|$).
- Being $\mathbf{A}$ tridiagonal, its inverse is computed by efficient numerical algorithms (based on the Gauss-Jordan method).
- Once $\mathbf{A}^{-1}$ is known, the velocities $v_2, \ldots, v_{n-1}$ are computed as

$$\mathbf{v} = \mathbf{A}^{-1}\mathbf{c}$$

  and the problem is solved.

Spline

The total duration of a spline is

$$T = \sum_{k=1}^{n-1} T_k = t_n - t_1$$

It is possible to define an optimality problem aiming at minimizing T. The values of T_k must be computed so that T is minimized and the constraints on the velocity and acceleration are satisfied.

Formally the problem is formulated as

$$\begin{cases} \min_{T_k} \quad T = \sum_{k=1}^{n-1} T_k \\ \text{tale che} \qquad \begin{array}{ll} |\dot{q}(\tau, T_k)| < v_{max} & \tau \in [0\, T] \\ |\ddot{q}(\tau, T_k)| < a_{max} & \tau \in [0\, T] \end{array} \end{cases}$$

Non linear optimization problem with linear objective function, solvable with classical techniques from the operational research field.

Spline - Example

A spline trough the points $q_1 = 0$, $q_2 = 2$, $q_3 = 12$, $q_4 = 5$ must be defined, minimizing the total duration T and with the constraints: $v_{max} = 3$, $a_{max} = 2$.

The non linear optimization problem

$$min \quad T = T_1 + T_2 + T_3$$

is defined, with the constraints reported in the following slide.

By solving this problem (e.g. with the Matlab *Optimization Toolbox*) the following values are obtained:

$$T_1 = 1.5549, \qquad T_2 = 4.4451, \qquad T_3 = 4.5826, \qquad \Rightarrow \qquad T = 10.5826\ sec$$

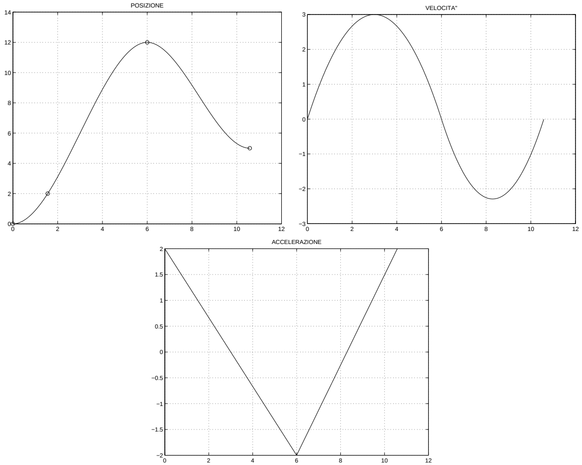
Spline - Esempio

Constraints on the optimization problem:

$$
\left\{
\begin{array}{lllllll}
a_{11} & & & & \leq & v_{max} & \text{(velocità iniziale del I tratto } \leq v_{max}) \\
a_{21} & & & & \leq & v_{max} & \text{(velocità iniziale del II tratto } \leq v_{max}) \\
a_{31} & & & & \leq & v_{max} & \text{(velocità iniziale del III tratto } \leq v_{max}) \\
a_{11} & + & 2a_{12}T_1 & + & 3a_{13}T_1^2 \leq & v_{max} & \text{(velocità finale del I tratto } \leq v_{max}) \\
a_{21} & + & 2a_{22}T_2 & + & 3a_{23}T_2^2 \leq & v_{max} & \text{(velocità finale del II tratto } \leq v_{max}) \\
a_{31} & + & 2a_{32}T_3 & + & 3a_{33}T_3^2 \leq & v_{max} & \text{(velocità finale del III tratto } \leq v_{max}) \\
a_{11} & + & 2a_{12}\left(-\frac{a_{12}}{3a_{13}}\right) & + & 3a_{13}\left(-\frac{a_{12}}{3a_{13}}\right)^2 \leq & v_{max} & \text{(velocità all'interno del I tratto } \leq v_{max}) \\
a_{21} & + & 2a_{22}\left(-\frac{a_{22}}{3a_{23}}\right) & + & 3a_{23}\left(-\frac{a_{22}}{3a_{23}}\right)^2 \leq & v_{max} & \text{(velocità all'interno del II tratto } \leq v_{max}) \\
a_{31} & + & 2a_{32}\left(-\frac{a_{32}}{3a_{33}}\right) & + & 3a_{33}\left(-\frac{a_{32}}{3a_{33}}\right)^2 \leq & v_{max} & \text{(velocità all'interno del III tratto } \leq v_{max}) \\
2a_{12} & & & & \leq & a_{max} & \text{(accelerazione iniziale del I tratto } \leq a_{max}) \\
2a_{22} & & & & \leq & a_{max} & \text{(accelerazione iniziale del II tratto } \leq a_{max}) \\
2a_{32} & & & & \leq & a_{max} & \text{(accelerazione iniziale del III tratto } \leq a_{max}) \\
2a_{12} & & & + & 6a_{13}T_1 \leq & a_{max} & \text{(accelerazione finale del I tratto } \leq a_{max}) \\
2a_{22} & & & + & 6a_{23}T_2 \leq & a_{max} & \text{(accelerazione finale del II tratto } \leq a_{max}) \\
2a_{32} & & & + & 6a_{33}T_3 \leq & a_{max} & \text{(accelerazione finale del III tratto } \leq a_{max})
\end{array}
\right.
$$

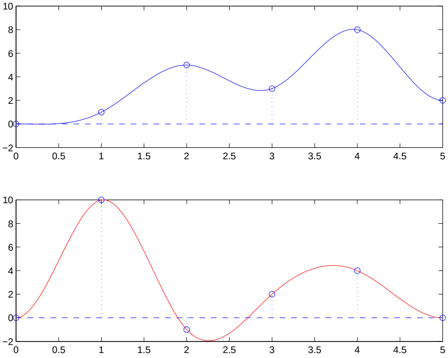
Spline - Esempio

Position, velocity and acceleration profiles of the optimal trajectory.



Spline

The above procedure for computing the spline is adopted also for more motion axes (joints). Notice that the matrix $\mathbf{A}_j(\mathbf{T}) = \mathbf{A}(\mathbf{T})$ is the same for all the joints (depends only on the parameters T_k), while the vector $\mathbf{c}(\mathbf{T}, \mathbf{q}_j, v_{i1}, v_{in})$ depends on the specific i-th joint.



Spline

From the expressions of matrix $\mathbf{A}$ and the vector $\mathbf{c}$ ($\mathbf{A}\mathbf{v} = \mathbf{c}$)

$$\mathbf{A} = \begin{bmatrix} 2(T_1+T_2) & T_1 & & & \\ T_3 & 2(T_2+T_3) & T_2 & & \\ & & \vdots & & \\ & & T_{n-2} & 2(T_{n-3}+T_{n-2}) & T_{n-3} \\ & & & T_{n-1} & 2(T_{n-2}+T_{n-1}) \end{bmatrix}$$

$$\mathbf{c} = \begin{bmatrix} \frac{3}{T_1 T_2}\left[T_1^2(q_3-q_2)+T_2^2(q_2-q_1)\right] - T_2 v_1 \\ \frac{3}{T_2 T_3}\left[T_2^2(q_4-q_3)+T_3^2(q_3-q_2)\right] \\ \vdots \\ \frac{3}{T_{n-3} T_{n-2}}\left[T_{n-3}^2(q_{n-1}-q_{n-2})+T_{n-2}^2(q_{n-2}-q_{n-3})\right] \\ \frac{3}{T_{n-2} T_{n-1}}\left[T_{n-2}^2(q_n-q_{n-1})+T_{n-1}^2(q_n-q_{n-2})\right] - T_{n-2} v_n \end{bmatrix}$$

Spline

If

- the duration T_k of each interval is multiplied by a constant λ (linear scaling)
- the initial and final velocities are null

one obtains that the new duration T', the velocities and accelerations of the new trajectory are:

$$T' = = \lambda T$$

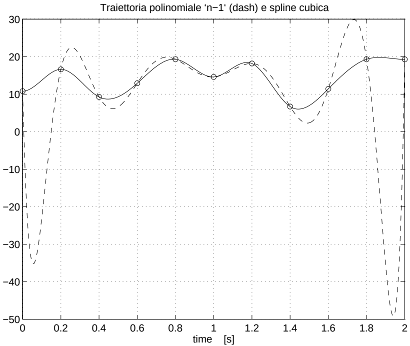
$$v'_k = \frac{1}{\lambda} v_k$$

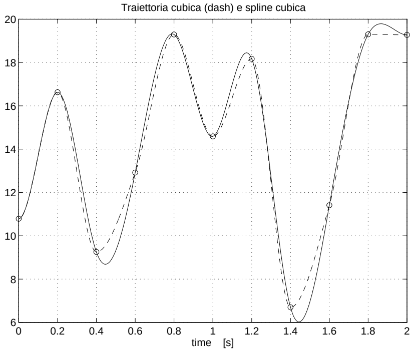
$$a_k = \frac{1}{\lambda^2} a_k$$

Spline - Example

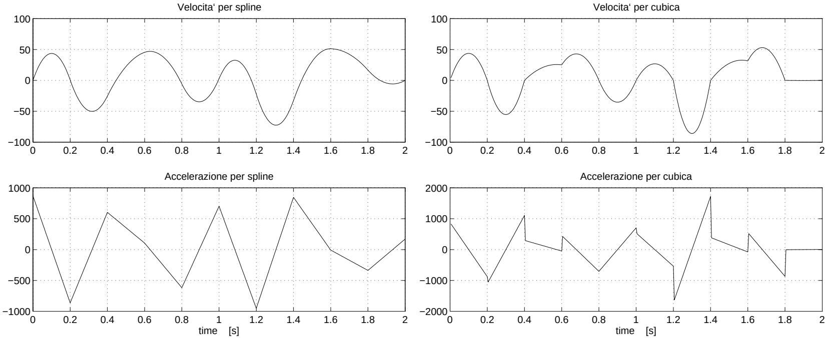
Comparison of a $n-1$ polynomial, a spline, and a composition of cubic polynomials.

11 points, $\quad v_{in} = v_{fin} = 0 \quad m/s$





Spline - Example



Trajectory Planning for Robot Manipulators

Scaling Trajectories

Claudio Melchiorri

Dipartimento di Ingegneria dell'Energia Elettrica e dell'Informazione (DEI)

Università di Bologna

email: claudio.melchiorri@unibo.it

Scaling trajectories

Due to several reasons, like limits on the actuation system (torques, accelerations, velocities, ...) or computational efficiency, it is often requested to *scale* trajectories and motion laws.

It is possible to adopt

- **Kinematic** scaling procedures
- **Dynamic** scaling procedures

Kinematic scaling of trajectories

If a trajectory is expressed in parametric form as a function of a parameter $\sigma = \sigma(t)$, by changing the parameterization it is possible to obtain in a simple manner a trajectory satisfying constraints on velocity or accelerations.

For this purpose, it is convenient to express the trajectory in *normal form*, i.e.:

$$p(t) = p_0 + (p_1 - p_0)s(\tau) = p_0 + Ls(\tau)$$

being $s(\tau)$ a proper parameterization, with

$$0 \le s \le 1, \qquad \tau = \frac{t - t_0}{t_1 - t_0} = \frac{t - t_0}{T}$$

In this manner, it results

$$\frac{dp}{dt} = \frac{L}{T} s'(\tau) \qquad \frac{d^2 p}{dt^2} = \frac{L}{T^2} s''(\tau)$$

$$\frac{d^3 p}{dt^3} = \frac{L}{T^3} s'''(\tau) \qquad \dots$$

$$\frac{d^n p}{dt^n} = \frac{L}{T^n} s^{(n)}(\tau)$$

Kinematic scaling of trajectories

From

$$
\begin{aligned}
\frac{dp}{dt} &= \frac{L}{T}s'(\tau) & \qquad \frac{d^2p}{dt^2} &= \frac{L}{T^2}s''(\tau) \\
\frac{d^3p}{dt^3} &= \frac{L}{T^3}s'''(\tau) & \qquad \dots \\
\frac{d^np}{dt^n} &= \frac{L}{T^n}s^{(n)}(\tau)
\end{aligned}
$$

it follows that the maximum values for the velocity, acceleration, etc. are obtained in correspondence of the maximum values of the functions $s', s'', \dots$.

These values and the corresponding time instants τ (t) are known from the chosen parameterization $s(\tau)$.

Notice that if the duration T of the trajectory is changed, it is possible to satisfy in an exact manner the given constraints or to optimize the trajectory itself (minimum time). Moreover, it is easily possible to co-ordinate more motion axes.

Kinematic scaling of trajectories

Polynomial trajectories of degree 3

Consider a parameterization expressed by a cubic polynomial

$$s(\tau) = a_0 + a_1\tau + a_2\tau^2 + a_3\tau^3, \qquad 0 \le s \le 1, \quad 0 \le \tau \le 1, \quad \tau = \frac{t}{T}$$

If the boundary conditions $p_0 = 0$, $v_0 = 0$, $v_1 = 0$ are specified, one obtains

$$a_0 = 0, \quad a_1 = 0, \quad a_2 = 3, \quad a_3 = -2$$

Therefore:

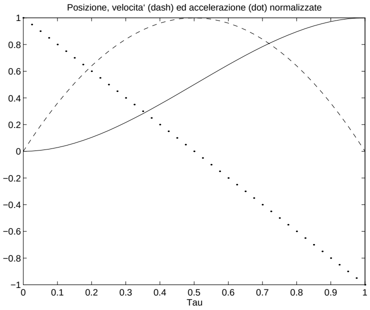
$$\begin{aligned} s(\tau) &= 3\tau^2 - 2\tau^3 \\ s'(\tau) &= 6\tau - 6\tau^2 \\ s''(\tau) &= 6 - 12\tau \\ s'''(\tau) &= -12 \end{aligned}$$

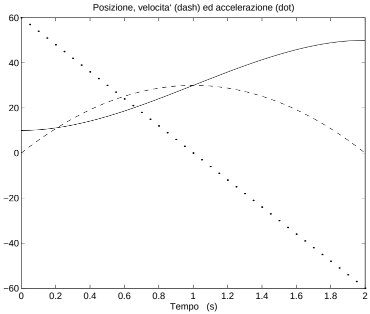
Kinematic scaling of trajectories

Then

$$s'_{max} = s'(0.5) = \frac{3}{2} \qquad \Longrightarrow \qquad \dot{q}_{max} = \frac{3L}{2T}$$

$$s''_{max} = s''(0) = 6 \qquad \Longrightarrow \qquad \ddot{q}_{max} = \frac{6L}{T^2}$$





Kinematic scaling of trajectories

Polynomial trajectories of degree 5

The polynomial $s(\tau)$ in normal form is now:

$$s(\tau) = a_0 + a_1\tau + a_2\tau^2 + a_3\tau^3 + a_4\tau^4 + a_5\tau^5, \qquad 0 \le s \le 1, \quad 0 \le \tau \le 1, \quad \tau = \frac{t}{T}$$

With null boundary conditions on accelerations and velocities, the following values for the parameters are obtained (trajectory 3-4-5)

$$a_0 = 0, \quad a_1 = 0, \quad a_2 = 0, \quad a_3 = 10 \quad a_4 = -14, \quad a_5 = 6$$

Then

$$\begin{aligned} s(\tau) &= 10\tau^3 - 15\tau^4 + 6\tau^5 \\ s'(\tau) &= 30\tau^2 - 60\tau^3 + 30\tau^4 \\ s''(\tau) &= 60\tau - 180\tau^2 + 120\tau^3 \\ s'''(\tau) &= 60 - 360\tau + 360\tau^2 \end{aligned}$$

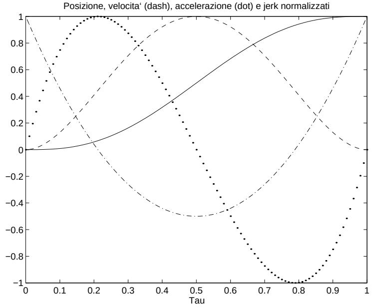
Kinematic scaling of trajectories

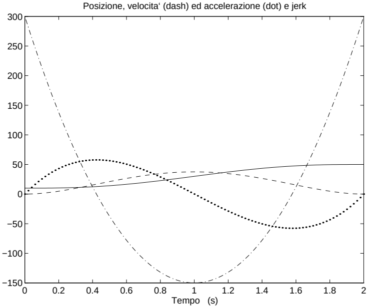
Therefore

$$s'_{max} = s'(0.5) = \frac{15}{8} \quad \Longrightarrow \quad \dot{q}_{max} = \frac{15L}{8T}$$

$$s''_{max} = s''(0.2123) = \frac{10\sqrt{3}}{3} \quad \Longrightarrow \quad \ddot{q}_{max} = \frac{10\sqrt{3}L}{3T^2}$$

$$s'''_{max} = s'''(0) = 60 \quad \Longrightarrow \quad \ddot{q}_{max} = 60\frac{L}{T^3}$$





Kinematic scaling of trajectories

Polynomial trajectories of degree 7

If a continuos jerk profile is requested, a polynomial with higher degree must be adopted. The normal form for a polynomial $s(\tau)$ of degree 7 is:

$$s(\tau) = a_0 + a_1\tau + a_2\tau^2 + a_3\tau^3 + a_4\tau^4 + a_5\tau^5 + a_6\tau^6 + a_7\tau^7$$

If null boundary conditions on velocity, acceleration and jerk are specified, the following parameters are obtained (trajectory 4-5-6-7)

$$a_0 = 0,\ a_1 = 0,\ a_2 = 0,\ a_3 = 0\ a_4 = 35,\ a_5 = -84,\ a_6 = 70,\ a_7 = -20$$

Therefore

$$\begin{aligned}
s(\tau) &= 35\tau^4 - 84\tau^5 + 70\tau^6 - 20\tau^7 \\
s'(\tau) &= 140\tau^3 - 420\tau^4 + 420\tau^5 - 140\tau^6 \\
s''(\tau) &= 420\tau^2 - 1680\tau^3 + 2100\tau^4 - 840\tau^5 \\
s'''(\tau) &= 840\tau - 5040\tau^2 + 8400\tau^3 - 4200\tau^4
\end{aligned}$$

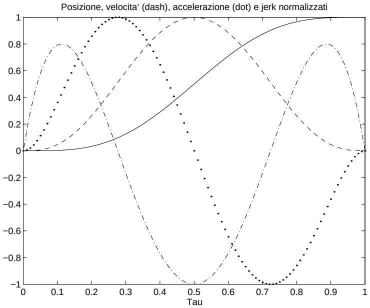
Kinematic scaling of trajectories

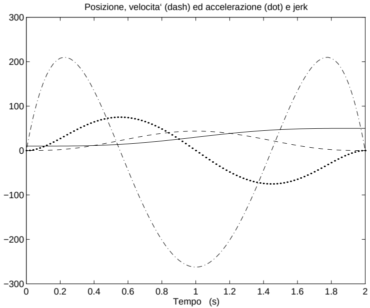
The maximum velocity and acceleration values are obtained for

$$s'_{max} = s'(0.5) = \frac{35}{16} \qquad \Longrightarrow \qquad \dot{q}_{max} = \frac{35L}{16T}$$

$$s''_{max} = s''\left(\frac{5 \pm \sqrt{5}}{10}\right) = \frac{84\sqrt{5}}{25} \qquad \Longrightarrow \qquad \ddot{q}_{max} = \frac{84\sqrt{5}}{25}\,\frac{L}{T^2}$$

$$s'''_{max} = s'''\left(\frac{1+\sqrt{3/5}}{2}\right) = 42, \qquad s'''_{min} = s'''(0.5) = -\frac{105}{2} \qquad \Longrightarrow \qquad \max_{\tau} |s'''| = \frac{105}{2}$$





Considerations on limits and durations of trajectories

From the previous examples, it is clear that if the displacement L and the duration T of a motion are specified, the profiles of velocity, acceleration and jerk are defined by the parameterization $s(\tau)$ chosen to generate the motion profile.
In particular, the maximum values for these variables are determined (for the sake of simplicity, consider the case $L > 0$).

	Pol. 3	Pol. 5	Pol. 7	Cicl.	Harmon.
Vel. $(*L/T)$	$\frac{3}{2} = 1.5$	$\frac{15}{8} = 1.875$	$\frac{35}{16} = 2.1875$	2	$\frac{\pi}{2} = 1.5708$
Acc. $(*L/T^2)$	6	$\frac{10\sqrt{3}}{3} = 5.7735$	$\frac{84\sqrt{5}}{25} = 7.5132$	$2\pi = 6.2832$	$\frac{\pi^2}{2} = 4.9348$
Jerk $(*L/T^3)$	12	60	$\frac{105}{2} = 52.25$	$4\pi^2 = 39.4784$	$\frac{\pi^3}{2} = 15.5031$

Notice that the polynomial of degree 7, originating a very smooth profile, requires higher velocity and acceleration values. Viceversa, the harmonic trajectory has a very good behavior.

Example: scaling a trajectory

Trajectory 3-4-5. Polynomial in *normal form*:

$$s(\tau) = a\tau^5 + b\tau^4 + c\tau^3 + d\tau^2 + e\tau + f$$

with

$$0 \leq s \leq 1, \qquad 0 \leq \tau \leq 1, \qquad \tau = \frac{t}{T}$$

The trajectory is

$$q(t) = q_0 + (q_1 - q_0)s(\tau) = q_0 + Ls(\tau)$$

and

$$\begin{aligned} \dot{q}(t) &= Ls'(\tau)\frac{1}{T} \\ \ddot{q}(t) &= Ls''(\tau)\frac{1}{T^2} \\ \dots \\ \frac{d^n q}{dt^n} &= Ls^{(n)}(\tau)\frac{1}{T^n} \end{aligned}$$

Example: scaling a trajectory

Then (trajectory 3-4-5, $f = e = d = 0$, and $a = 6$, $b = -15$, $c = 10$)

$$\begin{aligned} s'(\tau) &= 30\tau^4 - 60\tau^3 + 30\tau^2 \\ s''(\tau) &= 120\tau^3 - 180\tau^2 + 60\tau \\ s'''(\tau) &= 360\tau^2 - 360\tau + 60 \end{aligned}$$

and

$$s'_{max} = s'(0.5) = \frac{15}{8} \quad \Longrightarrow \quad \dot{q}_{max} = \frac{15L}{8T}$$

$$s''_{max} = s''(0.2123) = \frac{10\sqrt{3}}{3} \quad \Longrightarrow \quad \ddot{q}_{max} = \frac{10\sqrt{3}L}{3T^2}$$

$$\dots$$

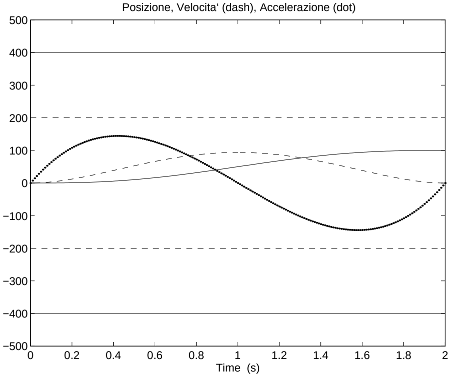
Given constraints on maximum acceleration and velocity, it is possible to properly scaling the trajectory.

Co-ordination of more motion axes made on the basis of the "most stressed" actuator.

Example: scaling a trajectory

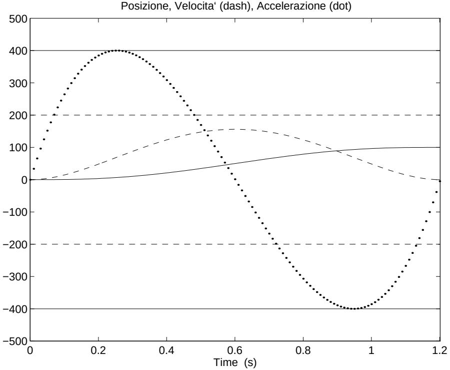
If:

$q_0 = 0;$ $q_1 = 100;$ $t_0 = 0;$ $t_1 = 2;$ $\dot{q}_{max} = 200;$ $\ddot{q}_{max} = 400$



Example: scaling a trajectory

$$T_{min,v} = \frac{15L}{8\dot{q}_{max}} = 0.9375\ s, \qquad T_{min,a} = \sqrt{\frac{10\sqrt{3}L}{3\ddot{q}_{max}}} = 1.2014\ s \qquad T_{min} = \max\{T_{min,v},\ T_{min,a}\}$$



Dynamic scaling of trajectories

When a trajectory is specified for a complex mechanical system, because of the dynamics of the actuation system, of the robot manipulator or of the load (dynamic couplings), torques non physically achievable by the actuators could be requested. In these cases, it is possible to scale the trajectory taking into account the dynamics of the system in order to obtain a physically achievable motion.

The dynamic model of a manipulator is

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}$$

Then, for each joint

$$\mathbf{m}_i^T(\mathbf{q})\ddot{\mathbf{q}} + \frac{1}{2}\dot{\mathbf{q}}^T\mathbf{C}_i(\mathbf{q})\dot{\mathbf{q}} + g_i(\mathbf{q}) = \tau_i \qquad\qquad i = 1, \dots, n$$

If

$$\mathbf{q} = \mathbf{q}(\sigma) \qquad\qquad \sigma = \sigma(t)$$

is a proper parameterization of the trajectory with a motion law such that

$$\dot{\mathbf{q}} = \frac{d\,\mathbf{q}}{d\sigma}\dot{\sigma}, \qquad\qquad \ddot{\mathbf{q}} = \frac{d^2\,\mathbf{q}}{d\sigma^2}\dot{\sigma}^2 + \frac{d\,\mathbf{q}}{d\sigma}\ddot{\sigma}$$

Dynamic scaling of trajectories

By substitution in the dynamic model:

$$\left[m_i^T(\mathbf{q}(\sigma)) \frac{d\ \mathbf{q}}{d\sigma} \right] \ddot{\sigma} + \left[m_i^T(\mathbf{q}(\sigma)) \frac{d^2\ \mathbf{q}}{d\sigma^2} + \frac{1}{2} \frac{d\ \mathbf{q}}{d\sigma}^T \mathbf{C}_i(\mathbf{q}(\sigma)) \frac{d\ \mathbf{q}}{d\sigma} \right] \dot{\sigma}^2 + g_i(\mathbf{q}(\sigma)) = \tau_i$$

from which

$$\alpha_i(\sigma)\ddot{\sigma} + \beta_i(\sigma)\dot{\sigma}^2 + \gamma_i(\sigma) = \tau_i$$

Notice that $\gamma_i(\sigma)$ (gravitational terms) depend on the position only (σ).

Dynamic scaling of trajectories

Let us suppose to compute the torques τ_i necessary to achieve the motion defined by $\mathbf{q} = \mathbf{q}(\sigma)$, $\sigma = \sigma(t)$:

$$\tau_i(t) = \alpha_i(\sigma(t))\ddot{\sigma}(t) + \beta_i(\sigma(t))\dot{\sigma}^2(t) + \gamma_i(\sigma(t)), \qquad i = 1, ..., n, \qquad t \in [0,\ T]$$

If the time-axis is changed (e.g. in a linear fashion $(x = kt)$), a different parameterization of the trajectory is obtained

$$t \rightarrow x = kt \qquad x \in [0,\ kT] \qquad \sigma(t) \rightarrow \hat{\sigma}(x)$$

Notice that in general even a non linear parameterization $x = x(t)$ could be considered.

With the new parameterization, one obtains:

$$\begin{aligned} \hat{\sigma}(x) &= \sigma(t) \\ \dot{\hat{\sigma}}(x) &= \frac{\dot{\sigma}(t)}{k} \\ \ddot{\hat{\sigma}}(x) &= \frac{\ddot{\sigma}(t)}{k^2} \end{aligned}$$

Dynamic scaling of trajectories

Therefore

- if $k > 1$ a *slower* motion is obtained
- if $k < 1$ a *faster* motion is obtained.

With the new parameterization, the torques compute as:

$$\begin{aligned}
\tau_i(x) &= \alpha_i(\hat{\sigma}(x))\ddot{\hat{\sigma}}(x) + \beta_i(\hat{\sigma}(x))\dot{\hat{\sigma}}^2(x) + \gamma_i(\hat{\sigma}(x)) \\
&= \alpha_i(\sigma(t))\frac{\ddot{\sigma}(t)}{k^2} + \beta_i(\sigma(t))\frac{\dot{\sigma}^2(t)}{k^2} + \gamma_i(\sigma(t)) \\
&= \frac{1}{k^2}[\tau_i(t) - \gamma_i(\sigma(t))] + \gamma_i(\sigma(t))
\end{aligned}$$

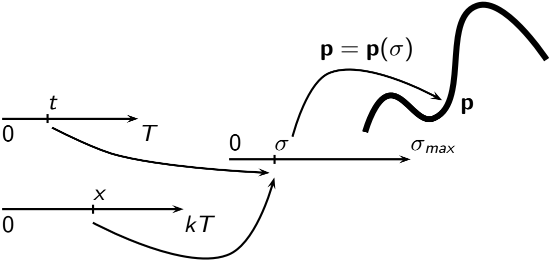
from which

$$\tau_i(x) - \gamma_i(x) = \frac{1}{k^2}[\tau_i(t) - \gamma_i(t)]$$

Dynamic scaling of trajectories

Some considerations:

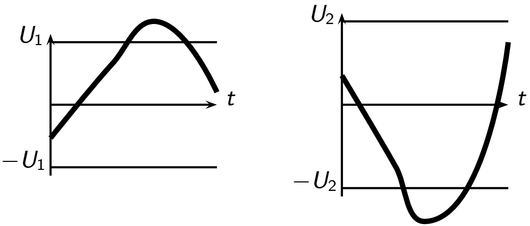
- it is not necessary to re-compute the whole trajectory
- neglecting the gravitational term, the 'new' torques are obtained by scaling by the factor $1/k^2$ the 'old' torques.
- the motion os slower if $k > 1$, and it is faster if $k < 1$ (total duration equal to kT)



$$\hat{\sigma}(x) = \hat{\sigma}(kt) = \sigma(t)$$

Dynamic scaling of trajectories

Example: Consider a 2 dof manipulator. In order to track a desired motion, the following torques should be generated:



By defining

$$k^2 = max\left\{1, \frac{|\tau_1|}{U_1}, \frac{|\tau_2|}{U_2}\right\} \geq 1$$

then:

- $x = kt$
- total time $= kT \geq T$

Then, the new torques are physically achievable, $(\tau(x) = \tau(t)/k^2)$, and at least one of them saturates in a point.

A *variable scaling* can be adopted to avoid slowing down the whole trajectory (saturation usually occurs in a single point).

For the optimal motion law (minimum time), at least one actuator saturates in each segment of the trajectory.

Trajectory Planning for Robot Manipulators

Analysis of Trajectories

Claudio Melchiorri

Dipartimento di Ingegneria dell'Energia Elettrica e dell'Informazione (DEI)

Università di Bologna

email: claudio.melchiorri@unibo.it

Introduction

Vibrations are undesired phenomena often present in automatic machines. They are basically due to the presence of structural elasticity in the mechanical system, and may be generated during the normal working cycle of the machine due to several reasons.

In particular, *vibrations may be produced if trajectories with a discontinuous acceleration profile are imposed to the actuation system.*

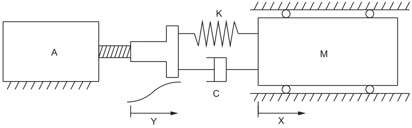
$\Longrightarrow$ Acceleration discontinuities $\rightarrow$ sudden variation of the inertial forces applied to the system.

$\Longrightarrow$ Relevant discontinuities of such forces, applied to an elastic system (i.e. any mechanical device), generate vibrational effects.

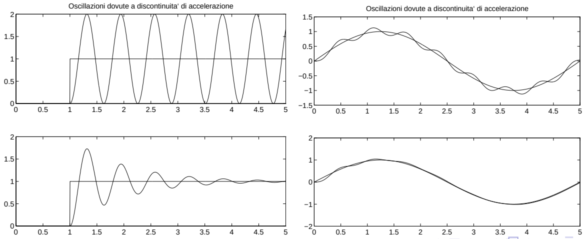
$\Longrightarrow$ Since every mechanism is characterized by some elasticity, this type of phenomenon must **always** be considered in the design of a trajectory, that therefore should have *a smooth acceleration profile or, more in general, a limited bandwidth*.

Example

Let consider a 1-dof mechanical system (output: position $x(t)$):



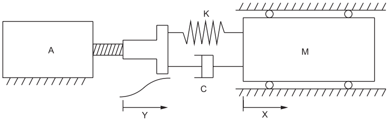
Acceleration $\ddot{x}(t)$ of the system when the (acceleration of the) input is a step or a sinusoidal function: without (top) and with damping (bottom).



Models for Analysis of Vibrations

Analysis of the vibration effects $\Longrightarrow$

Models that consider the elastic, inertial and dissipative properties of the elements of the mechanical system.



The complexity level of the model is usually chosen as a compromise between the desired precision and the computational burden.

The simplest criterion is to describe the mechanical devices, that are intrinsically distributed parameter systems, as lumped parameter systems, i.e. as rigid masses (without elasticity) and elastic elements (without mass).

Energy dissipative elements are introduced in order to consider frictional phenomena among moving parts.

The numerical values of the elements that describe inertia, elasticity and dissipative effects have to be determined by energetic considerations, i.e. trying to maintain the equivalence of the kinetic and elastic energy of the model with the energy of the corresponding parts of the mechanism under study.

The description of these phenomena can be either linear or nonlinear.

Linear model with one degree of freedom

Some considerations

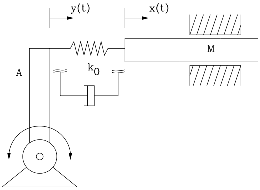
If $x(t)$ and $y(t)$ are the positions of mass M and A respectively, and $z(t) = y(t) - x(t)$, the dynamics of the system is described by:

$$m\ddot{x} + k_0(x - y) = 0$$

from which

$$\ddot{z} + \omega_0^2 z = \ddot{y}, \qquad \omega_0 = \sqrt{\frac{k_0}{m}}$$

being ω_0 the natural frequency of the mechanical system.



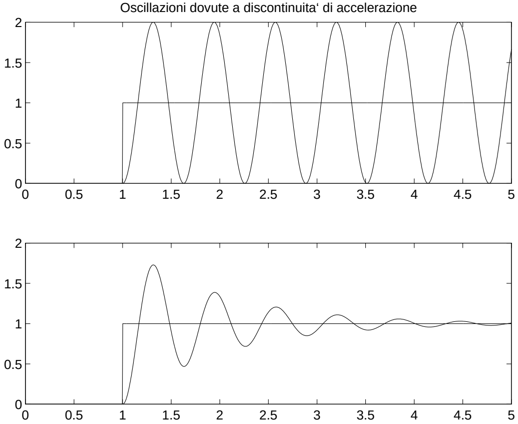
A model with viscous friction (coefficient b) on the mass M is described as

$$m\ddot{x} + b\dot{x} + k_0(x - y) = 0$$

Linear model with one degree of freedom

Output of the two models with a step input

Acceleration step
($b = 0$; $b \neq 0$)



Linear model with one degree of freedom

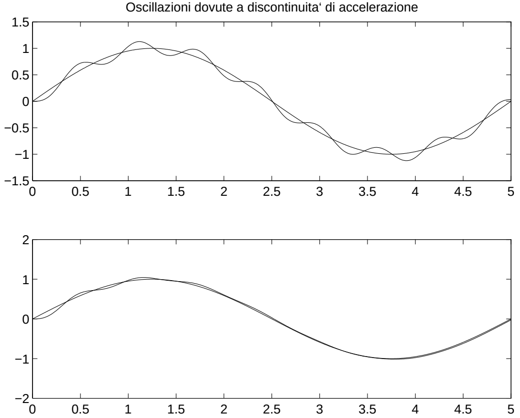
In the first case, due to oscillations the maximum value of $\ddot{x}$ is twice the value of $\ddot{y}$. Notice that this result does not depend on the stiffness k_0 of the mechanism:

- if k_0 increases, then the natural frequency ω_0 increases as well, while the amplitude of $\ddot{x}$ remains constant;
- the difference $z(t)$ between the positions of M and A depends on k_0 (if k_0 increases, then, $z(t)$ decreases).

Linear model with one degree of freedom

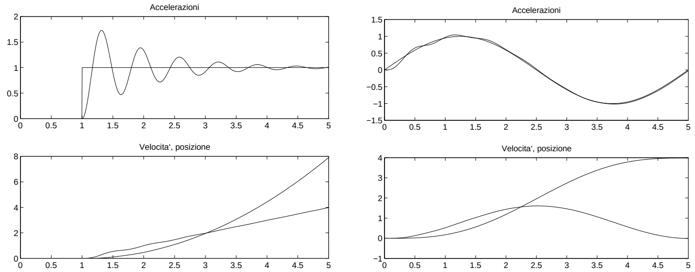
Output of the two models with a sinusoidal input:

Sinusoidal acceleration
$(b = 0; \quad b \neq 0)$



Linear model with one degree of freedom

Although the acceleration oscillations have not a real influence on the position of the mass M, they generate a structural stress to the mechanical device.



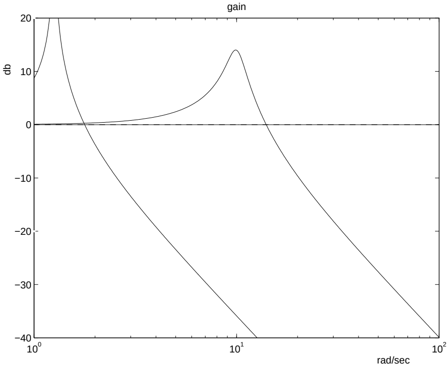
This phenomenon may be characterized by an analysis of the frequency content of the acceleration signal given as input to the system: the frequency range of the acceleration signal should be compared with the Bode diagram of the mechanism, and in particular with its natural frequency ω_0.

Linear model with one degree of freedom

Bode diagrams of the mechanical system ($b = 2, m = 1, k_0 = 100$) and of two acceleration signals:

$\rightarrow$ for the step, the Bode diagram is equal to $1 \; \forall \; \omega$,

$\rightarrow$ the frequency of the sinusoidal acceleration is $\omega = 2\pi T, \quad T = 5 \; s$.

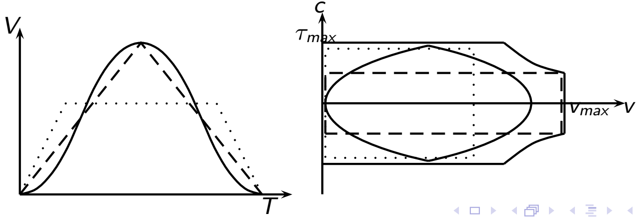


Comparison of the 'trapezoidal' and 'double S' trajectories

The 'trapezoidal' and 'double S' trajectories are very common in industrial practice, and therefore it is of interest a comparison of their main features. The criteria for the comparison are:

a) actuator usage;
b) duration of the trajectory;
c) analysis of the frequency content (vibrations induced on the mech. structure).

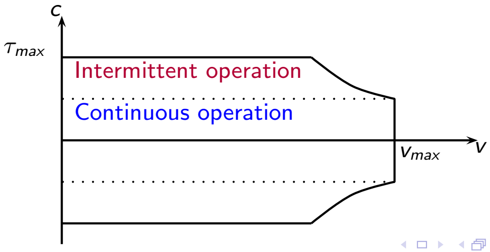
The trapezoidal, double S and triangular (limit case of trapezoidal) trajectories are considered for the analysis.
The duration of the trajectory is T, and the acceleration of the 'double S' and of the trapezoidal trajectories are $\sqrt{3}$ times the acceleration of the triangular profile, while the velocity of the trapezoidal trajectory has been set in order to obtain the same duration T.



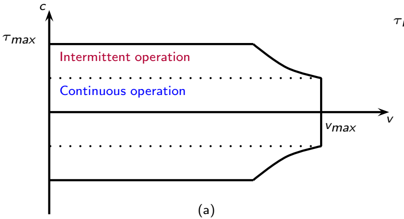
a - Use of actuators

From the data sheet, the following characteristics (among others) of an electric drive can be obtained:

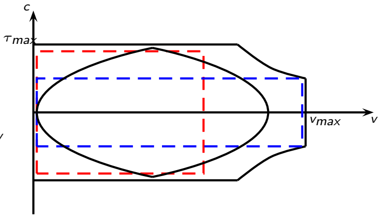
- *Continuous torque* (τ_c) (or *rated torque*): torque that the motor can produce continuously without exceeding thermal limits.
- *Peak torque* (τ_p): maximum torque that the motor can generate for short periods.
- *Rated speed* (ω_n): maximum value of the speed at rated torque (and at rated voltage).
- *Maximum power*: maximum amount of output power generated by the motor.



a - Use of actuators



(a)



(b)

- If the motor is in the continuous operation region, it may work for an indefinite period of time, while in the intermittent region it may work only for a limited amount of time.
- This limited period depends on the thermal dissipation properties of the motor and of the drive. On the other hand, different trajectories imply different utilizations of the motor, in particular with respect to the intermittent/continuous regions. As a matter of fact, the *double S trajectories allow to use the motor exploiting also the intermittent region*.
- Notice that with double S trajectories the maximum torque and the maximum velocities are obtained in different time instants.
- Theoretically, it is possible to enter the intermittent region also with the trapezoidal profile, but in this case the thermal power to be dissipated is higher than with double S trajectories.

a - Use of actuators

In comparing trajectories, there are some constitutive constraints for the motors that have to be considered:

1. the requested torque cannot in any case exceed the peak torque;
2. the RMS torque of the trajectory must be not higher than the continuous torque τ_{cont}.

The first is a mechanical constraint, since for all the motors there is a limit to the torque that can be generated.
The second is related to the thermal energy dissipation capability of the inverter. The trajectory must avoid heating of the motor.

If τ_{ss} and τ_{tr} are the maximum torque values of the double S and triangular trajectories, the RMS torque is computed as

$$\tau_{rms} = \sqrt{\frac{1}{T}\int_0^T C^2(t)\, dt}$$

a - Use of actuators

With the triangular profile, one gets

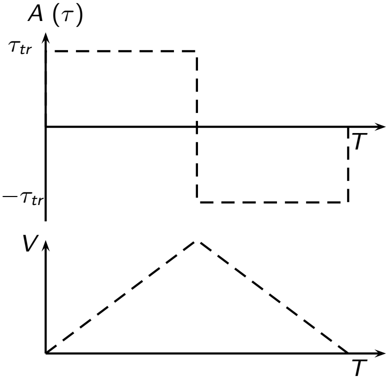
$$\tau^2(t) = \tau_{tr}^2, \qquad t \in [0,\, T]$$

then

$$\tau_{rms,tr} = \tau_{tr}$$

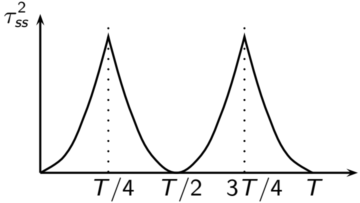
From the second constraint, one gets

$$\tau_{tr} = \tau_{cont}$$



a - Use of actuators

With the double S trajectory (consider the case without linear velocity segments), the torque is a linear function and then τ_{ss}^2 has a parabolic profile.



In this case

$$\tau_{rms,ss} = \frac{\tau_{ss}}{\sqrt{3}}$$

Therefore, the maximum torque achievable with this profile is

$$\tau_{ss} = \sqrt{3}\ \tau_{cont}$$

Notice that this trajectory allows a better exploitation of the motor with respect to the triangular profile: the maximum torque that it is possible to generate is higher $\tau_{ss} > \tau_{tr}$.

b - Duration of the trajectory

It is simple to show that the durations T_{tr} and T_{ss} of the triangular and double S (without linear velocity segments) trajectories are:

$$T_{tr} = 2\sqrt{\frac{L}{A_{max}}} \qquad\qquad T_{ss} = 2\sqrt{2}\sqrt{\frac{L}{A_{max}}}$$

Therefore

$$T_{ss} = \sqrt{2}\,T_{tr}$$

As expected, being smoother, the duration of the double S trajectory is 1.41 times higher than the duration of the triangular one.

b - Duration of the trajectory

On the other hand, with the double S trajectory it is possible to apply higher acceleration (torque) values (better usage of the actuator):

$$A_{max,ss} = \sqrt{3}\ A_{max,tr} = 1.7321\ A_{max,tr}$$

As a consequence, the duration T_{ss} of the double trajectory is reduced, and therefore

$$\frac{T_{ss}}{T_{tr}} = \sqrt{2}\sqrt{\frac{A_{max,tr}}{A_{max,ss}}} = \sqrt{2}\sqrt{\frac{1}{\sqrt{3}}} = 1.075$$

Notice that the condition $T_{ss} = T_{tr}$ is obtained with a torque $\tau_{ss} = 2\tau_{tr}$ ($A_{max,ss} = 2A_{max,tr}$).

c - Frequency analysis

In many motion control applications, when inertial loads have to be considered, the frequency range interested by the acceleration (torque) profile of the trajectory should be limited in order to avoid resonances or unmodeled dynamics of the mechanical structure.

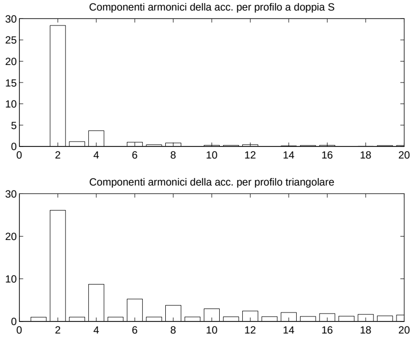
This aspect, that should always be taken into consideration, is of particular relevance in case of mechanisms with structural elasticities or high inertia. It is then important to evaluate the frequency content of the torque signals in order to understand their influence on the mechanical structure.

With this respect, it is obvious that the smoother the profiles, the better the results are (e.g. the double S trajectory is better that the triangular one because the frequency range is narrower).

c - Frequency analysis

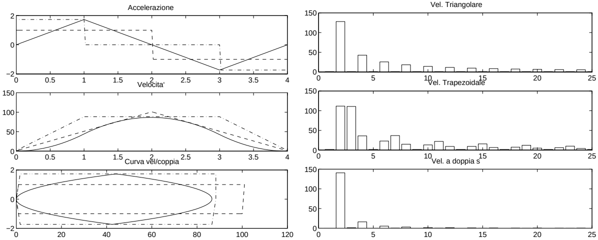
The figure, obtained with accelerations $A_{max,tr} : A_{max,ss} = 1 : \sqrt{3}$, it is possible to notice that the frequency content of the double S profile is lower (already in the second harmonic) than the triangular profile

The frequency range interested by the double S profile is narrower, and then its effect on resonances and unmodeled dynamics of the mechanical system (if present) is reduced.



c - Frequency analysis

Another example, considering a trapezoidal profile.



In case even double S trajectories are not smooth enough and oscillations are generated on the mechanical structure, smoother profiles should be adopted like, for example, trajectories with a trapezoidal jerk profile.

More in general, motion profiles with derivative up to a given order n should be considered: with a trapezoidal (triangular) velocity the trajectory is a C^1 function (continuous first derivative, while the second derivative is discontinuous), a double S trajectory is a C^2 function, etc.

An alternative approach is to use Spline functions with a proper order.

Coordination of more motion axes

In many applications, many motion axes are present and need to be coordinated or synchronized. It is therefore necessary to take proper actions for this purpose, ranging from a simple synchronization of the start/stop instants to more complex operations.

Example: automatic machine for packaging medicines (pills)

- *450 motion axes:*
  - 150 electric drives (DC, brushless),
  - 300 step motors
- grouped in *40 blocks* to be synchronized and coordinated
- *specific packages for the single client* (how many pills, what time, ...)

Coordination of more motion axes

Example: automatic machine for lifting TGV trains

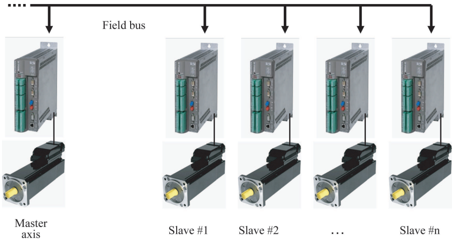
- Trains up to *200 meters* long
- Weight up to *386 tons*
- Accuracy between the two extremities: *1 mm*
- N. of lifting stations: 13

Coordination of more motion axes

In multi-axis machines based on mechanical cams, the synchronization of the different axes of motion is simply achieved by connecting the slaves to a single master (the coordination is performed at the mechanical level).

In case of electronic cams the problem must be considered in the design of the motion profiles for the different actuators (the synchronization is performed at the software level).

A common solution is to obtain the synchronization of the motors by defining a master motion, that can be either virtual (generated by software) or real (the position of an actuator of the machine), and then by using this master position as "time" (i.e. the variable $\theta(t)$) for the other axes.

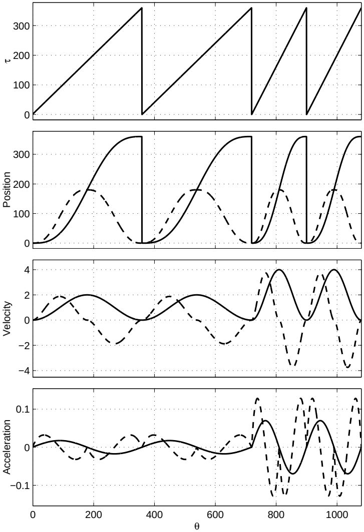


Coordination of more motion axes

An example is reported in the figure, where the variable τ is computed as a function of the angular position θ of the master. In the first two cycles ($\theta \in [0^0, 720^0]$) the motion is "slow" ($\tau = \theta$), while in the last one ($\theta \in [720^0, 1080^0]$) the motion is "fast" ($\tau = 2\,\theta$).

Two slave axes are present: the first one generates a cycloidal profile from $q_{c0} = 0^0$ to $q_{c1} = 360^0$ (solid), while the second one generates a polynomial profile of degree 5 (dashed) interpolating the points $q_{p0} = 0^0, q_{p1} = 180^0, q_{p2} = 0^0$, in both cases for $\tau = [0^0, 360^0]$.

Note that in the last cycle the velocity values are doubled, while the accelerations are four times those present in the the first cycles.



Coordination of more motion axes

In defining the (constant) velocity v_c of the master axis, i.e. the motion law $\theta(t)$, the most 'stressed' axis (in terms of velocity, acceleration, ...) should be taken into consideration in order to define profiles that can be generated by each motor:

$$v_c = \min\left\{ \frac{v_{max\,1}}{|\dot{q}_1(\theta)|_{max}}, \ldots, \frac{v_{max\,n}}{|\dot{q}_n(\theta)|_{max}}, \sqrt{\frac{a_{max\,1}}{|\ddot{q}_1(\theta)|_{max}}}, \ldots, \sqrt{\frac{a_{max\,n}}{|\ddot{q}_n(\theta)|_{max}}}, \sqrt[3]{\frac{j_{max\,1}}{|\dddot{q}_1(\theta)|_{max}}}, \ldots, \sqrt[3]{\frac{j_{max\,n}}{|\dddot{q}_n(\theta)|_{max}}} \right\}$$

Synchronization of different axis of motion can also be defined analytically, as already briefly discussed for trapezoidal velocity or spline trajectories.

Trajectory Planning for Robot Manipulators

Trajectories in the Workspace

Claudio Melchiorri

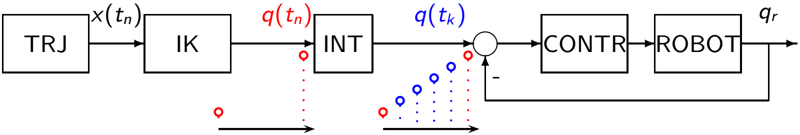
Dipartimento di Ingegneria dell'Energia Elettrica e dell'Informazione (DEI)

Università di Bologna

email: claudio.melchiorri@unibo.it

Workspace trajectories

If trajectories are defined in the workspace, it is necessary to use the inverse kinematic function to translate the motion specification to the joint space (where actuators operate). Since this increases the computational burden for trajectory planning, the operations of computing the trajectory and translating it to the joint space are made at a lower frequency with respect to the control frequency. Therefore, it is necessary to interpolate the data before assigning them to the low-level controllers: usually, a simple linear interpolation is adopted.



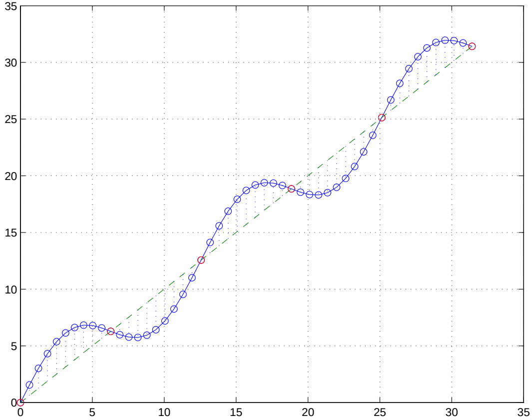
Typical values: $\Delta t_n = 10$ ms, $\Delta t_k = 1$ ms

$\Longrightarrow$ 10 values of $q(t_k)$ for each value of $q(t_n)$ $(x(t_n))$

$\Longrightarrow$ there is a delay Δt_n between the two sequences $q(t_k)$ and $q(t_n)$

Workspace trajectories

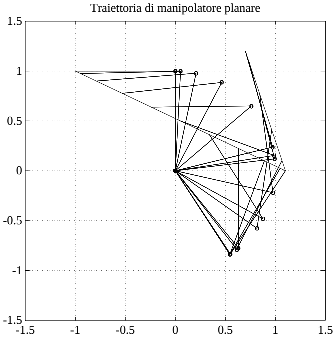
Another problem: the Cartesian positions actually achieved during the motion obtained by interpolating the points $q(t_n)$ are not those originally planned.



Workspace trajectories

For the computation of the workspace trajectories, it is possible to adopt one of the techniques used for the joint space (substituting the joint variable $q(t)$ with $x(t)$, i.e. a position or an orientation in the Cartesian space) or to define analytically the geometric path (e.g. an ellipse) as a function of time (i.e. $p = p(t)$) or, better, in a parametric form $\mathbf{p} = \mathbf{p}(s)$, being $s = s(t)$ a proper parameterization defining the motion law.

Example: planar 2 dof manipulator



Desired trajectory:

- total duration $3s$,
- start in $p_i = [-1.0,\ 1.0]$
- end in $p_f = [0.7,\ 1.2]$
- composed by two linear segments with intermediate point $p_m = [1.1,\ 0.0]$ per $t_m = 2s$.

Example: planar 2 dof manipulator

Consider the parametric form

$$
\begin{aligned}
x(t) &= x_0 + \frac{x_1 - x_0}{\Delta T}\,\tau \\
y(t) &= y_0 + \frac{y_1 - y_0}{\Delta T}\,\tau
\end{aligned}
$$

where $\Delta T = t_1 - t_0$, and $\tau \in [0, \Delta T]$ is defined so that desired position/velocity profiles are obtained, for example linear segments with parabolic blends (position in the workspace). The kinematic model is

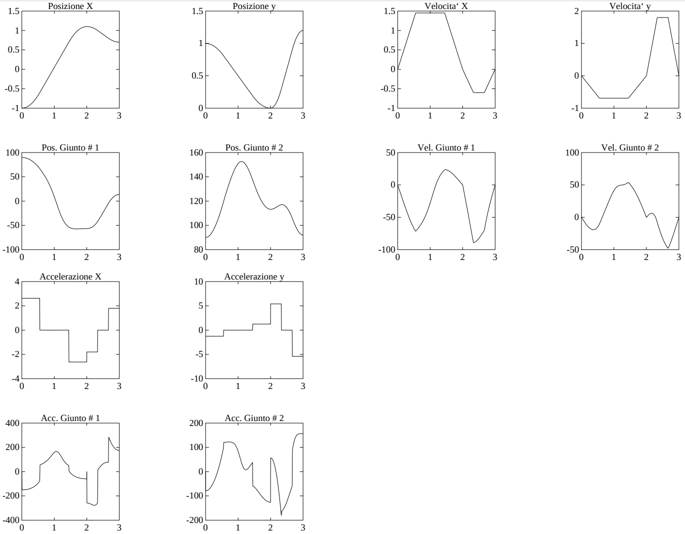
$$
x = l_1 C_1 + l_2 C_{12} \qquad\qquad y = l_1 S_1 + l_2 S_{12}
$$

while the inverse kinematic equations are

$$
C_2 = \frac{x^2 + y^2 - a_2^2 - a_3^2}{2 a_2 a_3} \qquad\qquad S_2 = \sqrt{1 - C_2^2}
$$

$$
\theta_2 = \mathrm{atan2}\,(S_2,\ C_2) \qquad\qquad \theta_1 = \mathrm{atan2}\,(y, x) - \mathrm{atan2}\,(a_2 S_2,\ a_1 + a_2 C_2)
$$

Example: planar 2 dof manipulator



Position trajectories

To plan a trajectory in the workspace, usually the geometric path p (line, circle, ellipse, ...) is defined as a function of a parameter $s(t)$: $p = p(s)$.

The parameter $s = s(t)$ is computed by using one of the techniques discussed for joint space trajectories. A classical approach is to plan $s(t)$ as a linear function with parabolic blends, in order to have in the work space acceleration/deceleration tracts (low stress for the mechanical and actuation system).

Notice that for parameterized trajectories the following conditions hold:

$$\dot{\mathbf{p}} = \frac{d\ \mathbf{p}}{ds}\dot{s}, \qquad \ddot{\mathbf{p}} = \frac{d\ \mathbf{p}}{ds}\ddot{s} + \frac{d^2\ \mathbf{p}}{ds^2}\dot{s}^2$$

Position trajectories

Curvature of a geometric path

Consider a path Γ in the workspace $\mathbb{R}^3$, expressed in parametric form

$$p = p(r) = \begin{bmatrix} x(r) \\ y(r) \\ z(r) \end{bmatrix}, \qquad r \in [r_a, \ r_b]$$

Assume that the curve is *regular*, i.e.

$$\dot{p} = \frac{d\ p}{dr} \neq 0, \qquad \forall r \in [r_a, \ r_b]$$

Given a point p_a of Γ, and a motion direction on the path, the *arc lenght* of a generic point $p(r)$ is defined as

$$s = \int_{p_a}^{p(r)} ||\dot{p}(\rho)|| d\rho$$

Position trajectories

By definition, the arc length represents the length of the arc of Γ defined by the two points p and p_a (if p follows p_a, or the opposite of such a length if p is before p_a). The value $s = 0$ is assigned to point p_a.
A bijective relationship exists between the values of the arc length s and the points of the path Γ, and then it is possible to use the arc length for a parametric expression of Γ.

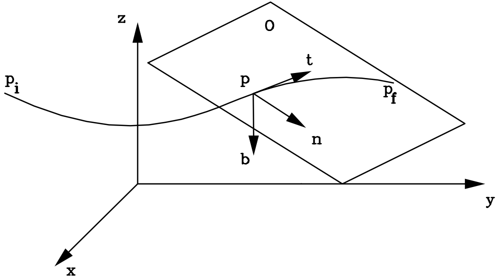
$$p = p(s)$$

It is possible to assign to each point p of Γ a reference frame (Frenet frame) defined by the following unit vectors

$$\begin{cases} \mathbf{t} = \dfrac{\dot{\mathbf{p}}}{||\dot{\mathbf{p}}||} & \text{tangent unit vector} \\ \mathbf{b} = \dfrac{\dot{\mathbf{p}} \times \ddot{\mathbf{p}}}{||\dot{\mathbf{p}} \times \ddot{\mathbf{p}}||} & \text{binormal unit vector} \\ \mathbf{n} = \mathbf{b} \times \mathbf{t} & \text{normal unit vector} \end{cases}$$

Position trajectories

- The unit vector $\mathbf{t}$ lies along the direction tangent to Γ in $\mathbf{p}$, and is directed along the positive s direction
- The unit vector $\mathbf{n}$ defines, with $\mathbf{t}$, the *osculating plane O*, defined as the plane containing point $\mathbf{p}$ and a point $\mathbf{p}' \in \Gamma$ when $\mathbf{p}' \rightarrow \mathbf{p}$.
- The unit vector $\mathbf{b}$ (*binormal*) is defined so that the frame $(\mathbf{t}, \mathbf{n}, \mathbf{b})$ is right-handed. Notice that it is not always possible to define uniquely the Frenet frame.



Position trajectories

Segment of a line

The linear geometric path between points $\mathbf{p}_i$ and $\mathbf{p}_f$ has a parametric representation expressed by

$$\mathbf{p}(s) = \mathbf{p}_i + \frac{s}{||\mathbf{p}_f - \mathbf{p}_i||}(\mathbf{p}_f - \mathbf{p}_i), \qquad s \in [0, \ \ ||\mathbf{p}_f - \mathbf{p}_i||]$$

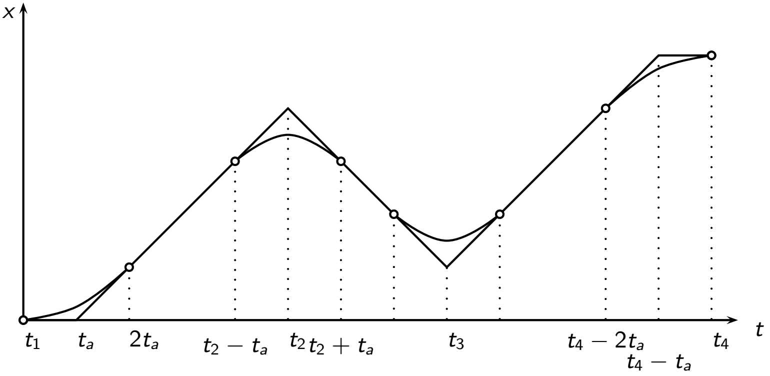
Moreover, by deriving $\mathbf{p}$ with respect to s, one obtains

$$\frac{d\ \mathbf{p}}{ds} = \frac{\mathbf{p}_f - \mathbf{p}_i}{||\mathbf{p}_f - \mathbf{p}_i||}, \qquad \frac{d^2\ \mathbf{p}}{ds^2} = 0$$

It is possible to plan a trajectory through a sequence of points with the same modalities seen in the joint space. If it is required to pass exactly through the intermediate points, then it is possible to compute the parameter s using one of the motion laws defined in the joint space (e.g. cubic, trapezoidal, . . .).
In case it is not required for the manipulator to pass through the intermediate points, the geometric path can be defined for example by linear segments with polynomial blends (position error, but non null velocity in the via points).

Position trajectories

A typical profile is shown below. The variable x is defined with a sequence of points interpolated with linear segments, while the real trajectory only approximates (in the vicinity of the via points) the given path.



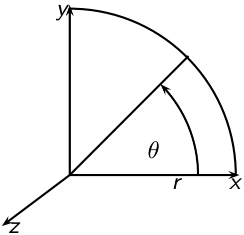
Position trajectories

Arc of a circle

A parametric representation of an arc of a circle is

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r\cos(\theta) \\ r\sin(\theta) \\ 0 \end{bmatrix} \qquad \theta \in [\theta_{min}, \theta_{max}]$$

where the parameter is the angle $\theta = \theta(t)$. Notice that if the path must be arbitrarily positioned/oriented in the 3D space, it is sufficient to multiply the (homogeneous) vector $\mathbf{p}$ by a proper transformation matrix T.

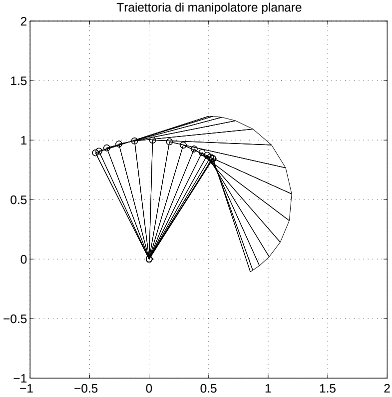


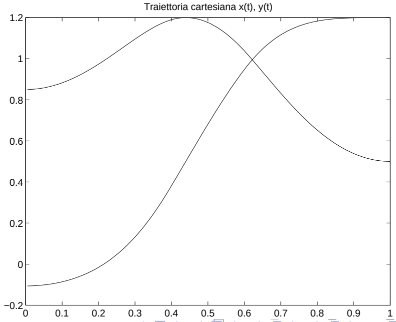
A motion law with acceleration/deceleration tracts (in the operational space) is obtained if the parameter (in this case: θ) is computed, for example, with a trapezoidal velocity profile.

Position trajectories - Example

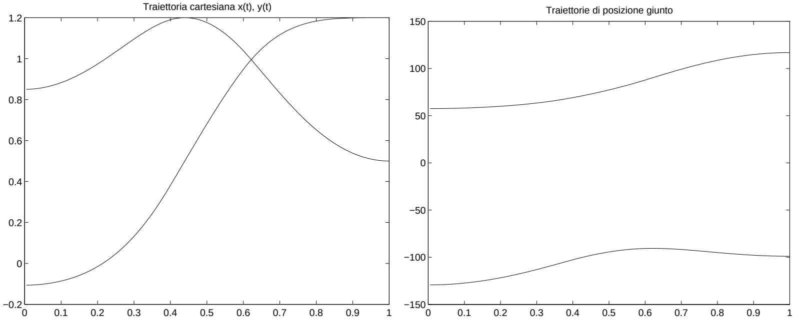
Planar 2 dof manipulator with links' lenght: $a_1 = a_2 = 1$. The desired circular motion is defined by

$$\text{center} = [0.5,\ 0.5], \qquad \text{radius r} = 0.7$$
$$\theta_i = -60, \quad \theta_f = 90, \quad T = 1 \text{ s}$$





Position trajectories



Rotational trajectories

Planning trajectories in terms of changes in orientation is somehow more complex than planning in position only. While it is quite simple to plan a motion between points $\mathbf{p}_i$ and $\mathbf{p}_f$, the same is not true for interpolating the orientation between two rotational matrices $\mathbf{R}_i$ and $\mathbf{R}_f$: for example if the elements r_{ij} are changed linearly from the initial (in $\mathbf{R}_i$) to the final (in $\mathbf{R}_f$) value, there is not guarantee that the intermediate matrices are real rotation matrices (orthogonal columns with unit norm).

Usually, the Euler or RPY angles are employed or, alternatively, the angle/axis representation.

With the Euler or RPY angles, two triples ϕ_i, ϕ_f are defined, and an interpolation based on one of the presented techniques can be adopted (advisable in any case continuity at least of the in rotational velocity).

Rotational trajectories

With the angle/axis representation, if $\mathbf{R}_i$ and $\mathbf{R}_f$ are the initial and final rotation matrices, then a matrix $\mathbf{R}_{i,f}$ exists such that

$$\mathbf{R}_i \; \mathbf{R}_{i,f} = \mathbf{R}_f$$

or

$$\mathbf{R}_{i,f} = \mathbf{R}_i^T \mathbf{R}_f = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

Then, the unit vector $\mathbf{w}$ and the rotational angle θ are

$$\theta_r = \mathrm{acos}\,\frac{r_{11} + r_{22} + r_{33} - 1}{2} \qquad (10)$$

$$\mathbf{w} = \frac{1}{2\sin\theta_r} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix} \qquad (11)$$

Rotational trajectories

It is now necessary to define a matrix $\mathbf{R}_t(t)$ so that $\mathbf{R}_t(0) = \mathbf{I}$ and $\mathbf{R}_t(t_f) = \mathbf{R}_{i,f}$. A choice can be

$$\mathbf{R} = \begin{bmatrix} w_x^2(1-C_\theta)+C_\theta & w_x w_y(1-C_\theta)-w_z S_\theta & w_x w_z(1-C_\theta)+w_y S_\theta \\ w_x w_y(1-C_\theta)+w_z S_\theta & w_y^2(1-C_\theta)+C_\theta & w_y w_z(1-C_\theta)-w_x S_\theta \\ w_x w_z(1-C_\theta)-w_y S_\theta & w_y w_z(1-C_\theta)+w_x S_\theta & w_z^2(1-C_\theta)+C_\theta \end{bmatrix}$$

where $\theta(t)$ is computed according to one of the previous motion law (cubic, trapezoidal, ...) from $\theta(0) = 0$ to $\theta(t_f) = \theta_r$, while $\mathbf{w}$ is defined as in (11).

The following rotation matrix is then obtained

$$\mathbf{R}(t) = \mathbf{R}_i \mathbf{R}_t(\theta(t))$$

Workspace trajectories

Final considerations

Some techniques for planning trajectories in the joint and in the work space have been illustrated.

If the trajectory is planned in the work space, the end-effector moves along well defined paths, a very important aspect in many industrial applications.

On the other hand, the computational burden is higher in case of work-space trajectories. For this reason, the frequency at which the trajectory in computed in lower than the control frequency, and an interpolation is then necessary.

Moreover, since the velocity/acceleration/torque limits required in the work-space may result non physically achievable in the joint space (i.e. in the actuation space) a re-computation of the trajectory might be necessary.

Workspace trajectories

Final considerations

Finally, singular configurations may generate problems if the trajectory is planned in the work space.

As a matter of fact, if a motion defined in the work space reaches points close to singular configuration, it should be avoided. Therefore, the trajectory should be checked in advance and, in case, not actuated or modified.

Clearly all these problems are not present if the trajectory is planned in the joint space.