Control of robot manipulators

Claudio Melchiorri

Dipartimento di Elettronica, Informatica e Sistemistica (DEIS)
Università di Bologna

email: claudio.melchiorri@unibo.it
Robot Position Control

Introduction

Decentralized position control
  - Cascade Control
  - Position feedback
  - Position and velocity feedback
  - Position, velocity and acceleration feedback
  - Feed-forward control

Centralized position control
  - PD controller with gravity compensation
  - Inverse dynamics control
Robot control

**Control problem:** definition of the input signals for the joints (e.g. torques or actuator input voltages) in order to achieve a predefined behavior for the manipulator.

The achievable performances can be very different because of:

- The many control techniques available to solve such a problem
- The hardware used to implement the control algorithms
- The mechanical configuration of the robot (anthropomorphic, cartesian, ...)

The robot performances are mainly influenced by the mechanical design and by the actuation system. For example:

- The cartesian configuration decouples the dynamics of the joints;
- DC motor with gearboxes have a linear dynamics that results “decoupled” from the non-linear dynamics of the robot. However, gearboxes usually introduce non-linear effects such as dead-zones, friction, elasticity, ...;
- Direct Drive motors on one hand ensure better performances and do not introduce non-linearities in the transmission chain; on the other hand a more relevant dynamic coupling between joints is present, and the (nonlinear) load dynamics is directly applied to the actuators without reduction effects.
Robot control

Control problems:
- Control of the robot’s motion (*position control* schemes);
  - joint-space control
  - workspace control.
- Control of the interaction with the workspace (*force control* schemes).

Control schemes:
- Decentralized (or independent) control schemes (SISO)
- Centralized control schemes (MIMO).

Dynamic model of a manipulator:

\[
M(q)\ddot{q} + C(q, \dot{q})\dot{q} + D\dot{q} + g(q) = \tau + J^T(q)F_a
\]

Control problem: define the generalized forces \(\tau\) to be applied to the joints in order to obtain a desired trajectory \(q_d(t)\).
Joint space control

Consideration #1

Actuators: positions $q_m(t)$, torques $\tau_m(t)$;

Gearboxes: reduction ratio $K_r$;

Joints: positions $q(t)$, generalized forces $\tau(t)$;

\[
K_r \ q = q_m \\
\tau_m = K_r^{-1} \tau
\]

$K_r$ is a diagonal matrix with elements $\gg 1$. 
Joint space control

Consideration #2

The diagonal of the matrix \( \mathbf{M}(\mathbf{q}) \) is composed by two kinds of elements:
- Inertia terms that do not depend on the robot’s configuration
- Terms that depend on the robot’s configuration.

Therefore:

\[
\mathbf{M}(\mathbf{q}) = \bar{\mathbf{M}} + \Delta \mathbf{M}(\mathbf{q})
\]

where \( \bar{\mathbf{M}} \) is a diagonal matrix with constant elements (i.e. the mean values of the joints inertia). From the robot dynamic model it follows:

\[
\tau_m = (K_r^{-1}\bar{\mathbf{M}}K_r^{-1})\ddot{\mathbf{q}}_m + D_m\dot{\mathbf{q}}_m + \mathbf{d}
\]

where

\[
D_m = K_r^{-1}DK_r^{-1}
\]

is the matrix collecting the motors friction coefficients, and

\[
\mathbf{d} = (K_r^{-1}\Delta \mathbf{M}(\mathbf{q})K_r^{-1})\ddot{\mathbf{q}}_m + (K_r^{-1}\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})K_r^{-1})\dot{\mathbf{q}}_m + K_r^{-1}\mathbf{g}(\mathbf{q})
\]

is a term that can be considered as a disturbance.
Joint space control

\[ \dot{q}_m = K_r \hat{M}^{-1} K_r (\tau_m - D_m \dot{q}_m - d) \implies \]

A manipulator (+ the actuation system) can be regarded as the composition of:

- a system with input \( \ddot{q}_m, \dot{q}_m, q_m \) and output \( d \), non-linear and with couplings
- a system with input \( \tau_m \) and output \( q_m \), linear and decoupled
Decentralized control

Each joint is considered independently, and the term $d$ is considered as an external disturbance. These considerations can be applied with proficiency when there is no direct coupling between the actuator and the joint (i.e. $K_r \gg I$).

**Joint independent control**

Let us consider a DC motor.

- **Electric dynamics (armature)**

  \[ v_A(t) = R_A i_A(t) + L_A \frac{d}{dt} i_A(t) + v_M(t) \]

  where $v_A$, $i_A$, $R_A$ and $L_A$ are the voltage, the electric current, and the armature resistance and inductance respectively.

- **Electro-mechanical coupling**

  \[ \tau_m(t) = k_t \ i_A(t) \]

  where $k_t$ is the electro-mechanical constant of the motor.
Decentralized control

- **Mechanical dynamics (rotor)**

\[ l \frac{d}{dt} \omega_m(t) = \tau_m(t) - b \omega_m(t) \]

where \( l \) is the moment of inertia of the motor and of an eventually connected payload and \( b \) the coefficient of friction of the overall system.

- **Mechano-electrical coupling**

\[ v_M(t) = k_v \omega_m(t) \]

Numerically, it results \( k_v = k_t \) if all the quantities are expressed in the International System.
Decentralized control

Using Laplace transformation, it results:

\[ V_A(s) = R_A I_A(s) + sL_A I_A(s) + V_M(s) \]

\[ C_m(s) = k_t I_A(s) \]

\[ s I \Omega_m(s) = C_m(s) - b \Omega_m(s) \]

\[ V_M(s) = k_v \Omega_m(s) \]

The two dynamics are then expressed by the transfer functions:

\[ I_A(s) = \frac{V_A(s) - V_M(s)}{R_A + s L_A} \]

\[ \frac{\Omega_m(s)}{C(s)} = \frac{1}{b + s l} \]
Decentralized control

\[ V_A(s) \rightarrow \frac{1}{R_A + sL_A} \rightarrow I_A(s) \rightarrow k_t \rightarrow C_m(s) \rightarrow \frac{1}{b + s I} \rightarrow \Omega_m(s) \rightarrow \frac{1}{s} \rightarrow \Theta_m(s) \]

\[ V_M(s) \rightarrow k_v \rightarrow \]

Considering both the transfer functions:

\[ \Omega_m(s) = \frac{k_t}{(b + s I)(R_A + s L_A) + k_t k_v} V_A(s) = G(s) V_A(s) \]

If \( b \approx 0 \):

\[ G(s) = \frac{k_t}{s^2 L_A I + s R_A I + k_t k_v} \]
Decentralized control

The system has two poles:

\[ p_{1,2} = \frac{-R_A I \pm \sqrt{(R_A I)^2 - 4 L_A I k_t k_v}}{2 L_A I} \]

If \( L_A \) is sufficiently small and such that \((R_A I)^2 - 4 L_A I k_t k_v > 0\)
then the two poles are real and negative. Moreover, if

\[ L_A \ll \frac{R_A^2 I}{k_t k_v} \]

and \( \sqrt{1-x} \approx 1 - x/2 \) (for small values of \( x \)), it results

\[ p_1 \approx -\frac{k_t k_v}{R_A I} \quad p_2 \approx -\frac{R_A}{L_A} \]

where \( p_1 \) is the \textit{electromechanical pole} and \( p_2 \) the \textit{electrical pole}. Then

\[ G(s) = \frac{k_t/(L_A I)}{(s + \frac{k_t k_v}{R_A L})(s + \frac{R_A}{L_A})} = \frac{1/k_t}{(1 + sT_m)(1 + sT_e)} \]

\[ T_m = \frac{R_A I}{k_t k_v} \quad \text{mechanical time constant} \]

\[ T_e = \frac{L_A}{R_A} \quad \text{electrical time constant} \]

\textbf{Usually} \( T_m \gg T_e \)
Decentralized control

Since usually $T_m \gg T_e$ ($T_e$ is negligible if compared with $T_m$), then the following simpler scheme can be considered:

\[
\begin{align*}
V_c & \rightarrow G_v \rightarrow \frac{k_t}{R_A} \rightarrow \frac{1}{l} \rightarrow \frac{1}{s} \rightarrow \frac{1}{s} \\
\end{align*}
\]

where $G_v$ represents the voltage gain of the motor drive unit. In this case, the input-output transfer function is

\[
G(s) = \frac{\Theta(s)}{V_A(s)} = \frac{k_m}{s(1 + sT_m)}
\]

where

\[
\begin{align*}
 k_m &= 1/k_v \quad \text{velocity-voltage gain} \\
 T_m &= \frac{R_A l}{k_t k_v} \quad \text{time constant of the motor}
\end{align*}
\]
Decentralized control with payload

If a gearbox is present between the motor and the load, then:

\[ \tau_m(t) \omega_m(t) = \tau(t) \omega(t) \quad (\omega < \omega_m) \]

and, since \( K_r \omega = \omega_m \) then \( \tau(t) = K_r \tau_m(t) \quad (\tau > \tau_m) \).

Thus, considering the variables before the gearbox (motor side) and using the Laplace transforms, one gets

\[ C_{r,m}(s) = \frac{C_r}{K_r} = \frac{1}{K_r} \left( b_c \Omega(s) + s l_c \Omega(s) \right) = \frac{1}{K_r^2} \left( b_c + s l_c \right) \Omega_m(s) \]
Decentralized control - Presence of a payload

By considering both the motor and the load, one obtains:

\[ C(s) = C_m(s) + C_{r,m}(s) \]
\[ = (b \Omega_m(s) + s l \Omega_m(s)) + \frac{1}{K_r^2} (b_c + s l_c) \Omega_m(s) \]
\[ = (b + \frac{b_c}{K_r^2}) \Omega_m(s) + s (l + \frac{l_c}{K_r^2}) \Omega_m(s) \]
\[ = b_T \Omega_m(s) + s l_T \Omega_m(s) \]

which leads to:

\[ \Omega_m(s) = \frac{C(s)}{b_T + s l_T} \]

\[ b_T = b + \frac{b_c}{K_r^2} \quad \quad l_T = l + \frac{l_c}{K_r^2} \]
A cascade control scheme can be profitably used if:

⇒ the dynamics of the process to be controlled can be schematized with two (or more) distinct dynamics $G_1(s)$ and $G_2(s)$

⇒ $G_1(s)$ is “faster” than $G_2(s)$

⇒ it is possible to measure the input variable $v$ of $G_2(s)$

In these cases, better performances can often be obtained by using two or more control loops in a cascade configuration.
Cascade control

With a cascade control scheme, the following positive features can be obtained:

1. ‘local’ compensation of disturbances acting in the inner control loops, e.g. the disturbance $d$ can be compensated ‘before’ it affects the output $y$;
2. the dynamics between $v_d$ and $v$ (with control) is ‘faster’ than the dynamics between $u$ and $v$ (without control);
3. the internal loop is more robust with respect to variations of the parameters;
4. it is possible to apply anti-saturation techniques to the variables $v_d$ and $v$;
5. it is possible to obtain a predefined dynamic behavior between $v_d$ and $y$ so that the design of $R_2(s)$ is easier. In particular, if $G_1(s)$ is ‘faster’ than $G_2(s)$, it is possible to define $R_1(s)$ so that, for the design of $R_2(s)$, it is possible to consider $v \approx v_d$ (the design process is easier).
Cascade control

With a single controller $R(s)$, the overall transfer function is:

$$G_{t0}(s) = \frac{R(s)G_1(s)G_2(s)}{1 + R(s)G_1(s)G_2(s)} = \frac{1}{1 + \frac{1}{R(s)G_1(s)G_2(s)}}$$

(1)

By using the cascade control scheme, the system transfer function is:

$$G_{t1}(s) = \frac{R_1(s)R_2(s)G_1(s)G_2(s)}{1 + R_1(s)G_1(s) + R_1(s)R_2(s)G_1(s)G_2(s)}$$

$$= \frac{1}{1 + \frac{1}{R_2(s)G_2(s)} \left(1 + \frac{1}{R_1(s)G_1(s)}\right)}$$

(2)
Cascade control

In a cascade control configuration, there are more degrees of freedom than in standard control schemes. Moreover, it is easier to affect the internal dynamics of the system.

To conclude, if it is possible to assume that $R_1(s)G_1(s) \gg 1$ in the frequency range interested by $G_2(s)$, then the overall transfer function mostly depends on the external control loop: $G_{t1}(s) \approx 1/(1 + 1/(R_2G_2))$.

More in general, in case of $n$ internal loops, the system control function is:

\[
G_t(s) = \frac{R_1(s)R_2(s) \cdots R_n(s)G_1(s)G_2(s) \cdots G_n(s)}{1 + R_1(s)G_1(s) + R_1(s)G_1(s)R_2(s)G_2(s) + \ldots + R_1(s)G_1(s) \cdots R_n(s)G_n(s)}
\]

\[
= \frac{1}{1 + \frac{1}{R_n(s)G_n(s)} \left(1 + \frac{1}{R_{n-1}(s)G_{n-1}(s)} \left(\ldots \left(1 + \frac{1}{R_1(s)G_1(s)}\right)\right)\right)}
\]
Feedback control

Reduction of the effects of the disturbance $d$:
- high gain;
- integral action to annihilate the steady state error (e.g. the effect of the gravitational term).

Possible solution: to use a PI controller

$$C(s) = K_c \frac{1 + sT_c}{s}$$
Feedback control

Possible control scheme

\[ \theta_r \xrightarrow{C_P(s)} C_V(s) \xrightarrow{C_A(s)} V_A \xrightarrow{k_t/R_A} \int \int \]

where

\[ d' = \frac{dR_A}{k_T} \]

and

- \( C_P(s) \) ➞ position control
- \( C_V(s) \) ➞ velocity control
- \( C_A(s) \) ➞ acceleration control

In the inner control loop, an integral action must be present.
Position feedback

\[ C_P(s) = K_P \frac{1 + sT_P}{s} \quad C_V(s) = 1 \quad C_A(s) = 1 \]

\[ k_{TV} = k_{TA} = 0 \]
Position feedback

Transfer functions of the direct path, of the feedback path and of the closed loop.

\[ C(s)G(s) = \frac{k_m K_P (1 + s T_P)}{s^2 (1 + s T_m)} \] (1)

\[ H(s) = k_{TP} \] (2)

\[ \Theta(s) \Theta_r(s) = \frac{1}{k_{TP}} \frac{1}{1 + \frac{s^2 (1 + s T_m)}{k_m K_P k_{TP} (1 + s T_P)}} = \frac{1}{k_{TP}} (1 + s T_P) \]

\[ (1 + \frac{2 \delta s}{\omega_n} + \frac{s^2}{\omega_n^2})(1 + s \tau) \] (3)

There are three poles: \( \delta, \omega_n \) are respectively the damping coefficient and the natural frequency of the pair of complex poles, and \(-1/\tau\) defines a real pole.

The value of \( \delta, \omega_n, \tau \) depends on the choices made for the control parameters:

- if \( T_P < T_m \) the system is unstable;
- if \( T_P > T_m \) then \( 1/\delta \omega_n > T_P > \tau \);
- if \( T_P \gg T_m \) then, in case of high closed-loop gain, \( \delta \omega_n > 1/\tau \approx 1/T_P \) and the zero in \(-1/T_P\) almost cancels out the real pole; anyway, it results \( \delta \omega_n \geq -1/2 \ T_m \).
Position feedback

1. Case $T_P < T_m$

Root locus and typical step response of the system for $k = k_m K_P k_T P T_P / T_m$
Position feedback

2. Case $T_P > T_m \implies \delta \omega_n < 1/T_P < 1/\tau$

Root locus and typical step response of the system for $k = k_m K_P k_T P P / T_m$

*Remark:* the root locus has an asymptote at $x = \sigma_a = -\frac{1}{2} \left[ \frac{1}{T_P} - \frac{1}{T_m} \right]$. 
Position feedback

3. Case $T_P \gg T_m \implies \delta \omega_n > 1/\tau \approx 1/T_P$ (for high values of $K$)

Root locus and typical step response of the system for $k = k_m K_P k_{TP} T_P / T_m$
Position feedback

The disturbance-output transfer function is:

\[
\frac{\Theta(s)}{D(s)} = -\frac{sR_A}{k_tK_pk_{TP}(1 + sT_P)} \cdot \frac{k_mK_pk_{TP}(1 + sT_P)}{1 + s^2(1 + sT_m)}
\]

Then, by choosing high values of \( K_P \), it is possible to reduce the effects of \( d(t) \) during the transient.

There are two complex conjugate poles in \((-\delta \omega_n \pm i\sqrt{1 - \delta^2} \omega_n)\), a real pole in \((-1/\tau)\) and a zero in the origin due to the use of the PI controller. For this reason, all the constant disturbances (e.g. the gravity terms) are compensated for when the robot reaches a predefined configuration.
Position feedback

For comparison purposes, let us compute the transfer function of the system without and with the control action:

\[
\frac{\Theta(s)}{D'(s)} = -\frac{k_m}{s(1 + sT_m)}
\]

\[
\frac{\Theta(s)}{D'(s)} = -\frac{sk_m}{s^2(1 + sT_m) + K_pk_mk_T(1 + sT_P)}
\]

The *disturbance reduction factor* $X_R$ due to the position feedback depends on $K_P$ and results:

\[X_R = K_Pk_P\]

On the other hand, to avoid oscillations in the closed loop system, it is not convenient to assign high values of $K_P$.
Position feedback

Output with an impulse disturbance (without and with control):

An estimation of the time necessary to compensate the disturbance effects on the output signal is given by the output recovery time

\[ T_R = \max\{ T_P, \frac{1}{\delta \omega_n} \} \]
Position and velocity feedback

\[ C_P(s) = K_P \quad C_V(s) = K_V \frac{1 + sT_V}{s} \quad C_A(s) = 1 \]

\[ k_{TA} = 0 \]
Position and velocity feedback

Transfer functions of the direct and feedback paths respectively:

\[
C(s)G(s) = \frac{k_m K_P K_V (1 + s T_V)}{s^2 (1 + s T_m)}
\]

\[
H(s) = k_{TP} (1 + s \frac{k_{TV}}{K_P k_{TP}})
\]

It might be convenient to choose the zero of the controller \((-1/T_V)\) in order to cancel the real pole in \(-1/T_m\) (i.e. by assigning \(T_V = T_M\)). In this case:

\[
C(s)G(s) = \frac{k_m K_P K_V}{s^2}
\]

and the overall transfer function is:

\[
\frac{\Theta(s)}{\Theta_r(s)} = \frac{C(s)G(s)}{1 + H(s)C(s)G(s)} = \frac{1}{k_{TP}} \frac{1}{1 + \frac{s k_{TV}}{K_P k_{TP}} + \frac{s^2}{k_m K_P k_{TP} K_V}} = \frac{1}{k_{TP}} \left(1 + \frac{2 \delta s}{\omega_n} + \frac{s^2}{\omega_n^2}\right)
\]
Position and velocity feedback

With proper choices for the control parameters, it is possible to obtain desired values for $\delta$ and $\omega_n$. If these are given, then the control gains are computed as:

$$K_V k_{TV} = \frac{2\delta \omega_n}{k_m}, \quad K_P k_{TP} K_V = \frac{\omega_n^2}{k_m}$$

Note that $k_{TV}, k_{TP}$ are known constants. The disturbance-output transfer function is:

$$\frac{\Theta(s)}{D(s)} = -\frac{s R_a}{k_t K_P k_{TP} K_V (1 + s T_m)} \frac{1}{1 + \frac{s k_{TV}}{K_P k_{TP} + \frac{s^2}{k_m K_P k_{TP} K_V}}}$$

This transfer function has a disturbance reduction factor given by:

$$X_R = K_P k_{TP} K_V$$

There are two complex poles with real part $-\delta \omega_n (= -k_m K_V k_{TV})$, a real pole in $s = -1/T_m$, and a zero in the origin due to the PI controller. The estimated time to compensate the disturbance effects on the output is:

$$T_R = \max\{ T_m, \frac{1}{\delta \omega_n} \}$$

better than in the previous case since $T_m \ll T_P$, and the real part of the poles is not constrained by $\delta \omega_n < 1/2 T_m$. 

$\text{C. Melchiorri (DEIS)}$
Position and velocity feedback

Root locus and typical step response of the system with $k = k_m K_V k_{TV}$

For the plot of the root locus, notice that the loop gain is

$$H(s) C(s) G(s) = k_m K_V k_{TV} \left( s + \frac{K_P k_{TP}}{k_{TV}} \right) \rightarrow \begin{cases} 
2 \text{ poles} & s = 0 \\
1 \text{ zero} & s = - \frac{K_P k_{TP}}{k_{TV}} 
\end{cases}$$
Position, velocity and acceleration feedback

\[ \theta_r \rightarrow K_P \rightarrow K_V \rightarrow K_A \frac{1+sT_A}{s} \rightarrow \frac{k_t}{R_A I} \rightarrow \int \rightarrow \int \rightarrow \dot{\theta}_m \rightarrow \dot{\theta}_m \rightarrow \theta_m \]

\[ C_P(s) = K_P \quad C_V(s) = K_V \quad C_A(s) = K_A \frac{1+sT_A}{s} \]

C. Melchiorri (DEIS)
Position, velocity and acceleration feedback

Given the transfer function between the voltage $V_A$ and the motor velocity $\dot{\Theta}_m$

$$G'(s) = \frac{k_m}{(1 + k_m K_A k_{TA})[1 + \frac{s T_m(1+k_m K_A k_{TA} T_A)}{1+k_m K_A k_{TA}}]}$$

the transfer functions of the direct and the feedback path and of the closed loop are, respectively:

$$C(s)G(s) = \frac{K_P K_V K_A (1 + s T_A)}{s^2} G'(s)$$

$$H(s) = k_{TP} \left(1 + \frac{s k_{TV}}{K_P k_{TP}}\right)$$

$$\frac{\Theta(s)}{\Theta_r(s)} = \frac{1}{k_{TP}} \frac{1}{1 + \frac{s k_{TV}}{K_P k_{TP}} + \frac{s^2(1 + k_m K_A k_{TA})}{k_m K_P k_{TP} K_V K_A}}$$
Position, velocity and acceleration feedback

Also in this case, it is advisable to cancel the pole in $-1/T_m$. This can be done by one of the two following (equivalent) choices:

1) $T_A = T_m$

2) $k_mK_Ak_T A \gg T_m \quad k_mK_Ak_T A \gg 1$
Position, velocity and acceleration feedback

The disturbance-output transfer function is:

\[
\frac{\Theta(s)}{D(s)} = - \frac{sR_a}{1 + \frac{sk_{TV}}{K_P k_{TP}} + \frac{s^2(1 + k_m K_A k_{TA})}{k_m K_P k_{TP} K_V K_A}} \frac{k_t K_P k_{TP} K_V K_A (1 + sT_A)}{sR_a}
\]

The noise reduction factor is:

\[X_R = K_P k_{TP} K_V K_A\]

The estimated time to compensate the disturbance effect on the output is:

\[T_R = \max\{T_A, \frac{1}{\delta \omega_n}\}\]

that can be better than the previous case because it is possible to choose \(T_A < T_m\).

With reference to a second order function \(W(s)\), if \(\delta, \omega_n, X_R\) are given, it results:

\[
\frac{2K_P k_{TP}}{k_{TV}} = \frac{\omega_n}{\delta}, \quad k_m K_A k_{TA} = \frac{k_m X_R}{\omega_n^2} - 1, \quad K_P k_{TP} K_V K_A = X_R
\]
Position, velocity and acceleration feedback

Root locus and typical step response of the system with
\[ k = k_m k_{TV} K_V K_A / (1 + k_m K_A k_{TA}). \]

These results are equivalent to those obtained with the position and velocity feedback control loop (second order system).
Remarks

1. The position and velocity signals can be easily measured, while in general the acceleration is not available. A state variable filter can be used to estimate the acceleration value.

\[
\frac{\Theta_f(s)}{\Theta(s)} = \frac{k_1 k_2}{s^2 + k_1 s + k_1 k_2}
\]

thus its dynamics is characterized by

\[
\omega_{nf} = \sqrt{k_1 k_2}
\]

\[
\delta_f = 0.5 \sqrt{k_1 / k_2}
\]

The filter transfer function is

The values have to be chosen so that the filter bandwidth is higher (at least one decade) than the motor bandwidth. On the other hand, this could generate problems related to the presence of high-frequency noise.

2. The controllers have been designed neglecting the actuation system nonlinearities, which could cause problems with high gains values.
Example

Given a position trajectory defined as a trapezoidal velocity profile, the velocity and acceleration profiles can be estimated by using a filter state variables ($k_1 = k_2 = 10, k_{TP} = k_{TV} = k_{TA} = 1$). The estimated values are:

Note the overshoot of the estimated acceleration (can be reduced with a proper choice of the filter’s parameters).
Example

Consider now the same example, but with a white noise (amplitude = 0.0005) added to the position signal.
Feed-forward control action

When the request of performance increases, and therefore “faster” trajectories are specified (i.e. with higher values for the velocity/acceleration signals), normally a performance degradation takes place in terms of tracking capabilities (the “disturbance” $d$ becomes more relevant).

A possible solution to overcome this problem is to include in the controller feed-forward actions, which can be computed on the basis of the values of the desired position, velocity and acceleration signals.
Feed-forward control action

The positive effects related to the use of a feedback controller are widely known from the basic control theory. In particular, the most well known advantages are:

- noise rejection
- larger bandwidth for the controlled system if compared to the original system
- robustness in case of parametric uncertainties.

On the other hand, beside feedback control scheme, it is known that it is possible to include in the control scheme one or more feed-forward control actions, i.e. control actions that are not based on the feedback paradigm.
Feed-forward control action

In principle, under ideal conditions, a feed-forward control scheme allows to track perfectly the reference signal. On the other hand, the feed-forward actions can be computed only if the process to be controlled and the noise signals are perfectly known, and therefore it is not realistic to use only this control principle for real plants where modeling errors and noise/disturbances are always present.

In any case, when high-performances are required, the use of feed-forward controllers can significantly increases the results achievable with the feedback controlled system.
Feed-forward control action

Control actions are usually included in the forward path of a control scheme for the following main reasons:

1. to properly filter the reference input, by modifying the transfer function between the desired input $y_d$ and the output $y$, in order to obtain some predefined static or dynamic properties for the system;
2. to improve tracking performance;
3. to compensate known (or at least measurable) disturbances acting on the system.

\[
C_f(s) \quad \rightarrow \quad C_a(s) \quad \rightarrow \quad R(s) \quad \rightarrow \quad G(s) \quad \rightarrow \quad y
\]

\[
y_d \quad \rightarrow \quad C_f(s) \quad \rightarrow \quad y_d - y
\]
Feed-forward control action

One of the main goal in the design of a trajectory tracking control law is to have, ideally, a null error, i.e. \( y_d \equiv y \). On the other hand, by using only feedback control actions (intrinsically based on the error \( e = y_d - y \)), in general it is not possible to ensure that this requirement is met.

A possible solution is based on control schemes including feed-forward control actions.

The overall general scheme includes the classical feedback controller \( R(s) \), and two feed-forward control actions \( C_f(s) \) and \( C_a(s) \).
Reference signal compensation

The relationship between the output signal and the reference signal is now:

\[
Y(s) = C_f(s) \frac{R(s)G(s)}{1 + R(s)G(s)} Y_d(s) + \frac{C_a(s)G(s)}{1 + R(s)G(s)} Y_d(s)
\]

\[
= \frac{C_f(s)R(s)G(s) + C_a(s)G(s)}{1 + R(s)G(s)} Y_d(s)
\]

By imposing the desired (ideal) condition \( Y(s) = Y_d(s) \), it follows:

\[
C_a(s) = \frac{1}{G(s)} + R(s) [1 - C_f(s)]
\]

or, if \( C_f(s) = 1 \),

\[
C_a(s) = G^{-1}(s)
\]

The last relationship highlights the fact that, once the process dynamics \( G(s) \) and the reference signal \( Y_d(s) \) are known, for a perfect tracking of the input signal (i.e. \( y \equiv y_d \)) the control input must be computed by “inverting” the system dynamics. Formally:

\[
U(s) = G^{-1}(s)Y_d(s)
\]
Reference signal compensation

It is not always possible to use feed-forward control actions. In particular, it is not possible when:

1. $G(s)$ has at least a zero with positive real part (non minimum phase systems)
2. $G(s)$ has significant time delay (such as $e^{-tds}$)
3. $G(s)$ is not a proper system (i.e. the degree of the numerator is not equal to the degree of the denominator)

In the first two cases a non-casual scheme should be used, while in the third one an approximation of $G(s)$ should be introduced, removing part of the dynamics to define a proper transfer function for the computation of $G^{-1}(s)$. 
Reference signal compensation

Let us consider an electrical actuator. If its electrical dynamics is neglected, it can be approximated as a double integrator, i.e. $G(s) = 1/s^2$.

With this approximation, the inverse dynamic model needed to implement a feed-forward action is $G^{-1}(s) = s^2$, that results in a non-feasible operations since it corresponds to a double time-derivation of the reference signal.

On the other hand, since the desired trajectories are usually specified in terms of position, velocity and acceleration signals, in this case this does not constitute a problem.

In order to implement the feed-forward control actions, it is not necessary to derive (twice) the reference signal, since the velocity and acceleration (jerk, ...) reference values are already available.

To conclude, notice that in case of perfect knowledge of the transfer function $G(s)$, and in case of physical realizability of $G^{-1}(s)$, the feedback loop (i.e. $R(s)$) does not substantially contribute to the tracking of the desired signal $y_d$, and it is used only to compensate for disturbances possibly acting on the system.
Reference signal compensation

As an example, consider

\[ G(s) = \frac{1}{s+10} \quad R(s) = \frac{100(s+10)}{s+100} \quad C_f(s) = 1 \]

The feed-forward action is not physically feasible. In fact:

\[ C_a(s) = \frac{1}{G(s)} = \frac{s+10}{1} \]

first order time-derivative!

By considering a partial compensation only (i.e. the static gain)

\[ C_a(s) = 10 \]

in steady state conditions (for constant input values) one obtains \( y = y_d \).
Reference signal compensation

(a) Reference (dashed) and output signals

(b) Reference (dashed) and output signals
Reference signal compensation

On the other hand, by considering

\[ C_a(s) = \frac{1}{G(s)} = \frac{s + 10}{1} \]

it follows that

\[ U(s) = C_a(s)Y_d(s) = 10Y_d(s) + sY_d(s) \]

Since the ‘s’ in the Laplace notation corresponds to a time derivative, if the reference velocity is known then the following control scheme can be applied, with \( k_{a,p} = 10, \ k_{a,v} = 1 \).
Reference signal compensation

Notice the perfect tracking of the reference signal.
Reference signal compensation

1. **Position feedback:**
   By approximating the relationship between the voltage $V_A$ and the acceleration $\ddot{\Theta}_m$ as
   \[
   \ddot{\Theta}_m = \frac{k_t}{R_A l} V_A = \frac{k_m}{T_m} V_A
   \]
   and by approximating the relationship between the voltage $V_A$ and the velocity $\dot{\Theta}_m$ (in steady state) as
   \[
   \dot{\Theta}_m = k_m V_A
   \]
   it is possible to express $V_A$ as:
   \[
   V_A = \frac{K_P(1 + sT_P)}{s} (k_{TP}\Theta_d - k_{TP}\Theta_m) + \frac{1}{k_m} \dot{\Theta}_d + \frac{T_m}{k_m} \ddot{\Theta}_d
   \]
   This is equivalent to assume an input signal equal to:
   \[
   \Theta'_r(s) = [k_{TP} + \frac{s^2(1 + sT_m)}{k_m K_P(1 + sT_P)}] \Theta_d(s)
   \]
   **Remark:** when not directly provided, the reference values of velocity and acceleration can be easily computed if the control trajectory is expressed in analytical form.
Decentralized feed-forward compensation

The following scheme is obtained (position feedback + decentralized feed-forward action):

\[
\begin{align*}
\ddot{\theta}_d & \rightarrow \frac{T_m}{k_m} \\
\dot{\theta}_d & \rightarrow \frac{1}{k_m} \\
\dot{\theta}_d & \rightarrow k_{TP} \\
\theta_d & \rightarrow \frac{K_p (1+sT_p)}{s} \\
\theta_d & \rightarrow M(s)
\end{align*}
\]
Decentralized feed-forward compensation

2. Position/Velocity feedback
Similarly to the previous case, in case of position and velocity feedback loops it is possible to compute

$$\Theta'_r(s) = [k_{TP} + \frac{sk_{TV}}{K_P} + \frac{s^2}{k_mK_PK_V}]\Theta_d(s)$$

Then, the overall control scheme is:
Decentralized feed-forward compensation

3. Position/Velocity/Acceleration feedback:
In this case, it is:

\[ \Theta'_r(s) = \left[ k_{TP} + \frac{sk_{TV}}{K_P} + \frac{s^2(1 + k_mK_Ak_{TA})}{k_mK_PK_VK_A} \right] \Theta_d(s) \]

The scheme for the position/velocity/acceleration feedback + decentralized feed-forward control action is:
Example

**Without feed-forward action**

**With feed-forward action**
Decentralized feed-forward compensation

On the basis of the previous schemes, it is possible to remark the fact that if the number of inner loops increases, then the required knowledge of the dynamics model is less important. As a matter of fact:

- Position loop: the two parameters $T_m, k_m$ are necessary;
- Position and velocity loops: only $k_m$;
- Position, velocity and acceleration loops: $k_m$ (with reduced ‘importance’ since it appears in the term $k_{TA} + \frac{1}{k_m K_A}$).

Perfect trajectory tracking $\implies$ Perfect knowledge of the model.

**Saturation problem:** can be more easily solved with control scheme with inner loops.
Decentralized feed-forward compensation

It is possible to obtain control structures equivalent to those introduced above but based the position feedback only, with standard controllers of the P, PI, PID, ... types.

- These control schemes are equivalent if the input/output transfer function and the disturbance reduction properties are considered;
- However, in this way, it is not possible to apply suitable techniques to avoid saturation problems of the “internal” variables (electrical currents, accelerations, velocities).

These architectures describe most of the control schemes currently adopted in industrial robots (based on PID, although with many configurations).

Once again, it is important to remark that on one hand it is desirable to have high gains in the inner control loops in order to minimize the effects of modeling errors but, on the other hand, that there are limits to these gains resulting from inaccuracies in the model, digital implementation of the control algorithms, non-modeled dynamics (e.g. elastic effects, friction, ...), noise.
Decentralized feed-forward compensation

Control scheme equivalent to the position feedback control (PI)

Control scheme equivalent to the position/velocity feedback control (PID)
Decentralized feed-forward compensation

Control scheme equivalent to the position/velocity/acceleration feedback control (PIDD²)

\[ \begin{align*}
\ddot{q}_d & = \frac{T_m}{k_m} \\
\dot{q}_d & = \frac{1}{k_m} \\
q_d & = k_{TP} \\
\end{align*} \]
Pre-calculated torque forward compensation

Let us consider the most general controller (i.e. the PIDD$^2$) with feed-forward actions on velocity and acceleration:

The output variable $u$ of the controller is:

$$u = a_2 \ddot{e} + a_1 \dot{e} + a_0 e + a_{-1} \int_0^t e(\zeta) d\zeta$$

where $e = \theta_d - \theta$, and the parameters $a_{-1}, a_0, a_1, a_2$ depend on the adopted design criteria.
Pre-calculated torque forward compensation

The contributions of the feed-forward actions and of the disturbance term $d$ can be computed as

$$\frac{1}{k_m} \dot{\theta}_d + \frac{T_m}{k_m} \ddot{\theta}_d - \frac{R_A}{k_t} d$$

where

$$\frac{T_m}{k_m} = \frac{IR_A}{k_t}$$

and

$$k_m = \frac{1}{k_v}$$

By calculating $\ddot{\theta}$, it follows:

$$a_2 \ddot{e} + a_1 \dot{e} + a_0 e + a_{-1} \int e(\zeta) d\zeta + \frac{1}{k_m} (\dot{\theta}_d - \dot{\theta}) + \frac{T_m}{k_m} (\ddot{\theta}_d - \ddot{\theta}) = \frac{R_A}{k_t} d$$

or, after simple calculations:

$$a'_2 \ddot{e} + a'_1 \dot{e} + a'_0 e + a'_{-1} \int e(\zeta) d\zeta = \frac{R_A}{k_t} d$$
Pre-calculated torque forward compensation

If $d(t) = 0$, this equation shows that the error goes asymptotically to zero for all the feasible trajectories (i.e. considering voltage, velocity, acceleration and workspace limits).

If $d(t) \neq 0$, the disturbance-output transfer function is:

$$\frac{E(s)}{D(s)} = W(s) = \frac{\frac{R_A}{k_t}s}{a_2s^3 + a_1's^2 + a_0's + a'_{-1}}$$

Therefore, the tracking error is ‘small’ if the disturbance frequency is ‘lower’ than the bandwidth where the amplitude of $W(s)$ is high (i.e. for ‘slow’ motions).
Pre-computed torque feed-forward compensation

Since the term $d(t)$ is not an exogenous disturbance, but it is known from the dynamic model, it is possible to improve further the control performances.

In fact, given the desired position, velocity and acceleration signals for the actuators, it is always possible to compute a feed-forward control action able to compensate for $d(t)$:

$$d_d = K_r^{-1} \Delta M(q_d) K_r^{-1} \ddot{q}_{md} + K_r^{-1} C(q_d, \dot{q}_d) K_r^{-1} \dot{q}_{md} + K_r^{-1} g(q_d)$$
Pre-computed torque feed-forward compensation

Some remarks:

- The residual disturbance \( \tilde{d} = d_d - d \) is null only in the ideal case of perfect tracking \( (q = q_d) \) and if the dynamic model is perfectly known. In any case, \( \tilde{d} \) is much smaller if compared to \( d \).

- Usually, the computation of the feed-forward action \( d_d \) is computationally expensive (a centralized computation is needed): its real-time implementation could require a processing time too long if compared to the sampling time \( T_s \);

- A possible solution could be to achieve a *partial compensation* only, by computing the most significative terms of \( d_d \) such as those related to the robot inertia (on the diagonal of the \( \mathbf{M} \) matrix) and those related to gravity. In fact, all the terms related to the velocity are relatively small for the velocities typically used for industrial robots (a few ripples/second). In this way, it is possible to compensate only for terms related to the robot configuration, and not to terms due to motion or to the interaction with the environment;

- Note that in case of repetitive trajectories, the compensation terms can be computed off-line.
Centralized control

Dynamic model of a manipulator:

\[ M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + D_v \dot{q} + g(q) = \tau \]

Actuators and gearboxes may be described by:

\[
\begin{align*}
K_r q &= q_m \\
v_A &= G_v v_c \\
v_A &= R_A I_A + K_v q_m \\
\tau_m &= K_t I_A = K_r^{-1} \tau
\end{align*}
\]

where \( K_r \) is the matrix with the reduction coefficients; \( G_v \) the gain of the motor drives; \( R_A \) the armature resistances; \( K_v, K_t \) the electro-mechanic constants of the motors.

Consider the manipulator as a MIMO system with \( n \) input and \( n \) output.
Centralized control

**Voltage control mode:**

\[
I_A = R_A^{-1}(v_A - K_v \dot{q}_m) \\
= R_A^{-1} G_v v_c - R_A^{-1} K_v K_r \dot{q}
\]

\[
\tau = K_r K_t I_A \\
= K_r K_t R_A^{-1} G_v v_c - K_r K_t R_A^{-1} K_v K_r \dot{q}
\]

Thus, the robot dynamic equation becomes:

\[
M(q)\ddot{q} + C(q, \dot{q})\dot{q} + D\dot{q} + g(q) = u
\]

where

\[
D = D_v + K_r K_t R_A^{-1} K_v K_r \\
u = K_r K_t R_A^{-1} G_v v_c
\]

\[
\implies \text{diagonal matrix with friction terms} \\
\implies \text{control input}
\]
Centralized control

Manipulator dynamic model:

\[ M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + D\dot{q} + g(q) = u \]

The torque \( \tau \) (real input generating the motion of the manipulator) is not equal to \( u \) (algebraically related to \( v_c \)) because of the counter-electromotive force proportional to the joints velocity \( \dot{q} \).
Centralized control

**Torque control mode:** With the voltage-control mode, it is not possible to directly provide to the joints the exact torques needed to move the robot (complex computation should be introduced to compensate for the velocity-dependent terms). It is preferable to use for the motor a current-control modality: the actuator acts as a torque generator.

\[ I_A = G_i \ v_c \]

It follows:

\[ M(q)\ddot{q} + C(q, \dot{q})\dot{q} + D\dot{q} + g(q) = u \]

where

\[ D = D_v \quad u = K_r K_t G_i v_c \]
Centralized control

After this preliminary comments, two important centralized control schemes are now introduced:

1. PD + gravity compensation
2. Inverse dynamics control
PD controller with gravity compensation

Given a desired reference configuration $q_d$, the goal is to define a controller ensuring the global asymptotic stability of the nonlinear dynamical system (i.e. the robot) described by:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + D\dot{q} + g(q) = u$$

For this purpose, let us define the error as

$$\tilde{q} = q_d - q$$

Consider a dynamic system with state $x$ expressed as

$$x = \begin{bmatrix} \tilde{q} \\ \dot{q} \end{bmatrix}$$

The *direct Lyapunov method* is exploited for the control law definition.
PD controller with gravity compensation

Let us consider the following candidate Lyapunov function

\[ V(q, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q} + \frac{1}{2} \ddot{q}^T K_P \ddot{q} > 0 \quad \forall \dot{q}, \ddot{q} \neq 0 \]

where \( K_P \) is a square \((n \times n)\) positive-definite matrix.

Function \( V(q, \dot{q}) \) is composed by two terms:

- \( \frac{1}{2} \dot{q}^T M(q) \dot{q} \)
  expressing the kinetic energy of the system;
- \( \frac{1}{2} \ddot{q}^T K_P \ddot{q} \)
  that can be interpreted as elastic energy stored by springs with stiffness \( K_P \);
  these springs are a ‘physical interpretation’ of the position control loops.
**PD controller with gravity compensation**

The reference configuration $\mathbf{q}_d$ is constant, then $\dot{\mathbf{q}} = -\mathbf{q}$ and the time derivative of $V$ is:

$$\dot{V} = \dot{\mathbf{q}}^T \mathbf{M} \ddot{\mathbf{q}} + \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M} \dot{\mathbf{q}} - \dot{\mathbf{q}}^T \mathbf{K}_P \ddot{\mathbf{q}}$$

Since the robot dynamics can be rewritten as $\mathbf{M} \ddot{\mathbf{q}} = \mathbf{u} - \mathbf{C} \dot{\mathbf{q}} - \mathbf{D} \dot{\mathbf{q}} - \mathbf{g}$, then

$$\dot{V} = \dot{\mathbf{q}}^T \mathbf{M} \ddot{\mathbf{q}} + \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M} \dot{\mathbf{q}} - \dot{\mathbf{q}}^T \mathbf{K}_P \ddot{\mathbf{q}}$$

$$= \dot{\mathbf{q}}^T (\mathbf{u} - \mathbf{C} \dot{\mathbf{q}} - \mathbf{D} \dot{\mathbf{q}} - \mathbf{g}) + \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M} \dot{\mathbf{q}} - \dot{\mathbf{q}}^T \mathbf{K}_P \ddot{\mathbf{q}}$$

$$= \frac{1}{2} \dot{\mathbf{q}}^T [\mathbf{M}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})] \dot{\mathbf{q}} - \dot{\mathbf{q}}^T \mathbf{D} \dot{\mathbf{q}} + \dot{\mathbf{q}}^T [\mathbf{u} - \mathbf{g}(\mathbf{q}) - \mathbf{K}_P \ddot{\mathbf{q}}]$$

In order to compute the control input $\mathbf{u}$, note that:

- $\dot{\mathbf{q}}^T [\mathbf{M}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})] \dot{\mathbf{q}} = 0$ due to the choice of $\mathbf{C}$
- $-\dot{\mathbf{q}}^T \mathbf{D} \dot{\mathbf{q}}$ is negative-definite

Thus, by setting

$$\mathbf{u} = \mathbf{g}(\mathbf{q}) + \mathbf{K}_P \ddot{\mathbf{q}}$$

it is possible to guarantee that $\dot{V}$ is negative-semidefinite. In fact:

$$\dot{V} = 0 \quad \dot{\mathbf{q}} = 0, \quad \forall \ddot{\mathbf{q}}$$
PD controller with gravity compensation

The same result can be achieved also by adding a second term to the control $u$:

$$u = g(q) + K_P \ddot{q} - K_D \dot{q}$$

By defining $K_D$ as a positive-definite matrix, it results

$$\dot{V} = -\dot{q}^T (D + K_D) \dot{q}$$

As a consequence, the convergence ratio of the system to the equilibrium is increased.

Note that the terms $K_D \dot{q}$ is equivalent to a derivative action in the control loop (PD and gravity compensations).
PD controller with gravity compensation

\[
\dot{q} + K_P \ddot{q} + K_D g(\cdot) = u - q_d
\]

**Remarks:**

- The control law is a linear PD controller with a nonlinear term (for gravity compensation). The system is globally asymptotically stable for any choice of \(K_P, K_D\) (positive-definite);

- The derivative action is fundamental in systems with low friction effects. Typical examples are manipulators equipped with Direct Drive motors: the low electrical dumping in this case is increased by the control action (derivative actions).
PD controller with gravity compensation

- The system evolves, and $V$ decreases, as long as $\dot{q} \neq 0$. Since $\dot{V}$ does not depend on $q$ ($\dot{V} = -\dot{q}^T(D + K_D)\dot{q}$), it is not possible to guarantee that in steady state (when $\dot{q} = 0$) also $\ddot{q} = 0$. On the other hand, the steady state can be computed from the system equation

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + D\dot{q} + g(q) = g(q) + K_P\ddot{q} - K_D\dot{q}$$

*Robot dynamics

*PD+g(q)*

In fact, in steady state ($\dot{q} = \ddot{q} = 0$) it results

$$K_P\ddot{q} = 0$$

that is ($K_P$ is positive definite):

$$q = q_d$$

- A perfect compensation of the gravity term $g(q)$ is necessary, otherwise it is not possible to guarantee the stability of the system (*robust control* problem).
Inverse dynamics control

The manipulator is considered as a nonlinear MIMO system described by

\[ M(q)\ddot{q} + C(q, \dot{q})\dot{q} + D\dot{q} + g(q) = u \]

or, in short:

\[ M(q)\ddot{q} + n(q, \dot{q}) = u \]

The goal is now to define a control input \( u \) such that the overall system can be regarded as a linear MIMO system.

This result (global linearization) can be achieved by using a nonlinear state feedback. It can be shown that this is possible because:

- the model is linear in the control input \( u \);
- the matrix \( M(q) \) is invertible for any configuration of the manipulator.

Let us choose the control input \( u \) (based on the state feedback):

\[ u = M(q)y + n(q, \dot{q}) \]
Inverse dynamics control

By using the control input $u$ defined as

$$u = M(q)y + n(q, \dot{q})$$

it follows that

$$M\ddot{q} + n = My + n$$

and thus (since $M$ is invertible)

$$\ddot{q} = y$$

where $y$ is the new input of the system.
Inverse dynamics control

This is called an inverse dynamics control scheme because the inverse dynamics of the manipulator must be calculated and compensated.

As long as \( y_i \) affects only \( q_i \) (\( y_i = \ddot{q}_i \)), the overall system is linear and decoupled with respect to \( y \).

\[
y \rightarrow \int \dot{q} \rightarrow \int q
\]

Now, it is necessary to define a control law \( y \) that stabilizes the system. By choosing

\[y = -K_P q - K_D \dot{q} + r\]

from \( \ddot{q} = y \) it follows

\[\ddot{q} + K_D \dot{q} + K_P q = r\]

that is asymptotically stable if the matrices \( K_P, K_D \) are positive-definite.
Inverse dynamics control

If matrices $K_P, K_D$ are diagonal matrices defined by

$$K_P = \text{diag}\{\omega^2_{ni}\} \quad K_D = \text{diag}\{2\delta_i\omega_{ni}\}$$

the dynamics of the $i$-th component is characterized by the natural frequency $\omega_{ni}$ and by the damping coefficient $\delta_i$.

A predefined trajectory $q_d(t)$ can be tracked by defining

$$r = \ddot{q}_d + K_D \dot{q}_d + K_P q_d$$

Then, the dynamics of the tracking error is:

$$\ddot{\tilde{q}} + K_D \dot{\tilde{q}} + K_P \tilde{q} = 0$$

The error is not null if and only if $\tilde{q}(0) \neq 0, \dot{\tilde{q}}(0) \neq 0$ and converges to zero with a dynamics defined by $K_P, K_D$. 
Inverse dynamics control

Two control loops are present:

- the first loop is based on the nonlinear feedback of the state and it provides a linear and decoupled model between $y$ and $q$ (double integrator);
- the second loop is linear and is used to stabilize the whole system; the design of this outer loop is quite simple because it has to stabilize a linear system.

As the inverse dynamics controller is based on state feedback, all the terms in the manipulator dynamic model $(M(q), C(q, \dot{q}), D, g(q))$ must be known and computed in real-time.
Inverse dynamics control

This kind of controller has some implementation problems:

- it requires the exact knowledge of the manipulator model (including payload, non-modeled dynamics, mechanical and geometrical approximations, ...);
- the real-time computation of all the dynamic terms involved in the control loop.

If, for computational reasons, only the principal terms are considered, then the control action cannot be precise due to the introduced approximations. It follows that control techniques able to compensate modeling errors are required:

- Robust control (sliding mode, ...)
- Adaptive control.
Inverse dynamics control - Example

Nonlinear system: \((2 + \sin q)\ddot{q} + \dot{q}^3 \sqrt{1 - 0.5 \cos q} + \sqrt{1 + q^2} = u\)

Desired trajectory: trapezoidal velocity profile.

\[ k_p = 100, \quad k_d = 14 \]