**Kinematics and dynamics of a robot arm - Reminder**

A robotic arm is an **open kinematic chain**

- Each link ($L_i$) is connected to the other links by means of a joint ($J_i$) that allows a relative motion of the two links.
- Usually, each joint is actuated and therefore it is possible to control the position of each joint.
- A variable $q_i$ is associate to each joint. It represent the relative position of the $i$-th joint with respect to the $(i-1)$-th joint.
Consider a robot arm with $n$ degrees of freedom (DOF). The following notation will be adopted:
\[
q = \begin{pmatrix} q_1 \\ \vdots \\ q_n \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ \vdots \\ x_6 \end{pmatrix}, \quad t = \begin{pmatrix} t_1 \\ \vdots \\ t_6 \end{pmatrix}, \quad w = \begin{pmatrix} w_1 \\ \vdots \\ w_6 \end{pmatrix}
\]

- joint variables vector $q$
- configuration of the end-effector $x$
- Twist of the end-effector $t$
- Wrench applied to the end-effector $w$

\[
J(q) = \text{Jacobian of the robot}
\]
There are several ways of describing the dynamics of a robotic arm. We will focus on the so called Eulero-Lagrange model. The general dynamic model of a robotic arm is given by the following nonlinear differential equation:

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

- It can be obtained by exploiting the Lagrangian formalism.
- It generalizes the Newton’s equations for a mass subject to the gravity force to the case of several interconnected rigid body.

**Inertia matrix.** It takes into account the effect of the mass and of the inertia of the links, it depends on the configuration.

**Centrifugal forces, Coriolis forces.** It depends both on the configuration and on the velocity.

**Friction.** It takes into account the friction that is present in the robot (e.g. the viscous friction on the joints).

**Gravity effect.** It depends on the configuration.
**Robot Control**

**Robot control:** Determine the torques to be applied at the joints such that the end-effector achieves a desired task.

The mechanical structure of a manipulator has a big impact on the way a robot has to be controlled. For example, controlling a Cartesian robot is much easier than controlling an anthropomorphic robotic arm.

The actuation system has also an impact on the way a robot has to be controlled. The use of DC motors with gearboxes characterized by a high gear ratio tends to decouple the joints and to attenuate the effect of the nonlinearities. Nevertheless, the mechanical transmission can introduce friction, backlash and other undesired effects. The use direct drive actuators eliminates the side effects due to the mechanical transmission but nonlinearities are not attenuated.

---

**Main control problems**

**Position Control:** Take the robot to a target position (regulation) or make the robot tracking a desired trajectory.

It can be done either in the joint space or in the task space.

**Interaction Control:** Control the way the robot interacts with the environment
Position control – Joint space

**PRO:** Simplifications in the design of the controller

**CON:** It is necessary to transform the setpoint of the end effector into a setpoint for the joints.

There are two main control philosophies for the joint space control:

1. **Decentralized control (or independent joint control):** Each joint is considered as a SISO system and its position is controlled independently of the position of the other joints. The coupling between the links is considered as a disturbance.

2. **Centralized control:** The coupling among the positions of the joints and the nonlinear dynamics of the robot are explicitly considered.

The generalized forces $\tau_i$ applied to the joints are generated by actuators connected to the joints by some mechanical driving elements $K_r$ is a diagonal matrix $(n \times n)$. The elements on the diagonal are much greater than 1 in such a way to amplify the torque driven to the joint.

\[
\begin{align*}
q_m &= K_r q \\
\tau_m &= K_r^{-1} \tau
\end{align*}
\]
Independent joint control

The matrix $M(q)$ can be always split as:

$$M(q) = \bar{M} + \Delta M(q)$$

where $\bar{M}$ is a constant diagonal matrix. The terms on the diagonal are the inertias seen by the joints.

Plugging the reduction equations into the Euler-Lagrange model of a robot, we have that:

$$\tau_m = K_r^{-1}\dot{\bar{M}}K_r^{-1}\ddot{q}_m + D_m\dot{q}_m + d$$

where

$$D_m = K_r^{-1}DK_r^{-1}$$

is the matrix of the viscous friction from the motors perspective.

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

is the nonlinear term that depends on the configuration and on its derivatives.
**Independent joint control**

The system made up by the manipulator and by the mechanical part of the actuation system can be decomposed into two subsystems.

The first has an input $\tau_m$ and an output $q_m$, and is *linear and decoupled*. This means that

1. the $i$-th input $\tau_i$ influences only the $i$-th joint variable $q_{mi}$
2. The evolution of $q_{m}$ does not influence the evolution of $q_{mj}$
3. The evolution of $q_{mj}$ is represented by a LTI system

The second subsystem receives as inputs $q_m$ and its first and second derivative, provides an output $d$ and it is *nonlinear and coupled* since it takes into account the nonlinear mechanical couplings of the robot.

---

**Independent joint control**

The main idea behind the independent joint control is to consider the robot as $n$ independent systems (the joints) and to treat and control each joint as a linear SISO system using linear control techniques. The contribution $d$ coming from the nonlinear part of the robot is treated as a disturbance and the controller has to be designed in such a way to attenuate the effect of the disturbance on the controlled system.

This strategy works well if $K_r >> 1$ and if the velocities are not too big.
Independent joint control – Joint Modeling

We will consider the case, very common in practice, of a joint actuated by a DC motor.

Armature circuit

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

\( v_A, i_A, L_A \) and \( R_A \) are the voltage, the current, the inductance and the resistance of the armature circuit respectively. \( v_M \) is the back EMF.

Rotor

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

\( I \) is the inertia of the rotor and \( b \) is the overall damping coefficient.

Coupling between the electrical and the mechanical domain

\[
\begin{align*}
\tau_m(t) &= k_i i_A(t) \\
v_m(t) &= k_i \omega_m(t)
\end{align*}
\]

It is useful to represent the motor by a transfer function. Using the Laplace transform, we can write:

\[
\begin{align*}
V_A(s) &= R_A I_A(s) + sL_A I_A(s) + V_M(s) \\
sI \Omega_m(s) &= C_m - b \Omega_m(s) \\
C_m(s) &= k_i I_A(s) \\
V_m(s) &= k_i \Omega_m(s)
\end{align*}
\]
Independent joint control - Joint Modeling

\[ I_A(s) = \frac{V_A(s) - V_M(s)}{R_A + sL_A} \]
\[ \Omega_m(s) = \frac{1}{C_m(s)} = \frac{1}{b + sI} \]

\[ V_A(s) + \frac{1}{R_A + sL_A} \]
\[ k_i \]
\[ C_m(s) \]
\[ \frac{1}{b + sI} \]
\[ \Omega_m(s) \]
\[ \frac{1}{s} \]
\[ \Theta_m(s) \]

Reducing the block diagram we obtain

\[ \Omega_m(s) = \frac{k_t}{(b + sI)(R_A + sL_A) + k_i k_v} V_A(s) = G(s)V_A(s) \]

Usually \( b \approx 0 \) and therefore:

\[ G(s) = \frac{k_t}{s^2 L_A I + sR_A I + k_i k_v} \]

The system has two poles

\[ p_{1,2} = \frac{-R_A I \pm \sqrt{(R_A I)^2 - 4L_A I k_i k_v}}{2L_A I} \]
Independent joint control – Joint Modeling

In practice $L_A$ is small enough to have $(R_A I)^2 - 4L_A I k_I k_v > 0$ and, therefore, to have $G(s)$ with two real poles. Doing some approximations, due to the fact that $L_A$ is small, we have that:

$$p_1 = \frac{k_I^2}{R_A I} \quad \quad p_2 = -\frac{R_A}{L_A}$$

$p_1$ is the electromechanical pole while $p_2$ is the electric pole.

Putting in evidence the time constants, $G(s)$ can be rewritten as:

$$G(s) = \frac{1/k_I}{(1/T_m s)(1 + T_e s)}$$

where

$$T_m = \frac{R_A I}{k_I k_v} \quad \quad T_e = \frac{L_A}{R_A}$$

mechanical time constant \quad electric time constant

Usually $T_e << T_m$ and therefore the electric pole can be neglected and it is possible to further simplify $G(s)$ as:

$$G(s) = \frac{1/k_I}{1 + T_m s}$$
Ac actuated joint of a robotic arm can be modeled by the following block diagram:

\[
\begin{align*}
V_A(s) & \quad D \\
\quad k_t R_A & \quad 1 \\
\quad 1 & \quad \Omega_m \\
\quad 1 & \quad \Theta_m \\
& \quad \Theta_m
\end{align*}
\]

The transfer function that links the position of the rotor to the input voltage is given by:

\[
\frac{\Theta_m(s)}{V_A(s)} = \frac{1}{s} \frac{\Omega_m(s)}{V_A(s)} = \frac{k_m}{s(1 + T_m s)}
\]

The position of the joint is given by

\[
\Theta(s) = k_r^{-1} \Theta_m(s)
\]

A model of the system to control has been derived. In order to improve performance, the cascaded control technique will be exploited.

Cascaded control is useful when:

1. The dynamics of the process to control can be decomposed in two or distinct more sub-dynamic
2. The dynamics of \( G_1(s) \) is faster than the dynamics of \( G_2(s) \)
3. It is possible to measure \( v \), the input of \( G_2(s) \)
Independent joint control

In order to control the output $y$, it can be convenient to build a control scheme with more cascaded feedback loops:

The inner loop can compensate part of the dynamics and attenuate the disturbance. In this way $v_d = v$ and the control problem that need to be solved by $R_2(s)$ is easier.

It is a “divide et impera” strategy. Solving several subproblems is usually than solving a single complex problem.

Independent joint control

The cascaded control has several advantages with respect to traditional single loop control:

1. The disturbance can be compensated by the internal loop in such a way to practically eliminate its contribution on the output $y$.

2. Thanks to the internal loop, it is possible to obtain a dynamic relation between $v_d$ and $v$ that is faster than the one that can be obtained between $u$ and $v$.

3. Greater robustness of the inner system (thanks to the inner loop)

4. Possibility of imposing a desired dynamics between $v_d$ and $v$ and, consequently, simplification of the design of $R_2(s)$
**Independent joint control**

The transfer function of the controlled system with a single loop is given by:

\[
H_1(s) = \frac{Y(s)}{Y_d(s)} = \frac{R(s)G_1(s)G_2(s)}{1 + R(s)G_1(s)G_2(s)} = \frac{1}{1 + \frac{1}{R(s)G_1(s)G_2(s)}},
\]

If the system has two cascaded loops, the transfer function of the controlled system is given by:

\[
H_2(s) = \frac{Y(s)}{Y_d(s)} = \frac{R_1(s)R_2(s)G_1(s)G_2(s)}{1 + R_1(s)G_1(s) + R_1(s)R_2(s)G_1(s)G_2(s)} = \frac{1}{1 + \frac{1}{R_2(s)G_2(s)}(1 + \frac{1}{R_1(s)G_1(s)})}.
\]

The cascaded control provides more degrees of freedom in the design of the controlled. Furthermore, if \( R_1(s)G_1(s) \gg 1 \) in the desired bandwidth, it is possible to consider \( v_d \approx v \) and, consequently, to simplify the design of the controller \( R_2(s) \).

---

**Independent joint control**

Consider the actuated joint of a robot:

\[
\begin{align*}
V_A(s) & \quad \frac{k_s}{R_s} \quad \frac{-D}{1} \quad \frac{1}{s} \quad \frac{1}{s} \quad \Theta_m \\
\end{align*}
\]

In order to have an efficient reduction of the disturbance \( D \), the control action has to be such that:

1. The effect of the disturbance on the output has to be highly attenuated.
2. It possess an integral action in order to compensate the effect of the gravity on the output at steady state.
There are three cascaded feedback loops. In each loop there is a controller:

<table>
<thead>
<tr>
<th>Controller</th>
<th>Kind of Controller</th>
<th>Transducer Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_p(s)$</td>
<td>Position Controller</td>
<td>$k_{TP}$</td>
</tr>
<tr>
<td>$C_v(s)$</td>
<td>Velocity Controller</td>
<td>$k_{TV}$</td>
</tr>
<tr>
<td>$C_A(s)$</td>
<td>Acceleration Controller</td>
<td>$k_{TA}$</td>
</tr>
</tbody>
</table>

The control action of the innermost loop must contain an integral action (i.e. a pole in the origin) for compensating the effect of gravity at steady state.
Independent joint control

We will consider the possible control solutions for a single joint. We will first consider a single position loop and we will then add the other control loops to see the benefits introduced by the cascaded control.

Performance Indexes:

1) Output behavior
2) Disturbance attenuation

Consider the case where there is only a position loop:

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

\[ k_{TV} = k_{TA} = 0 \]
Independent joint control

Output Analysis (D'=0)

\[ G(s) = \frac{k_m K_p (1 + s T_p)}{s^2 (1 + s T_m)} \]
\[ H(s) = k_{TP} \]

The system has one zero and three poles. \( \delta, \omega_n \) and \( \tau \) depend on the choices made for the controller parameters.

Three cases are possible, depending on the relation between \( T_P \) and \( T_m \). In each situation, the root locus can be exploited for analyzing the influence of the gain \( k_m K_p k_{TP} \) on the behavior of the closed loop system.

1° CASE: \( T_p < T_m \)

The system is unstable for each possible value of the gain \( k_m K_p k_{TP} \).
**Independent joint control**

**II° CASE:** \( T_p > T_m \rightarrow \delta \omega_n < 1/T_p < 1/\tau \)

The system is stable but gains have to be low for avoiding big oscillations in the output.

---

**Independent joint control**

**III° CASE:** \( T_p >> T_m \rightarrow \delta \omega_n > 1/\tau \approx 1/T_p \)

The zero in \(-1/T_p\) attacks one of the real poles. For the same gains, the output has more damped oscillation with respect to case 2.
Independent joint control

Disturbance rejection analysis ($\Theta_r=0$)

The disturbance rejection characteristics of the closed loop system are described by the transfer function linking the output to the disturbance. This function is given by:

$$\frac{\Theta(s)}{D(s)} = \frac{sR_A}{K_cK_pk_{TP}(1+sT_p)} \frac{\frac{sR_A}{k_mK_pk_{TP}(1+sT_p)}}{1 + \frac{s^2(1+sT_m)}{k_mK_pk_{TP}(1+sT_p)}}$$

The zero in the origin is due to the integral action of the controller. When $\Theta=\text{const}$ it allows to compensate the effect of the gravity.

It is evident that for reducing the effect of the disturbance on the output it is convenient to choose a big $K_p$. The disturbance rejection factor due to the position loop is given by:

$$X_R = K_pk_{PT}$$

In order to minimize the effect of the disturbance it is necessary to choose a very big $K_p$. However, considering the output analysis, it is not convenient to choose very big gains since they may introduce big oscillations. In practice a trade off solution between the transient of the output and the disturbance rejection is chosen.

Consider now the case where a velocity loop is nested into a position loop:

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

$$k_{TA} = 0$$
**Independent joint control**

**Output Analysis (D'=0)**

\[ G(s) = \frac{k_m K_p K_v (1+s T_v)}{s^2 (1+s T_m)} \]

\[ H(s) = k_T P (1 + s \frac{k_T V}{K_p k_T P}) \]

The zero of the controller in \( s = -1/T_v \) can be chosen in such a way to cancel the real pole of the motor by setting \( T_v = T_m \). In this case the transfer function of the controlled system is given by:

\[
\Theta_r(s) = \frac{1}{k_T P} \frac{s k_T V}{K_p k_T P} + \frac{s^2}{k_m K_p k_T P K_v} = \frac{1}{k_T P} \frac{2 \delta \omega_n}{\omega_n} + \frac{s^2}{\omega_n^2}.
\]

Where \( \delta \) and \( \omega_n \), that characterize the output of the system, depend on the control parameters.

---

**Independent joint control**

By properly choosing \( K_p \) and \( K_v \) it is possible to set the poles of the feedback system in any desired configuration and, therefore, it is possible to set an arbitrary fast and arbitrary damped behavior to the output of the system. In particular, if

In particular if \( \delta \) and \( \omega_n \), are given as a specification (i.e. the position of the poles of the feedback loop), \( K_p \) and \( K_v \) are determined by:

\[ K_v K_i = \frac{2 \delta \omega_n}{k_m} \quad K_p k_T P K_v = \frac{\omega_n^2}{k_m} \]
Independent joint control

Disturbance rejection analysis ($\Theta_r=0$)

The disturbance rejection can be characterized by considering the transfer function relating the disturbance to the output:

\[
\Theta(s) = -\frac{sRA}{1 + \frac{sk_rK_pK_TK_V}{K_pK_TK_V(1+T_m)}}
\]

The disturbance rejection factor due to the control loops is given by:

\[
X_R = K_pK_TK_V
\]

The disturbance rejection factor is fixed once the dynamic behavior of the output of the controlled system has been fixed. The transient due to the disturbance vanishes with the same dynamics with which the transient on the output vanishes. The behavior of the transient can be imposed arbitrarily.

Consider the case in which the acceleration loop is nested into the velocity loop:

\[
C_p(s) = K_p \quad C_V(s) = K_V \quad C_A(s) = K_A \frac{1+sT_A}{s}
\]
Independent joint control

Output Analysis (D'=0)

\[ G(s) = \frac{k_m K_p K_v k_A (1 + s T_a)}{s^2 (1 + k_m K_A k_T A) [1 + \frac{s T_m (1 + k_m K_A k_T A)}{1 + k_m K_A k_T A}]} \]

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

The zero of the controller in \( s = 1/T_a \) can be chosen such to cancel the effect of the real pole of the motor in \( s = 1/T_m \) by setting \( T_a = T_m \).

In this case the overall transfer function is given by:

\[ \Theta(s) = \frac{1}{k_{TP}} \]

\[ \Theta_r(s) = 1 + \frac{s k_{TV}}{K_p k_{TP}} + \frac{s^2 (1 + k_m K_A k_T A)}{k_m K_p k_{TP} k_V k_A} = 1 + \frac{2 \delta}{\omega_n} + \frac{s^2}{\omega_n^2} \]

where \( \delta \) and \( \omega_n \), that determine the position of the poles of the system, depend on the control parameters.

Independent joint control

The transfer function relating the disturbance to the output is given by:

\[ \frac{\Theta(s)}{D(s)} = -\frac{s k_A}{K_A k_{TP} k_V k_A (1 + s T_A)} \]

The disturbance rejection factor is given by:

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

Thanks to the third degree of freedom introduced by the acceleration loop, it is possible to set arbitrarily both the dynamic behavior of the output and the disturbance rejection factor.
Independent joint control

Let the specifications be given in terms of $\delta$, $\omega_n$ ( dynamics of the output) and $X_R$ ( desired disturbance rejection). The gains $K_p$, $K_V$ and $K_A$ are determined by the following relations:

$$\frac{2K_pk_{TP}}{k_{TV}} = \frac{\omega_n}{\delta} \quad k_mK_Ak_{TA} = \frac{k_mX_R}{\omega_n^2} - 1 \quad K_pk_{TP}k_VK_A = X_R$$

The addition of the acceleration loop allows to set an arbitrary dynamics of the output and an arbitrary disturbance rejection factor to the controlled system.

Independent Joint Control

When fast trajectories need to be tracked by the joint, the tracking performance of feedback controllers (even if they have several nested loops) are low.

A possible solution is to insert a feedforward control action that exploits the knowledge of the position, velocity and acceleration setpoints.

In case of independent joint control, a feedforward control action is implemented in each controlled joint and, therefore, we call this additional control structure decentralized feedforward control.
Independent Joint Control

Feedback controllers
- Disturbance rejection, robustness with respect to parameters variations, higher bandwidth.

Feedforward controllers (Open loop controllers)
- They are not based on the feedback of a signal, they are based on the perfect knowledge of the model of the plant, theoretically they allow a perfect tracking of a setpoint.

Since the model of the plant IS NEVER perfectly known, it is not possible to use control schemes with only feedforward control actions.

The main idea of the feedforward compensation is to build a hybrid control scheme that contains both feedback control actions and feedforward control actions. In this way it is possible to combine the benefits of the feedback with those of the feedforward.
Usually a feedforward control action is added to a feedback controller for:

1. Properly filtering the setpoint, modifying the transfer function between \( y \) and \( y_d \) for achieving desired static and dynamic characteristics for the controlled system.

2. Improving the achievable performance in terms of tracking

3. Compensate known (or measurable) disturbances acting on the system

One of the main objectives of a control scheme is to achieve a perfect tracking, namely \( y = y_d \). In feedback control scheme this is not always achievable since the control action is built on the basis of \( e = y_d - y \). The use of a feedforward term provides two more degrees of freedom \((C_f(s) eC_a(s))\) to the designer. The extra degrees of freedom allow to achieve better performance and perfect tracking.

When using a feedforward control action we have that:

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

Setting the perfect tracking condition, \( Y(s) = Y_d(s) \), we obtain:

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

A possible choice is:

\[
C_f(s) = 1 \quad C_a(s) = G^{-1}(s)
\]
**Independent Joint Control**

Thus, given the model of the plant $G(s)$ and the setpoint $Y_d(s)$, in order to achieve a perfect tracking it is necessary to "invert" the plant dynamics. The control action $U(s)$ must contain the term $G^{-1}(s)Y_d(s)$.

This control action is not always implementable. In fact if:

1. $G(s)$ has zeros with a positive real part (non minimum phase systems)
2. $G(s)$ has significant delays (i.e. it contains terms as $e^{-Ts}$)
3. $G(s)$ is not a proper system (i.e. if $n_{\text{poles}} > n_{\text{zeros}}$)

the feedforward control action is not realizable. In the first case an unstable controller would be necessary, while in the second and in the third case anticipative controllers would be required.

In these cases, approximations are used. $C_d(s)$ is built in such a way to be equal to $G^{-1}(s)$ in the desired frequency band.

---

**Independent Joint Control**

When considering the problem of controlling the position of an electric motor, the transfer function of the plant can be approximated, disregarding multiplicative constants, with $G(s)=1/s^2$. Thus, the feedforward control action, which must contain the term $G^{-1}(s)=s^2$, is not physically realizable since it contains a double derivation on the position setpoint.

In practice, this is not a problem since, when the position setpoint is computed, it is possible to obtain also the velocity and acceleration setpoints. **It is not necessary to derive the setpoint since its derivatives are already available.**

In the **ideal case**, namely if $G^{-1}(s)$ is physically realizable and perfectly known, the feedback loop and the controller $R(s)$ would be useless for tracking the reference and they would contribute only for the attenuation of the disturbances acting on the system. In **real applications** the contribution of the feedback controller is **crucial** for compensating the effect of the imperfect knowledge of the model, of the imperfect realizability of the feedforward control action and of the disturbances acting on the system.
Consider:

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

The feedforward control action is not realizable since it should be:

\[
C_d(s) = G^{-1}(s) = s + 10
\]

Approximate the ideal feedforward control action:

\[
C_d(s) = 10
\]

In this way we have a perfect tracking for \( y_d \text{=} \text{const.} \)
The introduction of the feedforward, even if approximated, significantly improves the performance in terms of tracking.

The ideal feedforward controller is given by $C_a(s)=s+10$ and therefore, the feedforward control action should ideally be:

$$(s + 10)Y_d(s) = sY_d(s) + 10Y_d(s)$$

From this relation, recalling that multiplying by $s$ in the Laplace domain corresponds to deriving in the time domain, it follows that if the derivative of the setpoint is available (e.g., the velocity), then it is possible to implement the ideal feedforward using the following control scheme with $k_{a,p}=10$ and $k_{a,v}=1$.
**Independent Joint Control**

\[ y_d \text{ (dashed)} \quad \text{and} \quad y \text{ (solid)} \]

Since the ideal feedforward has been implemented, perfect tracking has been achieved.

---

**Pressure control in a Methane conversion system**

**Methane conversion system**

- Fuel tank
- Pressure reduction
- Gas control using
- Rail
- Filter
- Commutator

---

Cristian Secchi
Pressure control in a Methane conversion system

Mechanical pressure controllers

- Control done by a valve, piston and spring
- Robust systems
- Made up of one or more stages

Limits:

- Impossible to change the output pressure with respect to the nominal value
- Impossible to tune them while they are working
- Presence of elements sensitive to temperature

Control structure
Control Realization

- The pressure measured in the rail is used to build a pressure error $\Delta P$ given as an input to the controller.
- The rail requests a variable gas flow.
- The value of $P_n$ is considered as a disturbance.

Control architecture

PID control

Results achieved with a PID

After having optimized the gains $K_p$, $K_i$ e $K_d$ the following results are obtained:

- The system behaves well in terms of output pressure oscillations if the requested gas flow is constant.
- If the requested gas flow changes, the system does not satisfy the stability specifications in terms of stability of $P_{out}$. 
Feedforward Control

A feedforward controller is added to the feedback control architecture

Results

Behavior of the output pressure considering variable gas flow requests

- Requested gas flow
  - \( P_{in} = 260 \) [bar]
  - \( P_{target} = 3 \) [bar]

- Output pressure (\( P_{out} \))
Consider a joint controlled only with a single position loop.

As a first approximation, the relation linking the acceleration to the input voltage is given by:

\[ \ddot{\theta} = \frac{k_t}{R_A I} V_A = \frac{k_m}{T_m} V_A \]

and the one that relates the velocity to \( V_A \) is given by:

\[ \dot{\theta} = k_m V_A \]

Thus, inverting the approximated relations, it is possible to build the overall control scheme.
**Independent Joint Control**

By simply computations, it is possible to show that the insertion of the feedforward is equivalent to consider a control scheme with a position loop and with a setpoint given by:

\[
\Theta'_d(s) = [k_{TP} + \frac{s^2(1+sT_m)}{k_mK_P(1+sT_P)}]\Theta_d(s)
\]

Velocity and acceleration values are necessary for computing the new setpoint. They can be easily computed, if not already available, if the trajectory is expressed in an analytical form.

**Independent Joint Control**

Consider the case with a position and a velocity loop. Following the same steps as before, the following control scheme can be obtained:

\[
\Theta'_d(s) = \frac{k_mK_V}{s} + \frac{k_T V_A(s)}{s} + \frac{k_{TP} V_A(s)}{s} + \frac{k_P}{s} (\Theta_m(s) - \Theta_m(s))
\]
**Independent Joint Control**

By simply computations, it is possible to show that the insertion of the feedforward is equivalent to consider a control scheme with a position loop and with a setpoint given by:

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

Velocity and acceleration values are necessary for computing the new setpoint. They can be easily computed, if not already available, if the trajectory is expressed in an analytical form.
Independent Joint Control

By simply computations, it is possible to show that the insertion of the feedforward is equivalent to consider a control scheme with a position loop and with a setpoint given by:

$$\Theta'(s) = [k_{TP} + \frac{s k_{TV}}{K_p} + \frac{s^2(1 + k_m K_A k_{TA})}{k_m K_p K_v K_A}] \Theta_d(s)$$

Velocity and acceleration values are necessary for computing the new setpoint. They can be easily computed, if not already available, if the trajectory is expressed in an analytical form.

---

Independent Joint Control

The more the control loops, the less the knowledge about the model for implementing the feedforward control action is requested.

**Position loop**: $T_{mv} k_m$

**Position and velocity loops**: $k_m$

**Position, velocity and acceleration loops**: $k_{mv}$ but with a reduced weight

The addition of the feedforward actions significantly improves the performance in terms of tracking. The improvements are the better, the more accurate is the knowledge about the model.
Independent Joint Control

It is possible to build control schemes that are equivalent to the ones presented so far and that use only a position feedback and standard controllers (P, PI, PID, PID²).

The two alternative control structures are equivalent in terms of disturbance rejection and trajectory tracking. However, the elimination of the internal loops, drastically decreases the possibility of acting on the dynamics of the internal variable and makes less intuitive the choice of the control gains.

Nevertheless, since the PID is the most used controller in industries, the control schemes with standard controllers are the most used.

Independent Joint Control

The control schemes illustrated so far allow an asymptotic tracking of a desired trajectory in case of no disturbance acting on the joint.

The inevitable presence of a disturbance produces an error that deteriorates the trajectory tracking performance.

So far, the disturbance has been considered as an unknown exogenous input acting on the joint and the controllers have been designed to attenuate its effect as much as possible.
Independent Joint Control

Actually, things can be improved because the disturbance acting on the joint has a well known expression and it can be computed once the model of the robot is known.

Let $q_{md}, \dot{q}_{md}, \ddot{q}_{md}$ be the acceleration, velocity and position setpoints for ALL the motors of the joints. The following feedforward action can be computed.

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

namely the analytic expression of the disturbance using the setpoints.

This feedforward control action tends to compensate the effect of the disturbance acting on the joints because of the mechanical coupling of the manipulator. This compensation is called precomputed torque feedforward compensation.

Control scheme with precomputed torque feedforward compensation
Independent Joint Control

- Even if the residual disturbance term $d_i = d_{q_d} - d$ vanishes only in case of perfect tracking ($q=q_d$) and on perfect knowledge of the model, the residual disturbance $d_i$ is much smaller than $d$. Thus, the precomputed torque compensation technique reduces the effect of the coupling disturbance, decreases the disturbance rejection effort requested to the feedback controllers and allows to use lower gains.

- The calculation of the feedforward action $d_{q_d}$ is computationally demanding (it is a centralized term) and the computation time may exceed the sampling time.

- A possible solution to this problem is to make a partial compensation and to compute only the most significant terms of the disturbance, namely those due to inertia and to gravity.

Decentralized Control

- In the independent joint control, each joint is considered separately.

- Actually, a manipulator is not a set of $n$ decoupled systems but it is a multivariable system with $n$ inputs (the torques on the joints).

- If the operation velocity are very high and/or if the reduction gain $K_r$ is close to 1 (direct drive motors) then the nonlinear terms due to the mechanical coupling influence significantly the dynamics of the system and, therefore, considering the components of $d$ is no more admissible.
Centralized Control

- Centralized control algorithms consider the robot as a whole and exploit its characteristics, linear and nonlinear, for achieving the desired objectives.

- The centralized approach is more rigorous than the decentralized one and it allows to design non linear controllers that allow to obtain global stability and perfect tracking of desired setpoints.

- The performance that can be obtained using centralized control algorithms is better than the ones that can be obtained by decentralized controllers. Thus, centralized control becomes necessary when considering advanced application, high precision positioning or tracking of very fast trajectories.

- The precomputed torque compensation is an example of centralized controller.

In order to build a centralized controller, it is necessary to take into account the dynamic model of the robot:

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

The actuation (DC motors) and reduction system is governed by:

\[
\begin{align*}
K_t^{-1}\tau &= K_i i_a \\
v_a &= R_a i_a + K_v \dot{q}_m \\
K_v \dot{q} &= \dot{q}_m \\
v_a &= G_v v_c
\end{align*}
\]

- \(K_t\): torque constants matrix
- \(K_i\): reduction matrix
- \(R_a\): armature resistance matrix
- \(K_v\): voltage constants matrix
- \(G_v\): amplifiers gains matrix
- \(v_c\): vector of the control voltages of the servomotors
Centralized Control

By simple computations, the model of the manipulator together with its actuation system is given by:

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

where

\[ D = D_v + K_r K_r^{-1} K_i K_r \quad u = K_r K_r^{-1} G_v v_c \]

Centralized Control

- The dynamic structure of the system to be controlled is the same
- The overall system is controlled using a voltage

\[ v_c \rightarrow K_r K_r^{-1} G_v \rightarrow Manipolatore \]

\[ K_r K_r^{-1} K_i K_r \]
Centralized control

Using a voltage controlled system, the torques provided to the joints depend on matrices $K_v$, $K_t$ and $R_A$ of the motors that are influenced by operative conditions (e.g. temperature).

In order to reduce the sensitivity to the parametric variations, it is convenient to consider current controlled actuation systems. In this case the actuators behave as controlled torque generators:

$$I_a = G_i v_c$$

Thus, the armature current rather than the armature voltage is imposed.

Centralized Control

By simple computations, the model of the manipulators together with its actuation system is given by:

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

where

$$D = D_v$$

$$u = K_r K_i G_i v_c = \tau$$

The overall system is torque controlled.

In the controllers that will be illustrated in the following, the control output is the torque, $v_c$ can be achieved by simply dividing the output of the controller by the gain constants.
**PD + gravity compensation**

- It is a very simple centralized control
- It combines the linear action of the PD controller with a nonlinear compensation term
- It solves the regulation problem, namely the problem of taking the manipulator in a desired configuration which is globally asymptotically stable.
- It is useful for tasks where a precise positioning of the end-effector is requested (e.g. pick and place)

Let $q_d=(q_1, ..., q_n)^T$ be a configuration where the robot needs to be positioned. In other words, the controller needs to guarantee that the point $q_d$ is an asymptotically stable equilibrium point for the system described by:

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

Gravity tends to take the robot towards a configuration corresponding to a minimum of the gravitational potential energy.

The main idea is to compensate the force due to the gravitational energy and to introduce a new force that takes the robot in $q_d$. 
PD + gravity compensation

From an energetic point of view, the idea is to make an energy shaping, namely to shape the energy of the controlled system in such a way that it has the desired form.

If left free to evolve, the manipulator tends to go in a configuration corresponding to the minimum of the (gravitational) potential energy. The goal of the controller is to impose to the controlled system a "desired" potential energy, with a minimum in the desired configuration \( q_d \).

The control law compensates the gravitational term \( g(q) \) and replaces it with an elastic term.

\[
u = g(q) + K_P(q_d - q)\]

where \( K_P \) is a positive definite gain matrix and it can be interpreted as a stiffness of an elastic effect introduced on the joints.

PD + gravity compensation

The behavior of the controlled system is given by:

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

from which:

\[
M(q)\ddot{q} + C(q, \dot{q})\dot{q} = K_P(q_d - q)
\]

The term relative to the gravitational potential has been replaced by a term associated to an elastic potential.

The configuration \( q_d \) is an equilibrium configuration for the manipulator. The regulation problem is solved if \( q_d \) is asymptotically stable.
**PD + gravity compensation**

Defin the following variable:

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

Considering the change of variable

$$\begin{pmatrix} q \\ \dot{q} \end{pmatrix} \rightarrow \begin{pmatrix} \ddot{q} \\ \dot{q} \end{pmatrix}$$

it is possible to study the asymptotic stability of $q_d$ by studying the asymptotic stability of the origin of the system in the new coordinates.

---

**PD + gravity compensation**

Consider the following candidate Lyapunov function:

$$V(\ddot{q}, \dot{q}) = \frac{1}{2} \ddot{q}^T M(q) \dot{q} + \frac{1}{2} \ddot{q}^T K P \ddot{q}$$

The function is composed by two terms:

- $\frac{1}{2} \ddot{q}^T M(q) \dot{q}$: The kinetic energy of the system
- $\frac{1}{2} \ddot{q}^T K P \ddot{q}$: The elastic potential energy introduced by the controller.
**PD + gravity compensation**

Since $M(q)$ and $K_P$ are positive definite for every $q$, the function $V$ is positive definite. Consider now its orbital derivative. Taking into account that:

$$\dot{q} = -\ddot{q}$$

we have that:

$$\dot{V}(\ddot{q}, \dot{q}) = \dot{q}^T M(q) \ddot{q} + \frac{1}{2} \dot{q}^T \dot{M}(q) \dot{q} - \dot{q}^T K_P \ddot{q}$$

but:

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

---

**Controllo PD + Compensazione di gravità**

Thus:

$$\dot{V}(\ddot{q}, \dot{q}) = \dot{q}^T (K_P \ddot{q} - C(q, \dot{q}) \dot{q} - D \dot{q}) + \frac{1}{2} \dot{q}^T \dot{M}(q) \dot{q} - \dot{q}^T K_P \ddot{q}$$

from which

$$\dot{V}(\ddot{q}, \dot{q}) = \frac{1}{2} \dot{q}^T (\dot{M}(q) - 2C(q, \dot{q})) \dot{q} - \dot{q}^T D \dot{q}$$

but, for the properties of the Euler-Lagrange model:

$$\dot{q}^T (\dot{M}(q) - 2C(q, \dot{q})) \dot{q} = 0$$

therefore

$$\dot{V}(\ddot{q}, \dot{q}) = -\dot{q}^T D \dot{q}$$
PD + gravity compensation

Since D is a positive definite matrix, the orbital derivative of V is negative definite and, therefore, \( q_d \) is an asymptotically stable equilibrium configuration for the controlled system. Since the Lyapunov function is radially unbounded, the equilibrium point is GAS.

The robot goes to the desired configuration with the same velocity as the one with which V goes to 0. This velocity depends on matrix D, the damping of the robot. In order to impose an arbitrary convergence velocity, it is useful to introduce a derivative term in the control action:

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

where \( K_D \) is a positive definite matrix that can be interpreted as an additional friction introduced by the controller.

PD + gravity compensation

In this way, considering the same Lyapunov function as before and making the same computations, we obtain:

\[
\dot{V}(\bar{q}, \dot{q}) = -\dot{q}^T(D + K_D)\dot{q}
\]

By properly choosing the gain matrix \( K_D \) it is possible to control the velocity with which the robot tends to the desired configuration \( q_d \).
The control action is composed by a nonlinear term, compensating gravity, and by a linear PD term. The target configuration $q_d$ is GAS for all positive definite $K_P$ and $K_D$.

- The derivative control action ($K_D$) is necessary for systems characterized by low friction.

- The computation of the term has to be very accurate otherwise the control strategy does not allow to stabilize $q_d$. 

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

$m_i = 1 \text{ Kg}$
$q_i = \text{i-th joint variable}$
$l_i = 1 \text{ Nsec}\text{^2/} \text{rad}\text{^2}$
$q_i = 1 \text{ m}$
$a_{ci} = 0.5 \text{ m}$
$g = -9.8 \text{ m/sec}\text{^2}$
$t_i = \text{i-th torque}$

$$0 \mathbf{H}_2 = \begin{pmatrix}
C_{12} & -S_{12} & 0 & a_1 C_1 + a_2 C_{12} \\
S_{12} & C_{12} & 0 & a_1 S_1 + a_2 S_{12} \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix} = \begin{pmatrix}
C_{12} & -S_{12} & 0 & C_1 + C_{12} \\
S_{12} & C_{12} & 0 & S_1 + S_{12} \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}$$

Example

Starting configuration

$q_1 = \frac{x}{2} \text{ rad}$
$q_2 = -\frac{x}{2} \text{ rad}$
$x = 1 \text{ m}$
$y = 1 \text{ m}$
Example

Pick Configuration: $x = 1.71 \text{ m } y = 0.71 \text{ m}$

Place configuration: $x = 0 \text{ m } y = 1.41 \text{ m}$

Example

Controller parameters

- $K_{Px} = K_{Py} = 60 \text{ N/m}$
- $K_{Dx} = K_{Dy} = 20 \text{ Nsec/m}$
Example

Example
**Example**

![Graph of \( \tau_c \) complex in 3D space](image1)

![Graph of \( \tau_c \) complex in 3D space](image2)

**Example**

![Graph of forces in 2D space](image3)

![Graph of forces in 2D space](image4)
Computed torque control

- It is a centralized control strategy
- Useful for trajectory tracking problems
- It requires an accurate knowledge of the model of the manipulator
- It is based on the global linearization of the robot

The dynamic model of the robot can be rewritten as:

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

The main idea is to find a control action that globally linearizes the system, namely that transforms a nonlinear system in a MIMO linear system.

This is possible since \( M(q) \) is always invertible and it can be done by a nonlinear state feedback.
**Computed torque control**

Consider the following control action

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

where \( y \) represents a part of the input vector that still to be determined. Using this input, we obtain:

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

\[ \ddot{q} = y \]

Using the proposed control action, the controlled system has a dynamic behavior described by:

\[ \ddot{q} = y \]

where \( y \) is the part of the input that still to be determined.

The controlled system is linear and decoupled with respect to the input \( y \). The component \( y_i \) of the input vector influences only the component \( q_i \) of the joint variable.
**Computed torque control**

It is necessary to determine an input $y$ that makes the linear system asymptotically stable. If we choose:

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

we obtain:

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

where $r$ is a part of the input still to be determined. If $K_P$ and $K_D$ are positive definite, the linear system is asymptotically stable. If the system is left free to evolve ($r=0$), it evolves towards the GAS configuration $q=0$. By properly choosing the gain matrices $K_P$ and $K_D$, it is possible to assign an arbitrary dynamics to the system. In other words, it is possible to assign the eigenvalues of the state matrix describing the linear system.

---

**Computed torque control**

Let $q_d(t)$ be a desired trajectory to track. Setting:

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

we obtain

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

where $\ddot{q} = q_d - q$

Thus, choosing $K_D$ and $K_P$ positive definite, the tracking error dynamics is asymptotically stable and, therefore, the tracking error tends to 0 with a dynamic that can be arbitrarily chosen by properly choosing $K_P$ and $K_D$.

Using the computed torque controller, very good trajectory tracking performance can be achieved.
Computed torque control

There are two nested loops: the inner loop is a nonlinear state feedback that makes the controlled system linear and decoupled. The outer loop is a linear state feedback that stabilizes the overall system.

For implementing this control strategy it is necessary to compute in real-time and in a very accurate way all the terms of the dynamic model of the manipulator. This can be problematica and it is the weak point of the computed torque control.

There are other control techniques that do not need a precise knowledge of the model of the robot (adaptive control, sliding mode control)
**Example**

Joint position

Torques

\[ K_P = \begin{pmatrix} 100 & 0 \\ 0 & 100 \end{pmatrix} \quad K_D = \begin{pmatrix} 20 & 0 \\ 0 & 20 \end{pmatrix} \]

- \( m_i = 1 \text{ Kg} \)
- \( q_i = \text{i-th joint variable} \)
- \( l_i = 1 \text{ Nsec}^2/\text{rad}^2 \)
- \( a_i = 1 \text{ m} \)
- \( a_{\text{CI}} = 0.5 \text{ m} \)
- \( g = -9.8 \text{ m/sec}^2 \)
- \( t_i = \text{i-th torque} \)
Control in the task space

- So far we have assumed that the setpoint and its derivatives were given in the joint space.
- Usually the motion specifications are given in the task space and not in the joint space.
- It is necessary to do inverse kinematics for obtaining the setpoint for the control schemes designed in the joint space.

The change of coordinates is computationally demanding, especially if it is necessary to obtain also velocity and acceleration setpoints. Usually, the position setpoint is obtained by kinematic inversion and the velocity and acceleration setpoints are obtained through numeric derivation algorithms.

Alternatively, it is possible to consider control schemes built directly in the task space. In these schemes, the quantities measured at the joint are transformed, using direct kinematics, into the position in the task space. The control error is then built in the task space and it is used for generating the torques to be applied on the joints.

More powerful control unit or bigger sample times.

Control of interaction

Computationally demanding

Very useful for some applications
Control in the task space

Scheme 1

\[ J^{-1}(q) \] Controller Manipulator

\[ f(\cdot) \]

\( J^{-1}(q) \) allows to obtain the error in the joint space starting from the error in the operating space. The controller is implemented in the joint space. a nello spazio di giunto.

This scheme works well if the control errors are small.

It is necessary to be careful to stay away from singularity configurations.

Scheme 2

\[ J^T(q) \] Controller Manipolatore

\[ f(\cdot) \]

The controller is implemented directly in the task space, starting from the error \( \Delta x \). The control action is a wrench to apply to the end-effector and it is converted into a torque vector by \( J^T(q) \).

There are no implementation problems when the robot approaches singularity configurations.
Control in the task space

The PD + gravity compensation controller can be reformulated in the task space. The main idea behind the controller is very intuitive.

Problem: Design a controller in the task space that takes the end-effector to a desired configuration.

The controller needs to act as a spring-damper system. The spring pulls the end effector towards the target and the damper dampens the oscillations taking the robot to stop in the target configuration.

\[ K_P(x_d - x) \] Action proportional to the position error

\[ K_D(\dot{x}_d - \dot{x}) = -K_D\dot{x} \] Action proportional to the derivative of the position error

- A PD controller would be enough without gravity
- The gravity force “disturbs” the robot by “pulling it down”
- The gravity effect needs to be compensated.

\[ u = J^T(q)[K_P(x_d - x) - K_D\dot{x}] + g(q) \] Positive definite

\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
\[ \dot{q} \]
Control in the task space

The controller makes the target configuration $x_d$.

Define:

$$\vec{x} = x_d - x$$

Consider the following change of variables

$$\begin{pmatrix} q \\ \dot{q} \end{pmatrix} \rightarrow \begin{pmatrix} x \\ \dot{x} \end{pmatrix} \rightarrow \begin{pmatrix} \vec{x} \\ \dot{\vec{x}} \end{pmatrix}$$

The stability of $x_d$ can be analyzed by analyzing the stability of the origin in the new coordinates.

Control in the task space

Consider the following candidate Lyapunov function

$$V(\vec{x}, \dot{\vec{q}}) = \frac{1}{2} \dot{\vec{q}}^T M(q) \dot{\vec{q}} + \frac{1}{2} \vec{x}^T K_\text{p} \vec{x}$$

The function is composed by two terms

$$\frac{1}{2} \dot{\vec{q}}^T M(q) \dot{\vec{q}} \quad \text{The kinetic energy of the system}$$

$$\frac{1}{2} \vec{x}^T K_\text{p} \vec{x} \quad \text{The elastic potential energy imposed by the controller.}$$
Control in the task space

Since $M(q)$ and $K_P$ are positive definite for every $q$, $V$ is positive definite. Taking into account that:

$$\ddot{x} = -\dot{x} = -J(q)\dot{q}$$

we have that:

$$\dot{V}(\bar{x}, \dot{q}) = q^T M(q)\ddot{q} + \frac{1}{2}q^T \dot{M}(q)\dot{q} - \dot{q}^T J^T(q)K_P\bar{x}$$

but:

$$M(q)\ddot{q} = J^T(q)K_P\bar{x} - C(q, \dot{q})\dot{q} - D\dot{q} - J^T(q)K_DJ(q)\dot{q}$$

Thus:

$$V(\bar{x}, \dot{q}) = q^T(J^T(q)K_P\bar{x} - C(q, \dot{q})\dot{q} - D\dot{q} - J^T(q)K_DJ(q)\dot{q}) + \frac{1}{2}q^T \dot{M}(q)\dot{q} - \dot{q}^T J^T(q)K_P\bar{x}$$

from which

$$\dot{V}(\bar{x}, \dot{q}) = \frac{1}{2}q^T (M(q) - 2C(q, \dot{q})\dot{q} - D + J^T(q)K_DJ(q))\dot{q}$$

but, in virtue of the properties of the Euler-Lagrange model:

$$\dot{q}^T (M(q) - 2C(q, \dot{q}))\dot{q} = 0$$

Thus

$$\dot{V}(\bar{x}, \dot{q}) = -\dot{q}^T (D + J^T(q)K_DJ(q))\dot{q}$$

By properly choosing the gain matrix $K_D$, it is possible to tune the velocity with which the robot tends to the target configuration $x_d$. 

---

Control of Industrial Robot

Pag. 61
Control in the task space

The computed torque control can be reformulated in the task space:

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

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

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

\[ \dot{q} = y \]

where \( y \) represents a part of the control input that has still to be determined. The vector \( y \) has to be determined in such a way that the end-effector tracks the desired setpoint \( x_d \).

Control in the task space

Deriving the differential kinematic equation, we obtain:

\[ \ddot{x} = J(q)\ddot{q} + \dot{J}(q, \dot{q})\dot{q} \]

If \( J(q) \) is invertible, the following input can be chosen:

\[ y = J^{-1}(q)(\ddot{x}_d + K_D\dot{x} + K_Px - \dot{J}(q, \dot{q})\dot{q}) \]

which gives

\[ \ddot{x} + K_D\dot{x} + K_Px = 0 \]

where \( \ddot{x} = x_d - x \)

Thus, choosing \( K_D \) and \( K_P \) positive definite, the tracking error dynamics is asymptotically stable and, therefore, the tracking error tends to 0 with a dynamic that can be arbitrarily chosen by properly choosing \( K_P \) and \( K_D \).

Using the computed torque controller, very good trajectory tracking performance can be achieved.
Control in the task space

- Very good tracking performance can be achieved.

- There is a singularity problem since the control law depends on the inverse of the Jacobian. It is necessary to plan trajectories and build the control action in such a way that the manipulator stays away from singularity configurations.
CONTROL OF INDUSTRIAL ROBOTS
Laurea Magistrale in Ingegneria Meccatronica

CONTROL OF INDUSTRIAL ROBOTS
CONTROL OF ROBOTIC ARMS

Ing. Cristian Secchi
Tel. 0522 522235
e-mail: cristian.secchi@unimore.it