Control of industrial robots Notes[1]

Roberto Bochet

December 4, 2021

This work is realized in LaTeX, you can find the source code on https://github.com/RobertoBochet/cir-notes

[1]the whole document is based on [1] and [2]

Bibliography

[1] Paolo Rocco. *Slides of the PoliMi course "Control of industirial robots"*. `https://rocco.faculty.polimi.it/cir/index.html`. 2020.

[2] Bruno Siciliano et al. *Robotica. Modellistica, pianificazione e controllo*. it. 3rd ed. College. McGraw-Hill Education, 2008. ISBN: 978-88-386-6322-2.

Contents

Chapter 1

Kinematic redundancy

We let $\boldsymbol{x} = k(\boldsymbol{q})$ where $\boldsymbol{x}$ is the pose of the end effector in the task space, $\boldsymbol{q}$ the joint configuration. The kinematic function $k(\cdot)$ provides the transformation from the joint space $\mathbb{R}^n$ to the task space $\mathbb{R}^m$.

A robot is kinematically redundant if $n > m$.

n.b. task space might be a subset of the operational space, so the dimension of the second one might be less than the first, that is why the redundancy is a relative concept

1.1 Inverse kinematics

$$\boldsymbol{q} = k^{-1}(\boldsymbol{x})$$

When a robot is kinematically redundant the inversion of the kinematics gives infinite solutions and there are internal motion that not affect the pose of the end effector.

To solve this issue the problem is usually addressed in the speed domain.

$$\dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{1.1}$$

Where $\boldsymbol{J}(\boldsymbol{q})$ is the Jacobian of the robot associated to the task space.

1.1.1 Jacobian

For a redundant robot the Jacobian are a rectangular matrix with the shape of $(m \times n)$ where $n > m$ as we saw before.

$$\dim(\text{im}(\boldsymbol{J})) = m \quad \dim(\ker(\boldsymbol{J})) = n - m$$

The null space exists only for a redundant robot.

So we can compose a movement as

$$\dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}^* + \boldsymbol{P}\dot{\boldsymbol{q}}_0 \quad \text{where} \quad \text{im}(\boldsymbol{P}) \equiv \ker(\boldsymbol{J})$$

with $P\dot{q}_0$ that does not contribute to the movement of the end effector, in fact if we multiply the equation to J from left we get

$$\boldsymbol{J}\dot{\boldsymbol{q}} = \boldsymbol{J}\dot{\boldsymbol{q}}^* + \boldsymbol{J}P\dot{\boldsymbol{q}}_0 = \boldsymbol{J}\dot{\boldsymbol{q}}^* = \dot{\boldsymbol{x}}$$

where $JP\dot{\boldsymbol{q}}_0 = 0$ for each $\dot{\boldsymbol{q}}_0$.

1.1.2 Finding a generic solution

We can approach to the problem like a optimization one, as first thing we define a cost function in the joint space

$$g(\dot{\boldsymbol{q}}) = \frac{1}{2}\dot{\boldsymbol{q}}^T\boldsymbol{W}\dot{\boldsymbol{q}}$$

$\boldsymbol{W}$ is a symmetrical weight matrix $(n \times n)$ and $\boldsymbol{W} > 0$

we want to find the optimal solution under the constraint $\dot{\boldsymbol{x}} = \boldsymbol{W}(q)\dot{\boldsymbol{q}}$ so we can use the Lagrange multiplier

$$\mathcal{L}(\dot{\boldsymbol{q}}, \boldsymbol{\lambda}) = \frac{1}{2}\dot{\boldsymbol{q}}^T\boldsymbol{W}\dot{\boldsymbol{q}} + \lambda^T(\dot{\boldsymbol{x}} - \boldsymbol{J}\dot{\boldsymbol{q}})$$

$$\left(\frac{\partial \mathcal{L}}{\partial \dot{\boldsymbol{q}}}\right)^\mathsf{T} = 0 \quad \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}}\right)^\mathsf{T} = 0$$

from these we get

$$\boldsymbol{W}\dot{\boldsymbol{q}} - \boldsymbol{J}^T\boldsymbol{\lambda} = 0 \qquad \dot{\boldsymbol{x}} - \boldsymbol{J}\dot{\boldsymbol{q}} = 0$$

to combine the two equation we can find $\boldsymbol{\lambda}$

$$\dot{\boldsymbol{x}} = \boldsymbol{J}\boldsymbol{W}^{-1}\boldsymbol{J}^\mathsf{T}\boldsymbol{\lambda} \implies \boldsymbol{\lambda} = (\boldsymbol{J}\boldsymbol{W}^{-1}\boldsymbol{J}^\mathsf{T})^{-1}\dot{\boldsymbol{x}}$$

and with a recombination with the first equation we get the solution

$$\dot{\boldsymbol{q}} = \boldsymbol{W}^{-1}\boldsymbol{J}^{\mathsf{T}}(\boldsymbol{J}\boldsymbol{W}^{-1}\boldsymbol{J}^{\mathsf{T}})^{-1}\dot{\boldsymbol{x}}$$
$$\text{special case} \quad \boldsymbol{W} = \boldsymbol{I} \implies \dot{\boldsymbol{q}} = \boldsymbol{J}^{\dagger}\dot{\boldsymbol{x}}$$

The matrix $\boldsymbol{J}^{\dagger} = \boldsymbol{J}^{\mathsf{T}}(\boldsymbol{J}\boldsymbol{J}^{\mathsf{T}})^{-1}$ is called **right pseudo inverse** matrix. With dualism, we can define the matrix $\boldsymbol{J}^{\dagger}_{W} = \boldsymbol{W}^{-1}\boldsymbol{J}^{\mathsf{T}}(\boldsymbol{J}\boldsymbol{W}^{-1}\boldsymbol{J}^{\mathsf{T}})^{-1}$ calling **weighted pseudo inverse** matrix.

1.1.3 Finding a solution exploiting redundant

As we saw before if $\dot{\boldsymbol{q}}^*$ is a valid solution to inverse problem also $\dot{\boldsymbol{q}}^* + \boldsymbol{P}\dot{\boldsymbol{q}}_0$ is it. So, we can modify the general approach saw above to find a solution that exploiting the redundant. We consider a new cost function

$$g'(\dot{\boldsymbol{q}}) = \frac{1}{2}(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_0)^T(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_0)$$

To find a solution under the constraint 1.1 close to $\dot{\boldsymbol{q}}_0$.

$$\mathcal{L}'(\dot{\boldsymbol{q}}, \boldsymbol{\lambda}) = \frac{1}{2}(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_0)^T(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_0) + \boldsymbol{\lambda}^T(\dot{\boldsymbol{x}} - \boldsymbol{J}\dot{\boldsymbol{q}})$$

From $\frac{\partial \mathcal{L}'}{\partial \dot{\boldsymbol{q}}} = 0$ and the constraint 1.1 we can get $\boldsymbol{\lambda}$

$$\boldsymbol{\lambda} = (\boldsymbol{J}\boldsymbol{J}^{\mathsf{T}})^{-1}(\dot{\boldsymbol{x}} - \boldsymbol{J}\dot{\boldsymbol{q}}_0)$$

then we get the solution

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{\dagger}\dot{\boldsymbol{x}} + (\boldsymbol{I} - \boldsymbol{J}^{\dagger}\boldsymbol{J})\dot{\boldsymbol{q}}_0 \tag{1.2}$$

So we found $(\boldsymbol{I} - \boldsymbol{J}^{\dagger}\boldsymbol{J})$ a solution for the matrix $\boldsymbol{P}$.

1.1.3.1 Choosing $\dot{\boldsymbol{q}}_0$

Now we can choose $\dot{\boldsymbol{q}}_0$ to improve the behaviour of the robot exploiting its redundancy. A typical choice of $\dot{\boldsymbol{q}}_0$ called **projected gradient** has the shape

$$\dot{\boldsymbol{q}}_0 = k_0\left(\frac{\partial w(\boldsymbol{q})}{\partial \boldsymbol{q}}\right)^{\mathsf{T}}$$

Choosing $w(\boldsymbol{q})$ as a cost function to maximize to increase a specific performance:

- **Manipulability** measure to maximize the distance from singularities

$$w(\boldsymbol{q}) = \sqrt{\det(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}^{\mathsf{T}}(\boldsymbol{q}))}$$

- Distance form **joint limits**

$$w(\boldsymbol{q}) = \frac{1}{2n}\sum_{i=1}^{n}\left(\frac{q_i - \bar{q}_i}{q_{iM} - \bar{q}_{im}}\right)^2$$

- Distance from the **closet obstacle**

$$w(\boldsymbol{q}) = \min_{\boldsymbol{x},\boldsymbol{o}} \|\boldsymbol{x}(\boldsymbol{q}) - \boldsymbol{o}\|$$

$k_0 > 0$ can be arbitrary chosen.

1.1.4 Possible issues with redundant robots

Mainly two possible issues

- A close trajectory in the task space can be mapped as a open trajectory in the joint space
- Trajectories with the same initial and final task positions may end with different joint configurations

A possible solution to these problems is the method called **repeatable** or **cyclic**

The source of the problems above is the non repeatability of the kinematic inversion methods

A way to solve the kinematic inversion problem is based on the **augmented Jacobian**, it consists in add an auxiliary task in order to completely bound the kinematic inversion problem; where the main task is expressed by $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{q})$ and the auxiliary task $\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{q})$. Now we can introduce an enlarged version of the Jacobian that taking in account the auxiliary task.

$$\dot{\boldsymbol{x}}_{aug} = \begin{bmatrix}\dot{\boldsymbol{x}}\\ \dot{\boldsymbol{y}}\end{bmatrix} = \begin{bmatrix}\boldsymbol{J}(\boldsymbol{q})\\ \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{q}}\end{bmatrix}\dot{\boldsymbol{q}} = \boldsymbol{J}_{aug}(\boldsymbol{q})\dot{\boldsymbol{q}}$$

where $\boldsymbol{J}_{aug}$ is the **augmented Jacobian**. In this way the inverse kinematics problem is reduced to

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_{aug}^{-1}(\boldsymbol{q})\dot{\boldsymbol{x}}_{aug}$$

if now consider $\boldsymbol{y}$ constant we can write

$$\dot{\boldsymbol{q}} = \begin{bmatrix} \boldsymbol{J}(\boldsymbol{q}) \\ \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{q}} \end{bmatrix}^{-1} \begin{bmatrix} \dot{\boldsymbol{x}} \\ \boldsymbol{0} \end{bmatrix}$$

so we introduced additional holonomic constraints to the system, so the **augmented Jacobian** make the kinematic inversion **repeatable**.

The holonomic constraints can be selected as for the conditions for constrained optimality of a given objective function.

*n.b. the inversion of the **augmented Jacobian** may introduce new singularities called **algorithmic singularities***

1.2 Kinematic control

Exploiting the Equation 1.2 we can design a control law in the kinematic space (the robot dynamics is ignored) for a redundant robot. We can design a control law for the input $\dot{\boldsymbol{x}}$ of the inverted kinematic model, a simple control law could be

$$\dot{\boldsymbol{x}} = \dot{\bar{\boldsymbol{x}}} + \boldsymbol{K}(\bar{\boldsymbol{x}} - \hat{\boldsymbol{x}})$$

where $\hat{\boldsymbol{x}} = \boldsymbol{k}(\boldsymbol{q})$. The overall system is

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{\dagger}\left(\dot{\bar{\boldsymbol{x}}} + \boldsymbol{K}(\bar{\boldsymbol{x}} - \hat{\boldsymbol{x}})\right) + \left(\boldsymbol{I} - \boldsymbol{J}^{\dagger}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})\right)\dot{\boldsymbol{q}}_0$$

the implemented scheme is shown in Figure 1.1.

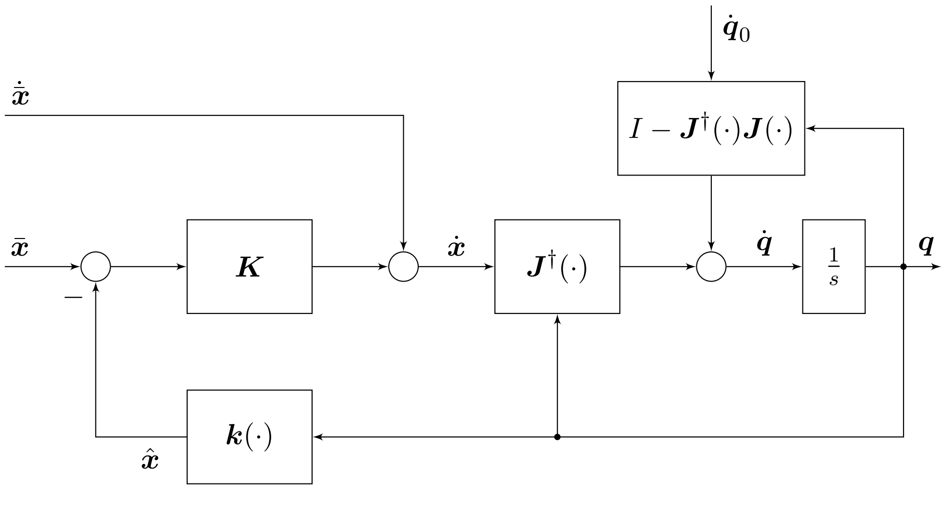

Figure 1.1: Kinematic control for a redundant robot

Chapter 2

Robot dynamics

To develop advance system of control and trajectory planning we need a dynamics representation of the robot, the bound between torques and the robot's motions.

2.1 Model exploiting Euler-Lagrange equation

We can exploit the Euler-Lagrange equation to derive a model fo the robot's dynamics.

$$\left(\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\boldsymbol{q}}}\right)^{\mathsf{T}} - \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{q}}\right)^{\mathsf{T}} = \boldsymbol{\xi} \qquad \mathcal{L} = \mathcal{T} - \mathcal{U}$$

where $\mathcal{T}$ is the kinetics energy, $\mathcal{U}$ is the potential energy and $\boldsymbol{\xi}$ the generalized forces associated to the vector $\boldsymbol{q}$.

2.1.1 Kinetics energy

$$\mathcal{T} = \sum_{i=1}^{n}(\mathcal{T}_{l_i} + \mathcal{T}_{m_i})$$

This is the sum of the kinetics contributions of all the robot's links, in particular $\mathcal{T}_{l_i}$ is the contribution from the i-th link and $\mathcal{T}_{m_i}$ from the i-th joint motor.

The contribution for the i-th link are defined as

$$\mathcal{T}_{l_i} = \frac{1}{2}\int_{V_i} \dot{\boldsymbol{p}}_i^{*\mathsf{T}} \dot{\boldsymbol{p}}_i^{*} \rho dV \tag{2.1}$$

where $\dot{\boldsymbol{p}}_i^{*}$ is the speed of a generic point of the link, and it can be expressed as

$$\dot{\boldsymbol{p}}_i^* = \dot{\boldsymbol{p}}_{l_i} + \boldsymbol{\omega}_i \times \boldsymbol{r}_i = \dot{\boldsymbol{p}}_{l_i} + \boldsymbol{S}(\boldsymbol{\omega}_i)\boldsymbol{r}_i \tag{2.2}$$

$\boldsymbol{p}_{l_i}$ is the position of the center of mass of the i-th link and $\boldsymbol{r}_i$ the vector that define the position of generic point respect the link's center of mass.

$$\boldsymbol{p}_{l_i} = \frac{1}{m_i}\int_{V_i} \boldsymbol{p}_i^* \rho dV \tag{2.3}$$

Combining Equation 2.1 and Equation 2.2 we get

$$\begin{aligned}
\mathcal{T}_{l_i} &= \frac{1}{2}\int_{V_i} \left(\dot{\boldsymbol{p}}_{l_i} + \boldsymbol{S}(\boldsymbol{\omega}_i)\boldsymbol{r}_i\right)^\mathsf{T} \left(\dot{\boldsymbol{p}}_{l_i} + \boldsymbol{S}(\boldsymbol{\omega}_i)\boldsymbol{r}_i\right) \rho dV \\
&= \frac{1}{2}\int_{V_i} \left(\dot{\boldsymbol{p}}_{l_i}^\mathsf{T}\dot{\boldsymbol{p}}_{l_i} + \dot{\boldsymbol{p}}_{l_i}^\mathsf{T}\boldsymbol{S}(\boldsymbol{\omega}_i)\boldsymbol{r}_i + \boldsymbol{r}_i^\mathsf{T}\boldsymbol{S}^\mathsf{T}(\boldsymbol{\omega}_i)\dot{\boldsymbol{p}}_{l_i} + \boldsymbol{r}_i^\mathsf{T}\boldsymbol{S}^\mathsf{T}(\boldsymbol{\omega}_i)\boldsymbol{S}(\boldsymbol{\omega}_i)\boldsymbol{r}_i\right) \rho dV \\
&= \frac{1}{2}\int_{V_i} \dot{\boldsymbol{p}}_{l_i}^\mathsf{T}\dot{\boldsymbol{p}}_{l_i}\rho dV + \int_{V_i} \dot{\boldsymbol{p}}_{l_i}^\mathsf{T}\boldsymbol{S}(\boldsymbol{\omega}_i)\boldsymbol{r}_i\rho dV + \frac{1}{2}\int_{V_i} \boldsymbol{r}_i^\mathsf{T}\boldsymbol{S}^\mathsf{T}(\boldsymbol{\omega}_i)\boldsymbol{S}(\boldsymbol{\omega}_i)\boldsymbol{r}_i\rho dV
\end{aligned}$$

We can identify in this equation three contributes:

Translation

$$\frac{1}{2}\int_{V_i} \dot{\boldsymbol{p}}_{l_i}^\mathsf{T}\dot{\boldsymbol{p}}_{l_i}\rho dV = \frac{1}{2}\dot{\boldsymbol{p}}_{l_i}^\mathsf{T}\dot{\boldsymbol{p}}_{l_i}\int_{V_i}\rho dV = \frac{1}{2}m_i\dot{\boldsymbol{p}}_{l_i}^\mathsf{T}\dot{\boldsymbol{p}}_{l_i}$$

Mutual

$$\int_{V_i} \dot{\boldsymbol{p}}_{l_i}^\mathsf{T}\boldsymbol{S}(\boldsymbol{\omega}_i)\boldsymbol{r}_i\rho dV = \dot{\boldsymbol{p}}_{l_i}^\mathsf{T}\boldsymbol{S}(\boldsymbol{\omega}_i)\int_{V_i}\boldsymbol{r}_i\rho dV = \dot{\boldsymbol{p}}_{l_i}^\mathsf{T}\boldsymbol{S}(\boldsymbol{\omega}_i)\int_{V_i}(\boldsymbol{p}_i^* - \boldsymbol{p}_{l_i})\rho dV = 0$$

Because from Equation 2.3 we can notice that

$$\int_{V_i}\boldsymbol{p}_i^*\rho dV = \boldsymbol{p}_{l_i}\int_{V_i}\rho dV \implies \int_{V_i}(\boldsymbol{p}_i^* - \boldsymbol{p}_{l_i})\rho dV = 0$$

Rotational

$$\frac{1}{2}\int_{V_i} \boldsymbol{r}_i^\mathsf{T}\boldsymbol{S}^\mathsf{T}(\boldsymbol{\omega}_i)\boldsymbol{S}(\boldsymbol{\omega}_i)\boldsymbol{r}_i\rho dV = \frac{1}{2}\boldsymbol{\omega}_i^\mathsf{T}\left(\int_{V_i}\boldsymbol{S}^\mathsf{T}(\boldsymbol{r}_i)\boldsymbol{S}(\boldsymbol{r}_i)\rho dV\right)\boldsymbol{\omega}_i$$

because of $\boldsymbol{S}(\boldsymbol{\omega}_i)\boldsymbol{r}_i = -\boldsymbol{S}(\boldsymbol{r}_i)\boldsymbol{\omega}_i$.

Let us define

$$\boldsymbol{I}_{l_i} = \int_{V_i} \boldsymbol{S}^\mathsf{T}(\boldsymbol{r}_i)\boldsymbol{S}(\boldsymbol{r}_i)\rho dV$$

we call it **inertia tensor** referred to the center of mass of the i-th link expressed in the base frame.

The **inertia tensor** is a symmetrical matrix 3×3 and its components are expressed as

$$\boldsymbol{I}_{l_i} = \begin{bmatrix} \int(r_{iy}^2 + r_{iz}^2)\rho dV & -\int r_{ix}r_{iy}\rho dV & -\int r_{ix}r_{iz}\rho dV \\ * & \int(r_{ix}^2 + r_{iz}^2)\rho dV & -\int r_{iy}r_{iz}\rho dV \\ * & * & \int(r_{ix}^2 + r_{iy}^2)\rho dV \end{bmatrix}$$

We have to note that the inertia tensor expressed in the base frame is dependent by robot configuration, so it would be better if we can express it in the joint frame. Exploiting the transformation of the rotation speed $\boldsymbol{\omega}$ remembering that $\boldsymbol{\omega}_i^i = \boldsymbol{R}_i^\mathsf{T}\boldsymbol{\omega}_i$ we can write an equivalent tensor expressed in the joint frame as $\boldsymbol{I}_i = \boldsymbol{R}_i\boldsymbol{I}_i^i\boldsymbol{R}_i^\mathsf{T}$.

So, we can write the rotation contributes of i-th joint as

$$\frac{1}{2}\int_{V_i} \boldsymbol{r}_i^\mathsf{T}\boldsymbol{S}^\mathsf{T}(\boldsymbol{\omega}_i)\boldsymbol{S}(\boldsymbol{\omega}_i)\boldsymbol{r}_i\rho dV = \frac{1}{2}\boldsymbol{\omega}_i^\mathsf{T}\boldsymbol{I}_i\boldsymbol{\omega}_i = \frac{1}{2}\boldsymbol{\omega}_i^\mathsf{T}\boldsymbol{R}_i\boldsymbol{I}_i^i\boldsymbol{R}_i^\mathsf{T}\boldsymbol{\omega}_i$$

Now we can finally write the contribution of the i-th joint to the kinematics energy as

$$\mathcal{T}_{l_i} = \frac{1}{2}m_i\dot{\boldsymbol{p}}_{l_i}^\mathsf{T}\dot{\boldsymbol{p}}_{l_i} + \frac{1}{2}\boldsymbol{\omega}_i^\mathsf{T}\boldsymbol{R}_i\boldsymbol{I}_i^i\boldsymbol{R}_i^\mathsf{T}\boldsymbol{\omega}_i$$

The velocity can easily write as

$$\begin{bmatrix}\dot{\boldsymbol{p}}_{l_i} \\ \boldsymbol{\omega}_i\end{bmatrix} = \begin{bmatrix}\boldsymbol{J}_P^{(l_i)} \\ \boldsymbol{J}_O^{(l_i)}\end{bmatrix}\dot{\boldsymbol{q}} = \boldsymbol{J}^{(l_i)}\dot{\boldsymbol{q}}$$

n.b. $\boldsymbol{J}^{(l_i)}$ *for a link* i *include only the contribution of the joints* $1, \ldots, i$*, so the column* $i+1, \ldots, n$ *are equal to* $\mathbf{0}$

the single non-null column of $\boldsymbol{J}^{(l_i)}$ is defined as

$$\begin{bmatrix}\boldsymbol{j}_{P_j}^{(l_i)} \\ \boldsymbol{j}_{O_j}^{(l_i)}\end{bmatrix} = \begin{cases} \begin{bmatrix}\boldsymbol{z}_{j-1} \\ \mathbf{0}\end{bmatrix} & \text{for prismatic joint} \\ \\ \begin{bmatrix}\boldsymbol{z}_{j-1} \times (\boldsymbol{p}_{l_i} - \boldsymbol{p}_{j-1}) \\ \boldsymbol{z}_{j-1}\end{bmatrix} & \text{for rotational joint} \end{cases}$$

$$\mathcal{T}_{l_i} = \frac{1}{2} m_i \dot{\boldsymbol{q}}^\top \boldsymbol{J}_P^{(l_i)\top} \boldsymbol{J}_P^{(l_i)} \dot{\boldsymbol{q}} + \frac{1}{2} \dot{\boldsymbol{q}}^\top \boldsymbol{J}_O^{(l_i)\top} \boldsymbol{R}_i \boldsymbol{I}_i^i \boldsymbol{R}_i^\top \boldsymbol{J}_O^{(l_i)} \dot{\boldsymbol{q}}$$

Now we can compute the sum for the kinetics energy (we consider $\mathcal{T}_{m_i} = 0$ for each motor for simplicity, but include motors contributions is an easy task)

$$\mathcal{T} = \sum_{i=1}^{n} \mathcal{T}_{l_i} = \frac{1}{2} \dot{\boldsymbol{q}}^\top \boldsymbol{B}(\boldsymbol{q}) \dot{\boldsymbol{q}} = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \dot{q}_i \dot{q}_j b_{ij}(\boldsymbol{q})$$

where $\boldsymbol{B}(\boldsymbol{q})$ is called **inertia matrix** and it is defined as

$$\boldsymbol{B}(\boldsymbol{q}) = \sum_{i=1}^{n} \left(m_i \boldsymbol{J}_P^{(l_i)\top} \boldsymbol{J}_P^{(l_i)} + \boldsymbol{J}_O^{(l_i)\top} \boldsymbol{R}_i \boldsymbol{I}_i^i \boldsymbol{R}_i^\top \boldsymbol{J}_O^{(l_i)} \right)$$

it is symmetrical, > 0 and it depends upon the robot configuration $\boldsymbol{q}$

2.1.2 Potential energy

Also for the potential energy we compute the total energy as the sum of the contribution of each link

$$\mathcal{U} = \sum_{i=1}^{n} \mathcal{U}_i$$

The potential energy for a link is given by the gravitational force in accordig to this formula

$$\mathcal{U}_i = - \int_{V_i} \boldsymbol{g}_0^\top \boldsymbol{p}_i^* \rho dV = -m_i \boldsymbol{g}_0^\top \boldsymbol{p}_i$$

where $\boldsymbol{g}_0$ is the gravity acceleration vector expressed in the base frame.

$$\mathcal{U} = - \sum_{i=1}^{n} m_i \boldsymbol{g}_0^\top \boldsymbol{p}_i$$

2.1.3 The Lagrangian

Now, we have $\mathcal{T}$ and $\mathcal{U}$, so we can finally compute the Lagrangian function and solve the Lagrangian equation

$$
\begin{aligned}
\mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}}) &= \mathcal{T}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \mathcal{U}(\boldsymbol{q}) \\
&= \frac{1}{2}\dot{\boldsymbol{q}}^{\mathsf{T}}\boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{q}} + \sum_{i=1}^{n} m_i \boldsymbol{g}_0^{\mathsf{T}} \boldsymbol{p}_i \\
&= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \dot{q}_i \dot{q}_j b_{ij}(\boldsymbol{q}) + \sum_{i=1}^{n} m_i \boldsymbol{g}_0^{\mathsf{T}} \boldsymbol{p}_i(\boldsymbol{q})
\end{aligned}
$$

Let us consider the Lagrangian equation referring to a generic joint

$$
\begin{gathered}
\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}_i} - \frac{\partial \mathcal{L}}{\partial q_i} = \xi_i \\
\frac{d}{dt}\frac{\partial \mathcal{T}(\boldsymbol{q}, \dot{\boldsymbol{q}})}{\partial \dot{q}_i} - \frac{\partial \mathcal{T}(\boldsymbol{q}, \dot{\boldsymbol{q}})}{\partial q_i} + \frac{\partial \mathcal{U}(\boldsymbol{q})}{\partial q_i} = \xi_i
\end{gathered}
\tag{2.4}
$$

Let us solve the three parts that compose the Lagrangian equation

$$
\begin{aligned}
\frac{d}{dt}\frac{\partial \mathcal{T}(\boldsymbol{q}, \dot{\boldsymbol{q}})}{\partial \dot{q}_i} &= \frac{d}{dt}\frac{\partial}{\partial \dot{q}_i}\left(\frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n} \dot{q}_j \dot{q}_k b_{jk}(\boldsymbol{q})\right) \\
&= \frac{d}{dt}\left(\sum_{j=1}^{n} \dot{q}_j b_{ij}(\boldsymbol{q})\right) \\
&= \sum_{j=1}^{n} \ddot{q}_j b_{ij}(\boldsymbol{q}) + \sum_{j=1}^{n} \dot{q}_j \frac{d b_{ij}(\boldsymbol{q})}{dt} \\
&= \sum_{j=1}^{n} \ddot{q}_j b_{ij}(\boldsymbol{q}) + \sum_{j=1}^{n} \dot{q}_j \sum_{k=1}^{n} \dot{q}_k \frac{\partial b_{ij}(\boldsymbol{q})}{\partial q_k}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial \mathcal{T}(\boldsymbol{q}, \dot{\boldsymbol{q}})}{\partial q_i} &= \frac{\partial}{\partial q_i}\left(\frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n} \dot{q}_j \dot{q}_k b_{jk}(\boldsymbol{q})\right) \\
&= \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n} \dot{q}_j \dot{q}_k \frac{\partial b_{jk}(\boldsymbol{q})}{\partial q_i}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial \mathcal{U}(\boldsymbol{q})}{\partial q_i} &= -\frac{\partial}{\partial q_i}\left(\sum_{j=1}^{n} m_j \boldsymbol{g}_0^{\mathsf{T}} \boldsymbol{p}_j\right) \\
&= -\sum_{j=1}^{n} m_j \boldsymbol{g}_0^{\mathsf{T}} \frac{\partial \boldsymbol{p}_j}{\partial q_i} \\
&= -\sum_{j=1}^{n} m_j \boldsymbol{g}_0^{\mathsf{T}} \boldsymbol{j}_{P_i}^{(l_j)}(\boldsymbol{q}) \\
&= g_i(\boldsymbol{q})
\end{aligned}
$$

So the Lagrange equation 2.4 became

$$
\sum_{j=1}^{n} \ddot{q}_j b_{ij}(\boldsymbol{q}) + \sum_{j=1}^{n} \dot{q}_j \sum_{k=1}^{n} \dot{q}_k \frac{\partial b_{ij}(\boldsymbol{q})}{\partial q_k} - \frac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n} \dot{q}_j \dot{q}_k \frac{\partial b_{jk}(\boldsymbol{q})}{\partial q_i} + g_i(\boldsymbol{q}) = \xi_i
$$

and if we rearrange the terms we get

$$
\sum_{j=1}^{n} \ddot{q}_j b_{ij}(\boldsymbol{q}) + \sum_{j=1}^{n} \sum_{k=1}^{n} \dot{q}_j \dot{q}_k \left(\frac{\partial b_{ij}(\boldsymbol{q})}{\partial q_k} - \frac{1}{2} \frac{\partial b_{jk}(\boldsymbol{q})}{\partial q_i}\right) + g_i(\boldsymbol{q}) = \xi_i
$$

and defined

$$
h_{ijk}(\boldsymbol{q}) = \frac{\partial b_{ij}(\boldsymbol{q})}{\partial q_k} - \frac{1}{2} \frac{\partial b_{jk}(\boldsymbol{q})}{\partial q_i}
$$

we get the result

$$
\sum_{j=1}^{n} \ddot{q}_j b_{ij}(\boldsymbol{q}) + \sum_{j=1}^{n} \sum_{k=1}^{n} \dot{q}_j \dot{q}_k h_{ijk}(\boldsymbol{q}) + g_i(\boldsymbol{q}) = \xi_i
$$

where we can recognize

b_{ii} : inertial moment as "seen" from the axis of the i-th joint
b_{ij} : effect of the acceleration of j-th joint on the i-th joint
$h_{ijj}\dot{q}_j^2$: centrifugal effect induced at i-th joint by the velocity of the j-th joint
$h_{ijk}\dot{q}_j\dot{q}_k$: Coriolis effect induced at i-th joint by the velocity of the joints j and k

Now, we can write the Lagrangian equations for all the joints in a matrix form as

$$
\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\xi} \tag{2.5}
$$

2.1.3.1 The matrix $\boldsymbol{C}$

The matrix $\boldsymbol{C}$ is not unique, and its elements must satisfy the equation

$$\sum_{j=1}^{n} c_{ij}\dot{q}_j = \sum_{j=1}^{n}\sum_{k=1}^{n} h_{ijk}\dot{q}_j\dot{q}_k$$

A possible definition of $\boldsymbol{C}$ can be found with this process

$$\begin{aligned}\sum_{j=1}^{n} c_{ij}\dot{q}_j &= \sum_{j=1}^{n}\sum_{k=1}^{n} h_{ijk}\dot{q}_j\dot{q}_k \\ &= \sum_{j=1}^{n}\sum_{k=1}^{n} \dot{q}_j\dot{q}_k\left(\frac{\partial b_{ij}}{\partial q_k} - \frac{1}{2}\frac{\partial b_{jk}}{\partial q_i}\right) \\ &= \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n} \dot{q}_j\dot{q}_k\left(\frac{\partial b_{ij}}{\partial q_k}\right) + \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n} \dot{q}_j\dot{q}_k\left(\frac{\partial b_{ik}}{\partial q_j} - \frac{\partial b_{jk}}{\partial q_i}\right)\end{aligned}$$

so the elements of $\boldsymbol{C}$ can be defined as

$$c_{ij} = \sum_{k=1}^{n} c_{ijk}\dot{q}_k \tag{2.6}$$

with

$$c_{ijk} = \frac{1}{2}\left(\frac{\partial b_{ij}}{\partial q_k} + \frac{\partial b_{ik}}{\partial q_j} - \frac{\partial b_{jk}}{\partial q_i}\right)$$

c_{ijk} are called **Christoffel symbols** of the first kind

2.1.4 The non conservative forces

The last thing we have to do is include the forces acting on the system expressed as $\boldsymbol{\xi}$ in the Equation 2.5.

We can identify tree types of forces:

- joint torques $\boldsymbol{\tau}$
- viscous friction torques $-\boldsymbol{F}_\nu\dot{\boldsymbol{q}}$
  with $\boldsymbol{F}_\nu$ a diagonal matrix of viscous friction coefficients
- static friction torques $-\boldsymbol{f}_s(\boldsymbol{q}, \dot{\boldsymbol{q}})$
  with $\boldsymbol{f}_s$ function that models the static friction at the joints

If we include all these three forces in the Equation 2.5 we get

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{F}_\nu\dot{\boldsymbol{q}} + \boldsymbol{f}_s(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} \tag{2.7}$$

2.1.5 Skew-symmetry of matrix $\dot{\boldsymbol{B}} - 2\boldsymbol{C}$

The derivation of the matrix $\boldsymbol{C}$ seen in the subsubsection 2.1.3.1 allows us to derive a singularity property of the Equation 2.7.

$$\boldsymbol{N}(\boldsymbol{q},\dot{\boldsymbol{q}}) = \dot{\boldsymbol{B}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$$

this matrix satisfies the following property

$$\boldsymbol{w}^\mathsf{T}\boldsymbol{N}(\boldsymbol{q},\dot{\boldsymbol{q}})\boldsymbol{w} = 0$$

To prove it, we take into account Equation 2.6

$$\begin{aligned} c_{ij} &= \frac{1}{2}\sum_{k=1}^{n}\left(\frac{\partial b_{ij}}{\partial q_k} + \frac{\partial b_{ik}}{\partial q_j} - \frac{\partial b_{jk}}{\partial q_i}\right)\dot{q}_k \\ &= \frac{1}{2}\sum_{k=1}^{n}\frac{\partial b_{ij}}{\partial q_k}\frac{dq_k}{dt} + \frac{1}{2}\sum_{k=1}^{n}\left(\frac{\partial b_{ik}}{\partial q_j} - \frac{\partial b_{jk}}{\partial q_i}\right)\dot{q}_k \\ &= \frac{1}{2}\dot{b}_{ij} + \frac{1}{2}\sum_{k=1}^{n}\left(\frac{\partial b_{ik}}{\partial q_j} - \frac{\partial b_{jk}}{\partial q_i}\right)\dot{q}_k \end{aligned}$$

So, we take the generic element

$$n_{ij} = \dot{b}_{ij} - 2c_{ij} = -\sum_{k=1}^{n}\left(\frac{\partial b_{ik}}{\partial q_j} - \frac{\partial b_{jk}}{\partial q_i}\right)\dot{q}_k$$

So, we can notice that $n_{ij} = -n_{ji}$, then the property is demonstrated.

As direct consequence we can state

$$\dot{\boldsymbol{q}}^\mathsf{T}\boldsymbol{N}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = 0 \tag{2.8}$$

for any choice of the matrix $\boldsymbol{C}$.

Let us try to find a physical interpretation to this property, let us consider the **principle of energy conservation**

$$\frac{1}{2}\frac{d}{dt}(\dot{\boldsymbol{q}}^\top \boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{q}}) = \dot{\boldsymbol{q}}^\top \left(\boldsymbol{\tau} - \boldsymbol{F}_\nu \dot{\boldsymbol{q}} - \boldsymbol{f}_s(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \boldsymbol{g}(\boldsymbol{q})\right) \tag{2.9}$$

we consider the left side member, and we try to solve it

$$\begin{aligned}
\frac{1}{2}\frac{d}{dt}(\dot{\boldsymbol{q}}^\top \boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{q}}) &= \frac{1}{2}\dot{\boldsymbol{q}}^\top \dot{\boldsymbol{B}}(\boldsymbol{q})\dot{\boldsymbol{q}} + \dot{\boldsymbol{q}}^\top \boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} \\
\text{using 2.7} \qquad &= \frac{1}{2}\dot{\boldsymbol{q}}^\top \dot{\boldsymbol{B}}(\boldsymbol{q})\dot{\boldsymbol{q}} + \dot{\boldsymbol{q}}^\top \left(\boldsymbol{\tau} - \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \boldsymbol{F}_\nu \dot{\boldsymbol{q}} - \boldsymbol{f}_s(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \boldsymbol{g}(\boldsymbol{q})\right) \\
&= \frac{1}{2}\dot{\boldsymbol{q}}^\top \left(\dot{\boldsymbol{B}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\right)\dot{\boldsymbol{q}} + \dot{\boldsymbol{q}}^\top \left(\boldsymbol{\tau} - \boldsymbol{F}_\nu \dot{\boldsymbol{q}} - \boldsymbol{f}_s(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \boldsymbol{g}(\boldsymbol{q})\right) \\
&= \frac{1}{2}\dot{\boldsymbol{q}}^\top \boldsymbol{N}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \dot{\boldsymbol{q}}^\top \left(\boldsymbol{\tau} - \boldsymbol{F}_\nu \dot{\boldsymbol{q}} - \boldsymbol{f}_s(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \boldsymbol{g}(\boldsymbol{q})\right)
\end{aligned}$$

We can notice that this equation coincides with the Equation 2.9 only if the Equation 2.8 is satisfied. In conclusion, we can state that Equation 2.8 is satisfied for each matrix $\boldsymbol{C}$ because it is a direct consequence of the **principle of energy conservation** of the system.

2.1.6 Summary model

So, we summarize how to define a dynamics robot model

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

let's try to define a procedure to write the equation's parameters

1. Compute $\boldsymbol{B}(\boldsymbol{q})$

   1.1. Compute the matrices $\boldsymbol{J}_P^{(l_i)}, \boldsymbol{J}_O^{(l_i)}$

   1.2. Compute $\boldsymbol{B}(\boldsymbol{q})$ as

$$\boldsymbol{B}(\boldsymbol{q}) = \sum_{i=1}^{n} \left(m_i \boldsymbol{J}_P^{(l_i)\top} \boldsymbol{J}_P^{(l_i)} + \boldsymbol{J}_O^{(l_i)\top} \boldsymbol{R}_i \boldsymbol{I}_i^i \boldsymbol{R}_i^\top \boldsymbol{J}_O^{(l_i)} \right)$$

2. Compute $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$

   2.1. Compute the **Christoffel symbols** c_{ijk} for all combinations of i, j, k in $\{1, \ldots, n\}^3$, (remembering $c_{ijk} = c_{ikj}$), as

$$c_{ijk} = \frac{1}{2}\left(\frac{\partial b_{ij}}{\partial q_k} + \frac{\partial b_{ik}}{\partial q_j} - \frac{\partial b_{jk}}{\partial q_i}\right)$$

2.2. Compute the $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ element by element as

$$c_{ij} = \sum_{k=1}^{n} c_{ijk}\dot{q}_k$$

3. (Optionally) As validation step you can check that $\dot{\boldsymbol{B}} - 2\boldsymbol{C}$ is antisymmetric

4. Compute $\boldsymbol{g}(\boldsymbol{q})$ elements with

$$g_i(\boldsymbol{q}) = -\sum_{j=1}^{n} m_j \boldsymbol{g}_0^{\mathsf{T}} \boldsymbol{j}_{P_i}^{(l_j)}(\boldsymbol{q})$$

2.1.7 Linearity in dynamic parameters

If we assume a simplified function for the static friction function

$$\boldsymbol{f}_s(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{F}_s \operatorname{sgn}(\dot{\boldsymbol{q}})$$

it is possible to demonstrate that the dynamic model of the manipulator is linear about a suitable set of dynamic parameters. In particular the Lagrangian can be written in the form

$$\mathcal{L} = \sum_{i=1}^{n} \left(\boldsymbol{\beta}_{\mathcal{T}_i}^{\mathsf{T}} - \boldsymbol{\beta}_{\mathcal{U}_i}^{\mathsf{T}}\right) \boldsymbol{\pi}_i$$

where $\boldsymbol{\pi}_i$ is a vector of constant parameters associated to i-th joint (mass, inertia, static friction, ...). If we solve the **Lagrangian equations** with the above Lagrangian we find that the linearity assumption is not lost, and we can write

$$\boldsymbol{\tau} = \boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\boldsymbol{\pi}$$

where $\boldsymbol{\pi}$ is the vector get by the concatenation of $\boldsymbol{\pi}_i$ and $\boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ is a matrix called **regression matrix**.

2.1.7.1 Parameters identification

It may be useful to develop a system to estimate the parameters vector $\boldsymbol{\pi}$ from experimental data due to the complexity of compute it via analytic method. Given a set of measures $\bar{\boldsymbol{\tau}}_1, \ldots, \bar{\boldsymbol{\tau}}_m$ and the corresponding matrix $\bar{\boldsymbol{Y}}_1, \ldots, \bar{\boldsymbol{Y}}_m$ (gotten from kinematics parameters and joints values $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}$, measured or estimated). If we define

$$\bar{\boldsymbol{\tau}} = \begin{bmatrix} \bar{\boldsymbol{\tau}}_1 \\ \vdots \\ \bar{\boldsymbol{\tau}}_m \end{bmatrix}, \qquad \bar{\boldsymbol{Y}} = \begin{bmatrix} \bar{\boldsymbol{Y}}_1 \\ \vdots \\ \bar{\boldsymbol{Y}}_m \end{bmatrix}$$

we can write

$$\bar{\boldsymbol{\tau}} = \bar{\boldsymbol{Y}}\boldsymbol{\pi}$$

this equation due to noise measures does not admit solution for $\boldsymbol{\pi}$ which it can be estimated with the **linear least squares method**[1] as

$$\hat{\boldsymbol{\pi}} = \left(\bar{\boldsymbol{Y}}^\mathsf{T}\bar{\boldsymbol{Y}}\right)^{-1}\bar{\boldsymbol{Y}}^\mathsf{T}\bar{\boldsymbol{\tau}} = \bar{\boldsymbol{Y}}^\dagger\bar{\boldsymbol{\tau}}$$

The trajectories used to generate $\bar{\boldsymbol{\tau}}$ and $\bar{\boldsymbol{Y}}$ must be enough rich (good conditioning of matrix $\bar{\boldsymbol{Y}}^\mathsf{T}\bar{\boldsymbol{Y}}$): they need to involve all components in the dynamic model.

2.2 Newton-Euler formulation

An alternative way to derive a dynamics model of a manipulator is exploiting **Newton-Euler equations**. As opposed to Lagrange formulation that is based on balance of energies, this kind of formulation is based on balances of forces and moments on each link.

We have to solve the equations in a recursive way, propagating the velocity and acceleration from the base to the end effector and the forces and moments in the opposite way.

For each joint we can find the following equations

$$\boldsymbol{f}_i - \boldsymbol{f}_{i+1} + m_i\boldsymbol{g}_0 = m_i\ddot{\boldsymbol{p}}_{C_i}$$
$$\boldsymbol{\mu}_i + \boldsymbol{f}_i \times \boldsymbol{r}_{i-1,C_i} - \boldsymbol{\mu}_{i+1} - \boldsymbol{f}_{i+1} \times \boldsymbol{r}_{i,C_i} = \boldsymbol{I}_i\dot{\omega}_i + \omega_i \times \boldsymbol{I}_i\omega_i$$

To include the generalized forces at joint we can write

We have to define generalized forces at joint, velocity and acceleration for prismatic and rotational joint

Prismatic

[1] https://en.wikipedia.org/wiki/Linear_least_squares

- generalized forces

$$\boldsymbol{\tau}_i = \boldsymbol{f}_i^{\mathsf{T}} \boldsymbol{z}_{i-1} + F_{v_i} \dot{d}_i + f_{s_i}$$

- velocities

$$\boldsymbol{\omega}_i = \boldsymbol{\omega}_{i-1}$$
$$\dot{\boldsymbol{p}}_i = \dot{\boldsymbol{p}}_{i-1} + \dot{d}_i \boldsymbol{z}_{i-1} + \boldsymbol{\omega}_i \times \boldsymbol{r}_{i-1,i}$$

- accelerations

$$\dot{\boldsymbol{\omega}}_i = \dot{\boldsymbol{\omega}}_{i-1}$$
$$\ddot{\boldsymbol{p}}_i = \ddot{\boldsymbol{p}}_{i-1} + \ddot{d}_i \boldsymbol{z}_{i-1} + 2\dot{d}_i \boldsymbol{\omega}_i \times \boldsymbol{z}_{i-1} + \dot{\boldsymbol{\omega}}_i \times \boldsymbol{r}_{i-1,i} + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \boldsymbol{r}_{i-1,i})$$

Rotational

- generalized forces

$$\boldsymbol{\tau}_i = \boldsymbol{\mu}_i^{\mathsf{T}} \boldsymbol{z}_{i-1} + F_{v_i} \dot{\theta}_i + f_{s_i}$$

- velocities

$$\boldsymbol{\omega}_i = \boldsymbol{\omega}_{i-1} + \dot{\theta}_i \boldsymbol{z}_{i-1}$$
$$\dot{\boldsymbol{p}}_i = \dot{\boldsymbol{p}}_{i-1} + \boldsymbol{\omega}_i \times \boldsymbol{r}_{i-1,i}$$

- accelerations

$$\dot{\boldsymbol{\omega}}_i = \dot{\boldsymbol{\omega}}_{i-1} + \ddot{\theta}_i \boldsymbol{z}_{i-1} + \dot{\theta}_i \boldsymbol{\omega}_{i-1} \times \boldsymbol{z}_{i-1}$$
$$\ddot{\boldsymbol{p}}_i = \ddot{\boldsymbol{p}}_{i-1} + \dot{\boldsymbol{\omega}}_i \times \boldsymbol{r}_{i-1,i} + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \boldsymbol{r}_{i-1,i})$$

The terms F_{v_i} and f_{s_i} that appear in the generalized forces are respectively the coefficient of viscous friction, and the function that characterize the static friction.

For both the joints types the acceleration of the center of mass is given by

$$\ddot{\boldsymbol{p}}_{C_i} = \ddot{\boldsymbol{p}}_i + \dot{\boldsymbol{\omega}}_i \times \boldsymbol{r}_{i,C_i} + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \boldsymbol{r}_{i,C_i})$$

2.2.1 The recursive algorithm

As we said above the **Newton-Euler formulation** is based on two kind of algorithm for the term propagation

Forward recursion from the base to the end effector, we propagate the velocities and accelerations. The initial condition is given by $\boldsymbol{\omega}_o, \ddot{\boldsymbol{p}}_0 - \boldsymbol{g}_0, \dot{\boldsymbol{\omega}}_0$, then the elements $\boldsymbol{\omega}_i, \dot{\boldsymbol{\omega}}_i, \ddot{\boldsymbol{p}}_i, \ddot{\boldsymbol{p}}_{C_i}$ are calculated recursively until the end effector.

Backward recursion from the end effector to the base, we propagate the forces and moments. The initial condition is given by $\boldsymbol{f}_{n+1} = \boldsymbol{f}_e, \boldsymbol{\mu}_{n+1} = \boldsymbol{\mu}_e$, the elements $\boldsymbol{f}_i$ and $\boldsymbol{\mu}_i$ are calculated recursively until the base as

$$\begin{aligned}
\boldsymbol{f}_i &= \boldsymbol{f}_{i+1} + m_i \ddot{\boldsymbol{p}}_{C_i} \\
\boldsymbol{\mu}_i &= -\boldsymbol{f}_i \times (\boldsymbol{r}_{i-1,i} + \boldsymbol{r}_{i,C_i}) + \boldsymbol{\mu}_{i+1} + \boldsymbol{f}_{i+1} \times \boldsymbol{r}_{i,C_i} + \boldsymbol{I}_i \dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times \boldsymbol{I}_i \boldsymbol{\omega}_i
\end{aligned}$$

2.2.1.1 Optimization

Note that the entities $\boldsymbol{r}_{i-1,i}$, $\boldsymbol{r}_{i,C_i}$ and $\boldsymbol{I}_i$ being referred in the base frame are functions of the robot's pose, so they are not constant, so to improve the efficient of the algorithm is a good idea refer them to the link frame so as to make them constant.

WARNING Missing examples

2.3 Euler-Lagrange vs Newton-Euler

Euler-Lagrange formulation

- It is a systematic and easy to understand.
- The returned formula is in a form in which the physical terms are separated (inertia matrix, Coriolis, centrifugal and gravitational). All these elements are useful to design a controller.
- It lends itself to the introduction into the model of more complex effects (like joint or link deformation).

Newton-Euler formulation

- It is a computational efficient recursive method.

2.3.1 Computation of direct dynamics

In the computation **direct dynamics** problem given the joint torques $\boldsymbol{\tau}(t)$ we have to calculate the acceleration of the joint $\ddot{\boldsymbol{q}}(t)$ and eventually compute via integration methods also the speeds $\boldsymbol{q}(t)$ and positions $\boldsymbol{q}(t)$. This is essential to simulate the model.

The **Euler-Lagrange formulation** can give an easy solution to this problem. We need to compute and invert this formula

$$\boldsymbol{B}(\boldsymbol{q})\ddot{q} + \boldsymbol{n}(\boldsymbol{q}, \dot{q}) = \boldsymbol{\tau}$$

where

$$\boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{F}_\nu \dot{\boldsymbol{q}} + \boldsymbol{f}_s(\boldsymbol{q}, \dot{\boldsymbol{q}})$$

so, inverting the function we get

$$\ddot{\boldsymbol{q}} = \boldsymbol{B}^{-1}(\boldsymbol{q})\left(\boldsymbol{\tau} - \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}})\right) \tag{2.10}$$

Also exploiting **Newton-Euler formulation** we can easily find the matrix $\boldsymbol{B}(\boldsymbol{q})$ and the vector $\boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ to solve the problem:

Given the implementation of the **Newton-Euler script** $\boldsymbol{\tau} = NE(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$, set $\ddot{\boldsymbol{q}} = 0$ and use the current values of $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$ the computed τ is exactly the searched vector $\boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}})$.

Then we set $\boldsymbol{g}_0 = 0$ into the script and $\dot{\boldsymbol{q}} = 0$ in order to eliminate gravitation, Coriolis, centrifugal and friction effects. With n interaction of the algorith we can compute the matrix $\boldsymbol{B}(\boldsymbol{q})$ column by column imposing $\ddot{q}_i = 1$ and $\ddot{q}_j = 0, i \neq j$.

So we can use always the Equation 2.10 with the computed parameters.

2.3.2 Computation of inverse dynamics

In the **inverse dynamics** problem the joint accelerations $\ddot{\boldsymbol{q}}(t)$, speeds $\dot{\boldsymbol{q}}(t)$ and position $\boldsymbol{q}(t)$ are given, and we have to compute the law of $\boldsymbol{\tau}(t)$ to produce that motion. This bound is useful in the **trajectory planning** and **model based control**.

An approach based on **Newton-Euler formulation** can be used to solve this problem efficiently.

Chapter 3

Motion planning

Chapter 4

Decentralized control

Once a desired robot trajectory is designed, the next step is to guarantee that the real trajectory as close as possible to it. Practically we have to design a control strategy in order to $\boldsymbol{q}_d(t) \approx \boldsymbol{q}(t)$.

The main way adopted in the control of industrial robot is the **independent joint control** approach; it is a decentralized control approach where each joint is controlled independently of each other.

WARNING Missing evaluation of control performance

4.1 Simplified dynamic model

Let us consider a simplified dynamics model of the robots

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} \tag{4.1}$$

In this model $\boldsymbol{\tau}, \boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}$ are referred to the joints, but in a real system we cannot act directly on the joints variables, instead we have to consider that we can act only on the motors that are bound to the joints via a transmission. So we introduce a dynamics model for a generic rigid transmission and joint motor

$$J_{mi}\ddot{q}_{m_i} + D_{m_i}\dot{q}_{m_i} = \tau_{m_i} - \tau_{lm_i}$$
$$\tau_{lm_i} = \tau_{l_i}/n_i$$

where

$$
\begin{aligned}
J_{m_i} &: \text{moment of inertial of the motor} \\
D_{m_i} &: \text{friction viscous coefficient of the motor} \\
\tau_{lm_i} &: \text{load torque on the axis of the motor} \\
\tau_{l_i} &: \text{load torque on the axis of the joint} \\
n_i &: \text{transmission reduction rate between motor and joint}
\end{aligned}
$$

to simplify we considered just the effect related to its spinning around its own axis and a rigid transmission.

Let us introduce the square matrix $\boldsymbol{N} = \text{diag}\{n_i\}$ so we can write

$$\boldsymbol{q}_m = \boldsymbol{N}\boldsymbol{q} \qquad \boldsymbol{\tau}_{lm} = \boldsymbol{N}^{-1}\boldsymbol{\tau}$$

Before we consider the entire model seen from the motor point of view, we observe that the matrix $\boldsymbol{B}(\boldsymbol{q})$ can be seen as the sum of two contributes, the first of average inertia and the other of the residual as function of the pose $\boldsymbol{q}$

$$\boldsymbol{B}(\boldsymbol{q}) = \bar{\boldsymbol{B}} + \boldsymbol{\Delta B}(\boldsymbol{q})$$

Now we can combine the models of motors/transmissions and robots

$$
\begin{aligned}
\boldsymbol{J}_m\ddot{\boldsymbol{q}}_m + \boldsymbol{D}_m\dot{\boldsymbol{q}}_m &= \boldsymbol{\tau}_m - \boldsymbol{\tau}_{lm} \\
\boldsymbol{J}_m\ddot{\boldsymbol{q}}_m + \boldsymbol{D}_m\dot{\boldsymbol{q}}_m + \boldsymbol{N}^{-1}\left(\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q})\right) &= \boldsymbol{\tau}_m
\end{aligned}
$$

replace the joints variables with motors ones

$$
\begin{aligned}
\boldsymbol{J}_m\ddot{\boldsymbol{q}}_m + \boldsymbol{D}_m\dot{\boldsymbol{q}}_m + \boldsymbol{N}^{-1}\boldsymbol{B}(\boldsymbol{q})\boldsymbol{N}^{-1}\ddot{\boldsymbol{q}}_m + \\
\boldsymbol{N}^{-1}\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\boldsymbol{N}^{-1}\dot{\boldsymbol{q}}_m + \boldsymbol{N}^{-1}\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}_m
\end{aligned}
$$

introduce the decomposition of $\boldsymbol{B}(\boldsymbol{q})$

$$
\begin{aligned}
\boldsymbol{J}_m\ddot{\boldsymbol{q}}_m + \boldsymbol{D}_m\dot{\boldsymbol{q}}_m + \boldsymbol{N}^{-1}\left(\bar{\boldsymbol{B}} + \boldsymbol{\Delta B}(\boldsymbol{q})\right)\boldsymbol{N}^{-1}\ddot{\boldsymbol{q}}_m + \\
\boldsymbol{N}^{-1}\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\boldsymbol{N}^{-1}\dot{\boldsymbol{q}}_m + \boldsymbol{N}^{-1}\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}_m \\
(\boldsymbol{J}_m + \boldsymbol{N}^{-1}\bar{\boldsymbol{B}}\boldsymbol{N}^{-1})\ddot{\boldsymbol{q}}_m + \boldsymbol{D}_m\dot{\boldsymbol{q}}_m + \boldsymbol{N}^{-1}\boldsymbol{\Delta B}(\boldsymbol{q})\boldsymbol{N}^{-1}\ddot{\boldsymbol{q}}_m + \\
\boldsymbol{N}^{-1}\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\boldsymbol{N}^{-1}\dot{\boldsymbol{q}}_m + \boldsymbol{N}^{-1}\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}_m
\end{aligned}
$$

introduce the vector $\boldsymbol{d}$ and the matrix $\bar{\boldsymbol{B}}_r = \boldsymbol{N}^{-1}\bar{\boldsymbol{B}}\boldsymbol{N}^{-1}$

$$\left(\boldsymbol{J}_m + \bar{\boldsymbol{B}}_r\right)\ddot{\boldsymbol{q}}_m + \boldsymbol{D}_m\dot{\boldsymbol{q}}_m + \boldsymbol{d} = \boldsymbol{\tau}_m$$

where

$$\boldsymbol{d} = \boldsymbol{N}^{-1}\boldsymbol{\Delta B}(\boldsymbol{q})\boldsymbol{N}^{-1}\ddot{\boldsymbol{q}}_m + \boldsymbol{N}^{-1}\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\boldsymbol{N}^{-1}\dot{\boldsymbol{q}}_m + \boldsymbol{N}^{-1}\boldsymbol{g}(\boldsymbol{q}) \tag{4.2}$$

We got a dynamics model where the motor torques $\boldsymbol{\tau}_m$ are the input, and the vector $\boldsymbol{d}$ is a non-linear function that we can treat as noise, you can see it in the Figure 4.1.

n.b. all the non-linearities are moved in the noise $\boldsymbol{d}$*.*

n.b. the bigger are the reduction ratios n_i*, the less relevant the noise* $\boldsymbol{d}$ *is.*

So, the decentralized control approach treat each joint as a SISO linear model subject to a noise. The control loop for each joint is independent of each others.

This method heavily relies on the assumption that the reduction ratios n_i values are large (generally true for industrial robotics manipulators), to can ignore the nonlinear contribution $\boldsymbol{d}$ in the controller design.

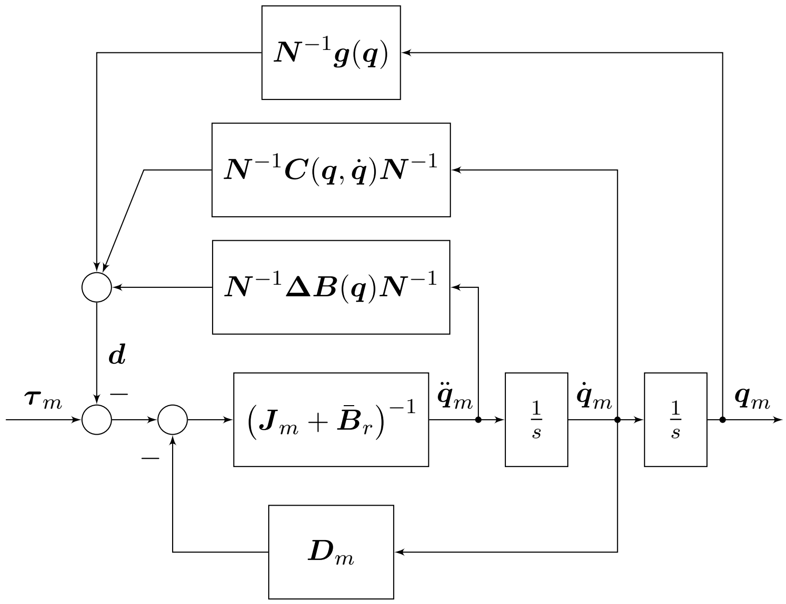

Figure 4.1: motor-transmission-joint dynamics model
n.b. the arrows for the dependencies of noise functions are been omitted

4.2 Electrical motor

Let's consider a simple DC motor

$$V(t) = Ri(t) + L\dot{i}(t) + E(t)$$
$$E(t) = K\dot{q}_m(t)$$
$$\tau_m(t) = Ki(t) = J_m\ddot{q}_m(t)$$

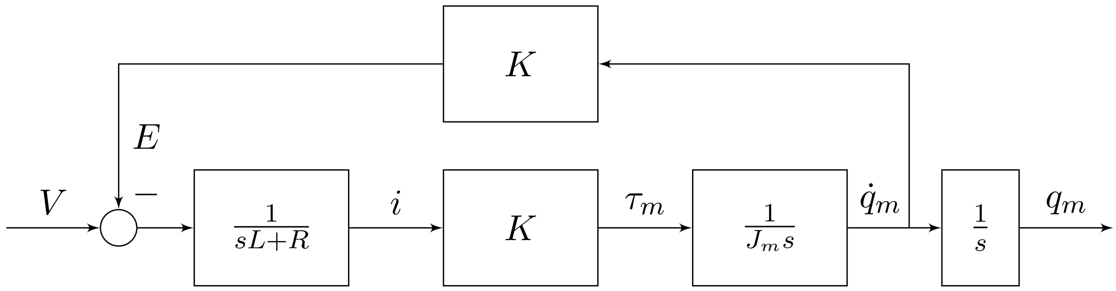

Figure 4.2: DC motor model

4.2.1 Control of the current

We can close a high frequency (thousands rad $/s$) loop for the current as it can be seen in Figure 4.3. E is considered to change as a slowly varying disturbance, that the controller can effectively reject. From the point of view of the mechanical system the current loop is so fast to consider the current change is instantaneous. So for the mechanical control we can consider to control directly the motor torque

$$\tau_m(t) = Ki(t) \approx K\bar{i}(t)$$

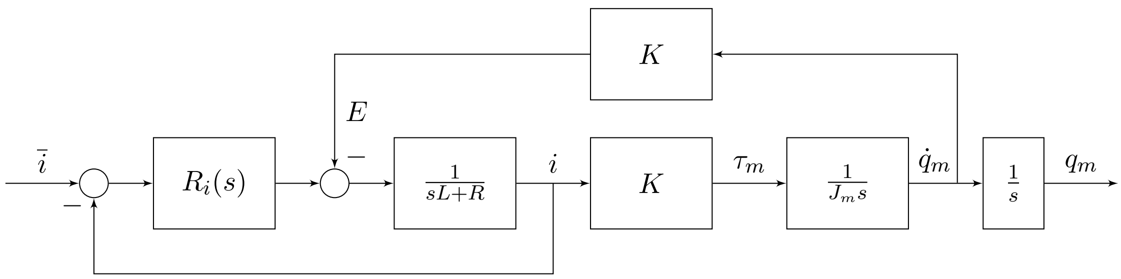

Figure 4.3: motor current control

4.3 Rigid transmission approximation

As we saw above, for a rigid transmission we have the following system

$$J_m \ddot{q}_m + d_m \dot{q}_m = \tau_m - \tau_{lm}$$
$$J_l \ddot{q}_l = n\tau_{lm} - \tau_l$$
$$q_m = nq_l$$

The first and the second equation can be merged exploiting the third equation

$$\left(J_m + \frac{J_l}{n^2}\right)\ddot{q}_m + d_m \dot{q}_m = \tau_m - \frac{\tau_l}{n}$$

The result model can be seen in the Figure 4.4

$$G_v(s) = \frac{1}{d_m + \left(J_m + \frac{J_l}{n^2}\right)s} = \frac{1/d_m}{1 + \frac{J_m + J_l/n^2}{d_m}s}$$

d_m is an uncertain small parameter, since d_m give a real stable pole we can consider the most conservative situation in which $d_m = 0$, so we get

$$G_v(s) = \frac{1}{\left(J_m + \frac{J_l}{n^2}\right)s}$$

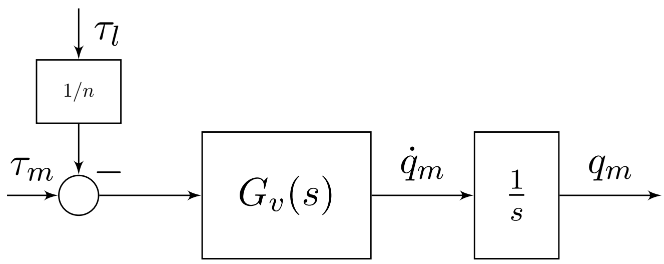

Figure 4.4: Mechanical system with rigid transmission

4.3.1 P/PI control

In order to control position of the motor we can close in cascade two loop (also to the current loop). The first with a **PI** regulator on the motor speed and the second with an **P** regulator on the motor position as you can see in Figure 4.5.

n.b. in Figure 4.5 the current loop is omitted because, as we said before, the current loop has a much higher bandwidth than the speed loop, so we can assume $\bar{\tau}_m = \tau_m$.

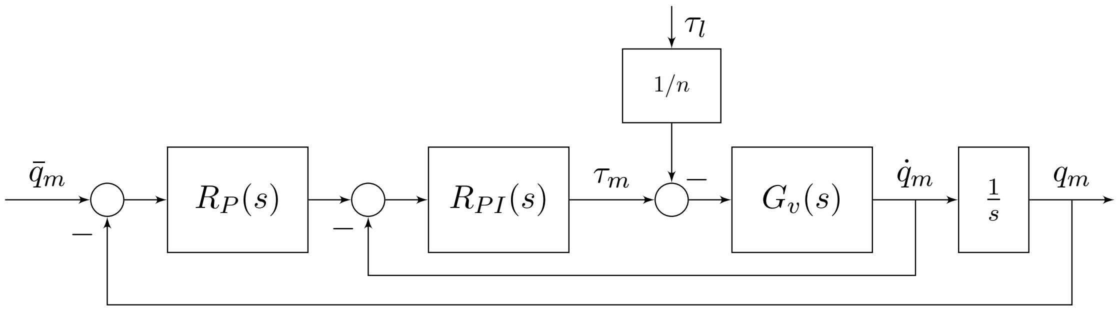

Figure 4.5: Control scheme P/PI with a rigid transmission

4.3.1.1 PI speed control design

As we stated the P/PI design is done with cascade approach, so as first step we can design the speed loop.

$$R_{PI}(s) = k_{pv}\frac{1+t_{iv}s}{s} \implies L_v(s) = \frac{k_{pv}}{\left(J_m + \frac{J_l}{n^2}\right)}\frac{1+t_{iv}s}{s^2}$$

We impose an ω_{cv} with k_{pv}

$$\omega_{cv} = \frac{k_{pv}}{\left(J_m + \frac{J_l}{n^2}\right)}$$

and to guarantee that the ω_{cv} desired coincide with the effective one we have to put the zero in low frequency respect it

$$t_{iv} \approx \frac{1}{(0.1 \div 0.3)\omega_{cv}}$$

With this regulator we get a close loop function for the speed loop

$$F_v(s) \approx \frac{1}{1+\frac{1}{\omega_{cv}}s}$$

4.3.1.2 P position control design

We can now design the $R_P(s)$ controller taking in account the speed closed loop designed above.

$$R_P(s) = k_{pp} \implies L_p(s) = k_{pp}\frac{1}{s\left(1+\frac{1}{\omega_{cv}}s\right)}$$

if we choose $\omega_{cp} \ll \omega_{cv}$ is enough to impose $k_{pp} = \omega_{cp}$

The overall closed loop on the position guarantee null static error and a bandwidth of ω_{cp}.

4.3.1.3 Speed feed-forward

A possible improvement of the **P/PI** scheme is the speed feed-forward scheme Figure 4.6. The block $k_{ff}s$ improves the speed of the response of the position control. The k_{ff} coefficient is chosen between 0 and 1.

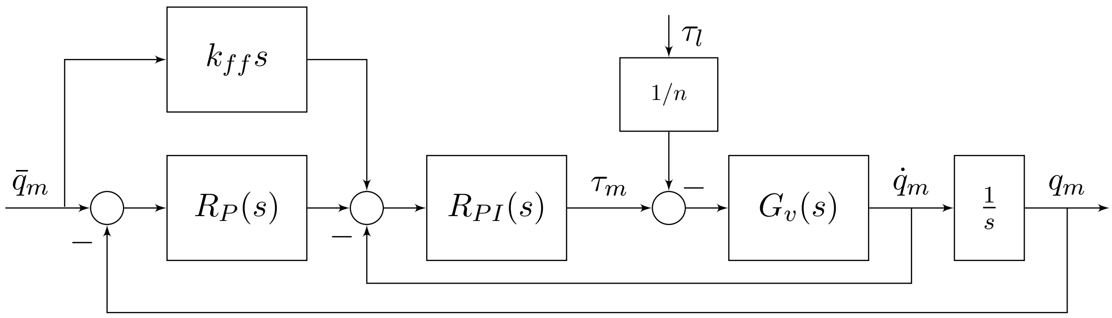

Figure 4.6: Control scheme P/PI with a rigid transmission and speed feed-forward

4.3.1.4 Position measure only

If only available the position measure but not the speed, it can be derived with a process of differentiation on the position. The control scheme became as Figure 4.7 that are equivalent to Figure 4.8 where all the controller block are substituted with a single **PID** on position loop.

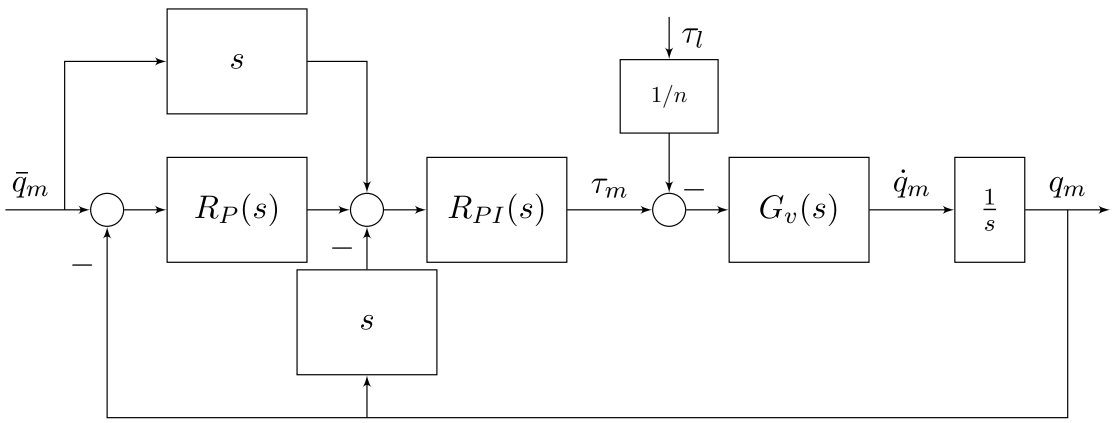

Figure 4.7: Control scheme with only position measurement

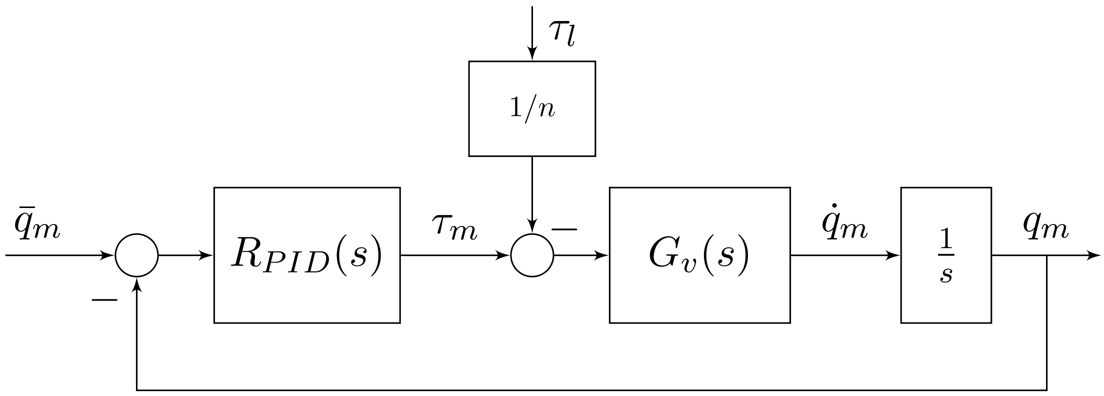

Figure 4.8: Control scheme PID with only position measurement

4.4 Elastic transmission

If we look only to the rigid transmission model we do not find significant limitations in the bandwidth of speed control, but from experimental data we can clearly find several limitations mainly bound to vibration, we have to improve the rigid model to take in account these limitations.

If we consider an elastic transmission we can write the following system

$$
\begin{gathered}
J_m\ddot{q}_m + d_m\dot{q}_m = \tau_m - \tau_{lm} \\
J_l\ddot{q}_l = n\tau_{lm} - \tau_l \\
\tau_{lm} = k_{el}(q_m - nq_l) + d_{el}(\dot{q}_m - n\dot{q}_l)
\end{gathered}
$$

The $G_v(s)$ now is composed of one real pole, two complex poles and two complex zeros; in particular we can find the parameter ω_z of the complex zeros as

$$\omega_z = \sqrt{\frac{k_{el}}{J_l/n^2}}$$

and a reasonable choice for the speed closed loop bandwidth is $\omega_{cv} \approx 0.7\omega_z$

Chapter 5

Centralized control

As opposed of the decentralized approach where each joint is controlled independently of each others, we can implement a control law based on **centralized approach** where the manipulator is controlled exploiting its overall model. This approach required obviously a model of the robot, but it generally guarantees better performance than the decentralized one.

n.b. as we saw in section 4.1 for a ***decentralized*** *control we stated that a high reduction ratios in transmissions between motors and joints is a fundamental requirement because this reduces the magnitude of the noise* $\boldsymbol{d}$*, if this decoupling effect is not guaranteed we must use the* ***centralized*** *approach.*

The **centralized control approach** allows us to develop several control schemes both in the joint space that in the operational one.

5.1 Control in joint space

In joint space control the design of the controller is done directly on the joints state.

5.1.1 Open loop

A first naive approach can be seen as an extension of a **decentralized controller** designed in the previous chapter.

The idea is to compensate the noise $\boldsymbol{d}$ adding to the decentralized scheme an open loop controller fed with the desired joints functions $\bar{\boldsymbol{q}}, \dot{\bar{\boldsymbol{q}}}, \ddot{\bar{\boldsymbol{q}}}$. Yuo can see the scheme in Figure 5.1.

The function of the feedforward controller can be designed from Equation 4.2 using as input the desired state

$$\hat{\boldsymbol{d}} = \boldsymbol{N}^{-1}\boldsymbol{\Delta B}(\bar{\boldsymbol{q}})\boldsymbol{N}^{-1}\ddot{\boldsymbol{q}}_m + \boldsymbol{N}^{-1}\boldsymbol{C}(\bar{\boldsymbol{q}}, \dot{\bar{\boldsymbol{q}}})\boldsymbol{N}^{-1}\dot{\boldsymbol{q}}_m + \boldsymbol{N}^{-1}\boldsymbol{g}(\bar{\boldsymbol{q}})$$

The calculation of $\hat{\boldsymbol{d}}$ is generally computationally expensive, so it is preferred to precalculate it offline $\hat{\boldsymbol{d}}$ if it is possible (i.e in the repeated trajectories).

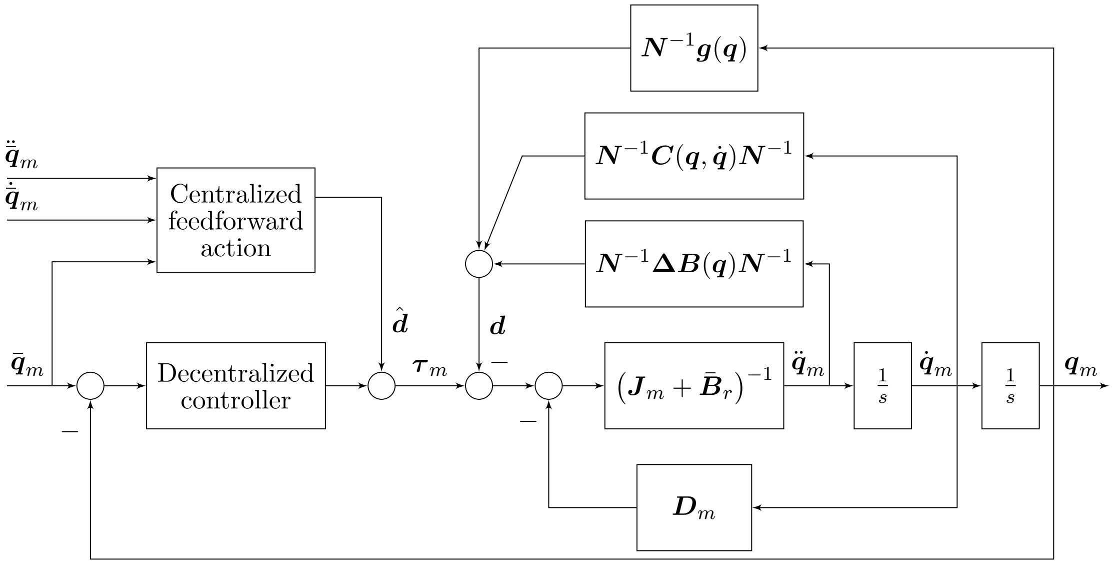

Figure 5.1: open loop feedforward $\boldsymbol{d}$ compensation

5.1.2 PD + gravity compensation

Let us consider again the simplified dynamics model Equation 4.1 and we try to define a control law based on it to hold the given pose $\bar{\boldsymbol{q}}$. To design a proper control law we will exploit the **Lyapunov method** Let us start defining the error function as

$$\tilde{\boldsymbol{q}}(t) = \bar{\boldsymbol{q}} - \boldsymbol{q}(t)$$

and the **Lyapunov function**

$$V(\tilde{\boldsymbol{q}}, \dot{\boldsymbol{q}}) = \frac{1}{2}\dot{\boldsymbol{q}}^\mathsf{T}\boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{q}} + \frac{1}{2}\tilde{\boldsymbol{q}}^\mathsf{T}\boldsymbol{K}_P\tilde{\boldsymbol{q}}$$

with $\boldsymbol{K}_P > 0$ and symmetrical.

$$
\begin{aligned}
\dot{V}(\dot{\boldsymbol{q}}, \tilde{\boldsymbol{q}}) &= \dot{\boldsymbol{q}}^\mathsf{T} \boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \frac{1}{2}\dot{\boldsymbol{q}}^\mathsf{T} \dot{\boldsymbol{B}}(\boldsymbol{q})\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}^\mathsf{T} \boldsymbol{K}_P \tilde{\boldsymbol{q}} \\
\text{using 4.1} \qquad &= \dot{\boldsymbol{q}}^\mathsf{T} \left(\boldsymbol{\tau} - \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \boldsymbol{g}(\boldsymbol{q})\right) + \frac{1}{2}\dot{\boldsymbol{q}}^\mathsf{T} \dot{\boldsymbol{B}}(\boldsymbol{q})\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}^\mathsf{T} \boldsymbol{K}_P \tilde{\boldsymbol{q}} \\
&= \frac{1}{2}\dot{\boldsymbol{q}}^\mathsf{T} \left(\dot{\boldsymbol{B}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\right)\dot{\boldsymbol{q}} + \dot{\boldsymbol{q}}^\mathsf{T} \left(\boldsymbol{\tau} - \boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{K}_P \tilde{\boldsymbol{q}}\right)
\end{aligned}
$$

as we saw in subsection 2.1.5 the first term is equal to zero

$$
= \dot{\boldsymbol{q}}^\mathsf{T} \left(\boldsymbol{\tau} - \boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{K}_P \tilde{\boldsymbol{q}}\right)
$$

if we impose the control law

$$
\boldsymbol{\tau} = \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{K}_P \tilde{\boldsymbol{q}} - \boldsymbol{K}_D \dot{\boldsymbol{q}} \tag{5.1}
$$

with $\boldsymbol{K}_D > 0$ and symmetrical, so $\dot{V}$ became

$$
\dot{V}(\dot{\boldsymbol{q}}, \tilde{\boldsymbol{q}}) = -\dot{\boldsymbol{q}}^\mathsf{T} \boldsymbol{K}_D \dot{\boldsymbol{q}}
$$

the function $\dot{V}(\dot{\boldsymbol{q}}, \tilde{\boldsymbol{q}})$ is semi-definite negative, but not definite negative (see the possible state $\dot{\boldsymbol{q}} = 0, \tilde{\boldsymbol{q}} \in \mathbb{R}$), so we have to study furthermore the asymptotically stability of the equilibrium $\dot{\boldsymbol{q}} = 0$. We consider the dynamic system from Equation 4.1 at the equilibrium $\dot{\boldsymbol{q}} = 0$ with the control law from Equation 5.1

$$
\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{K}_P \tilde{\boldsymbol{q}} - \boldsymbol{K}_D \dot{\boldsymbol{q}}
$$

and we get

$$
\boldsymbol{K}_P \tilde{\boldsymbol{q}} = 0 \implies \tilde{\boldsymbol{q}} = 0
$$

then for the **Krasowski - La Salle Lemma** we can state that the system with the control law from Equation 5.1 is globally asyntotically stable with the unique equilibrium in $\boldsymbol{q} = \bar{\boldsymbol{q}}$. This result is valid only if gravity contribution $\boldsymbol{g}(\boldsymbol{q})$ is perfectly compensated from the designed control law.

You can see the implemented control system in the Figure 5.2.

n.b. with this kind of regulator we do not have control on the time history with which the error $\tilde{\boldsymbol{q}}(t)$ goes to zero, so this way is not practicable if we want track a trajectory

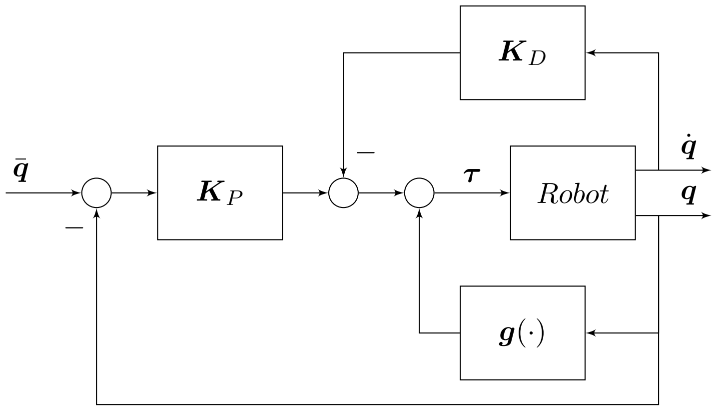

Figure 5.2: Closed loop with PD and gravity compensation

5.1.3 Inverse dynamics control

Let us try to solve the problem of the trajectory tracking. we rewrite the Equation 4.1 in the form

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{\tau}$$

with

$$\boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q})$$

If $\boldsymbol{B}(\boldsymbol{q})$ is full rank for each configuration of the manipulator we can define the control law called **inverse dynamics**

$$\boldsymbol{\tau} = \boldsymbol{B}(\boldsymbol{q})\boldsymbol{y} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \tag{5.2}$$

so, if the system knowledge is perfect, this law impose the dynamics $\ddot{\boldsymbol{q}} = \boldsymbol{y}$; from the extern the overall system appears as a double integrator.

Now we can design a control law for the function $\boldsymbol{y}$, a valid choice can be a **decoupled PD controller**

$$\boldsymbol{y} = \boldsymbol{K}_P\tilde{\boldsymbol{q}} + \boldsymbol{K}_I\dot{\tilde{\boldsymbol{q}}} + \ddot{\bar{\boldsymbol{q}}} \tag{5.3}$$

that imposes the error dynamics

$$\ddot{\tilde{\boldsymbol{q}}} + \boldsymbol{K}_D\dot{\tilde{\boldsymbol{q}}} + \boldsymbol{K}_P\tilde{\boldsymbol{q}} = 0$$

then, the error $\tilde{\boldsymbol{q}}$ is characterized by a second order dynamics that can be arbitrarily assigned by suitable choice of the parameters of the diagonal matrices $\boldsymbol{K}_P$ and $\boldsymbol{K}_I$. This allows us to use this technique to track an assigned trajectory.

You can see the implementation of this control in Figure 5.3.

n.b. this technique requires perfect knowledge of the dynamic model, which might be difficult in practice

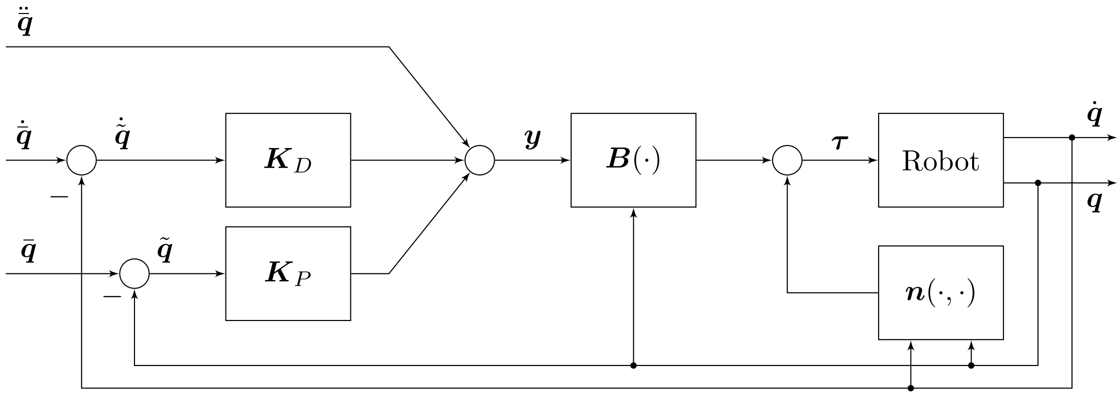

Figure 5.3: Closed loop with inverse dynamics control

5.1.3.1 Taking into account the uncertainty

As we saw in above the inverse dynamics control required perfect knowledge of the dynamics model of the robot, this is unrealistic in real robots, so we have to take into account the **model uncertainty**.

Let us consider a more realist control law

$$\boldsymbol{\tau} = \hat{\boldsymbol{B}}(\boldsymbol{q})\boldsymbol{y} + \hat{\boldsymbol{n}}(\boldsymbol{q}, \dot{\boldsymbol{q}})$$

so the compensated system become

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \hat{\boldsymbol{B}}(\boldsymbol{q})\boldsymbol{y} + \hat{\boldsymbol{n}}(\boldsymbol{q}, \dot{\boldsymbol{q}})$$

Let us define the **uncertainty** as

$$\tilde{\boldsymbol{B}}(\boldsymbol{q}) = \hat{\boldsymbol{B}}(\boldsymbol{q}) - \boldsymbol{B}(\boldsymbol{q}), \qquad \tilde{\boldsymbol{n}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \hat{\boldsymbol{n}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}})$$

still under the assumption that $\boldsymbol{B}(\boldsymbol{q})$ is invertible, for $\ddot{\boldsymbol{q}}$ we can write

$$\ddot{\boldsymbol{q}} = \boldsymbol{y} - \boldsymbol{\eta}$$

where

$$\boldsymbol{\eta} = \left(\boldsymbol{I} - \boldsymbol{B}^{-1}\hat{\boldsymbol{B}}\right)\boldsymbol{y} - \boldsymbol{B}^{-1}\tilde{\boldsymbol{n}}(\boldsymbol{q}, \dot{\boldsymbol{q}})$$

If we adopt the same control law we saw before in Equation 5.3, the error dynamic become

$$\ddot{\tilde{\boldsymbol{q}}} + \boldsymbol{K}_D\dot{\tilde{\boldsymbol{q}}} + \boldsymbol{K}_P\tilde{\boldsymbol{q}} = \boldsymbol{\eta}$$

the system still nonlinear, so we have to add in addiction to the PD controller a nonlinear term, function of the error to improve the robustness of the final system.

Let us define the second order derivative of the error

$$\ddot{\tilde{\boldsymbol{q}}} = \ddot{\boldsymbol{q}} - \ddot{\boldsymbol{q}} = \ddot{\boldsymbol{q}} - \boldsymbol{y} + \boldsymbol{\eta}$$

we define a new system whose state are the errors

$$\boldsymbol{\xi} = \begin{bmatrix} \tilde{\boldsymbol{q}}^{\mathsf{T}} \\ \dot{\tilde{\boldsymbol{q}}}^{\mathsf{T}} \end{bmatrix}$$

and the dynamics is defined by

$$\dot{\boldsymbol{\xi}} = \boldsymbol{H}\boldsymbol{\xi} + \boldsymbol{D}\left(\ddot{\boldsymbol{q}} - \boldsymbol{y} + \boldsymbol{\eta}\right)$$

with

$$\boldsymbol{H} = \begin{bmatrix} \mathbf{0} & \boldsymbol{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{2n\times 2n}, \qquad \boldsymbol{D} = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{I} \end{bmatrix} \in \mathbb{R}^{2n\times n}$$

Before we go any further, we need to characterize the uncertainty with three assumptions

- $\sup_{t\geq 0}\|\ddot{\boldsymbol{q}}\| < Q_M < \infty \qquad \forall \ddot{\boldsymbol{q}}$

  This assumption requires that the required acceleration is not infinite. It is obviously always verified, because a planned trajectory will never require an unlimited acceleration.

- $\|\boldsymbol{I} - \boldsymbol{B}^{-1}\hat{\boldsymbol{B}}\| \leq \alpha < 1 \qquad \forall \boldsymbol{q}$

  A matrix $\boldsymbol{B}$ is definite positive and it is lower and upper bound, so the following equation is always valid

  $$0 < B_m \leq \|\boldsymbol{B}^{-1}\| \leq B_M < \infty$$

  then, we can impose

  $$\hat{\boldsymbol{B}} = \frac{2}{B_M + B_m}\boldsymbol{I}$$

that always satisfy the assumption

$$\|\boldsymbol{I} - \boldsymbol{B}^{-1}\hat{\boldsymbol{B}}\| \leq \frac{B_M - B_m}{B_M + B_m} < 1$$

n.b. $\hat{\boldsymbol{B}} = \boldsymbol{B} \implies \alpha = 0$

- $\|\tilde{\boldsymbol{n}}\| \leq \Phi(\|\boldsymbol{\xi}\|) < \infty \qquad \forall \boldsymbol{q}, \dot{\boldsymbol{q}}$
  We can choose the form

$$\Phi(\|\boldsymbol{\xi}\|) = \alpha_0 + \alpha_1\|\boldsymbol{\xi}\| + \alpha_2\|\boldsymbol{\xi}\|^2$$

We add a term to the control law Equation 5.3

$$\boldsymbol{y} = \boldsymbol{K}_P\tilde{\boldsymbol{q}} + \boldsymbol{K}_I\dot{\tilde{\boldsymbol{q}}} + \ddot{\boldsymbol{q}} + \boldsymbol{w}$$

now we consider the error dynamics with this control law

$$\begin{aligned} \dot{\boldsymbol{\xi}} &= \boldsymbol{H}\boldsymbol{\xi} + \boldsymbol{D}\left(\boldsymbol{\eta} - \boldsymbol{K}_P\tilde{\boldsymbol{q}} - \boldsymbol{K}_I\dot{\tilde{\boldsymbol{q}}} - \boldsymbol{w}\right) \\ &= \tilde{\boldsymbol{H}}\boldsymbol{\xi} + \boldsymbol{D}\left(\boldsymbol{\eta} - \boldsymbol{w}\right) \end{aligned}$$

defining $\boldsymbol{K} = \begin{bmatrix}\boldsymbol{K}_P & \boldsymbol{K}_D\end{bmatrix}$ with

$$\tilde{\boldsymbol{H}} = (\boldsymbol{H} - \boldsymbol{D}\boldsymbol{K}) = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{K}_P & \boldsymbol{K}_D \end{bmatrix}$$

n.b. all the eigenvalues of $\tilde{\boldsymbol{H}}$ *are negative*

Now, we need to design a control law for $\boldsymbol{w}$, and to do this we will use the **Lyapunov method**. Consider the following **Lyapunov function** candidate

$$V(\boldsymbol{\xi}) = \boldsymbol{\xi}^\mathsf{T}\boldsymbol{Q}\boldsymbol{\xi} > 0, \qquad \forall \boldsymbol{\xi} \div \boldsymbol{0}$$

where $\boldsymbol{Q}$ is symmetric positive definite matrix.

$$\begin{aligned} \dot{V} &= \dot{\boldsymbol{\xi}}^\mathsf{T}\boldsymbol{Q}\boldsymbol{\xi} + \boldsymbol{\xi}^\mathsf{T}\boldsymbol{Q}\dot{\boldsymbol{\xi}} \\ &= (\boldsymbol{\xi}^\mathsf{T}\tilde{\boldsymbol{H}}^\mathsf{T} + (\boldsymbol{\eta} - \boldsymbol{w})^\mathsf{T}\boldsymbol{D}^\mathsf{T})\boldsymbol{Q}\boldsymbol{\xi} + \boldsymbol{\xi}^\mathsf{T}\boldsymbol{Q}(\tilde{\boldsymbol{H}}\boldsymbol{\xi} + \boldsymbol{D}(\boldsymbol{\eta} - \boldsymbol{w})) \\ &= \boldsymbol{\xi}^\mathsf{T}\left(\tilde{\boldsymbol{H}}^\mathsf{T}\boldsymbol{Q} + \boldsymbol{Q}\tilde{\boldsymbol{H}}\right)\boldsymbol{\xi} + (\boldsymbol{\eta} - \boldsymbol{w})^\mathsf{T}\boldsymbol{D}^\mathsf{T}\boldsymbol{Q}\boldsymbol{\xi} + \boldsymbol{\xi}^\mathsf{T}\boldsymbol{Q}\boldsymbol{D}(\boldsymbol{\eta} - \boldsymbol{w}) \end{aligned}$$

transposing the second element because it is a scalar

$$= \boldsymbol{\xi}^\top \left(\tilde{\boldsymbol{H}}^\top \boldsymbol{Q} + \boldsymbol{Q}\tilde{\boldsymbol{H}} \right) \boldsymbol{\xi} + 2\boldsymbol{\xi}^\top \boldsymbol{Q}\boldsymbol{D} \left(\boldsymbol{\eta} - \boldsymbol{w} \right)$$

set $\boldsymbol{z} = \boldsymbol{D}^\top \boldsymbol{Q}\boldsymbol{\xi}$ and $\tilde{\boldsymbol{H}}^\top \boldsymbol{Q} + \boldsymbol{Q}\tilde{\boldsymbol{H}} = -\boldsymbol{P}$

$$= -\boldsymbol{\xi}^\top \boldsymbol{P}\boldsymbol{\xi} + 2\boldsymbol{z}^\top \left(\boldsymbol{\eta} - \boldsymbol{w} \right)$$

n.b. because all eigenvalues of $\tilde{\boldsymbol{H}}$ are negative for any $\boldsymbol{P}$ positive definite symmetrical matrix also the solution $\boldsymbol{Q}$ is a positive definite symmetrical matrix

Let us analyze the derivative Lyapunov function, the first element $-\boldsymbol{\xi}^\top \boldsymbol{P}\boldsymbol{\xi}$ is always negative, so we need to analyze the second term. If $\boldsymbol{\xi} \in \ker(\boldsymbol{D}^\top \boldsymbol{Q})$, then $\boldsymbol{z} = 0$, so the system is asymptotically stable, in the other case we have to define a control function $\boldsymbol{w}$ that make the second term negative. From

$$\boldsymbol{w} = \rho \frac{\boldsymbol{z}}{\|\boldsymbol{z}\|}$$

we get

$$\boldsymbol{z}^\top \left(\boldsymbol{\eta} - \boldsymbol{w} \right) = \boldsymbol{z}^\top \left(\boldsymbol{\eta} - \rho \frac{\boldsymbol{z}}{\|\boldsymbol{z}\|} \right) = \boldsymbol{z}^\top \boldsymbol{\eta} - \rho \frac{\boldsymbol{z}^\top \boldsymbol{z}}{\|\boldsymbol{z}\|} \leq \|\boldsymbol{z}\|\|\boldsymbol{\eta}\| - \rho\|\boldsymbol{z}\| = \|\boldsymbol{z}\|(\|\boldsymbol{\eta}\| - \rho)$$

so, if $\rho > \|\boldsymbol{\eta}\|$ the globally asymptotically stability for the system is guaranteed. Looking for define a function ρ exploiting the assumptions

$$\begin{aligned} \|\boldsymbol{\eta}\| \leq \|\boldsymbol{I} - \boldsymbol{B}^{-1}\hat{\boldsymbol{B}}\| (\|\ddot{\boldsymbol{q}}\| + \|\boldsymbol{K}\|\|\boldsymbol{\xi}\| + \|\boldsymbol{w}\| + \|\boldsymbol{B}^{-1}\|\|\tilde{\boldsymbol{n}}\|) \\ \leq \alpha Q_M + \alpha\|\boldsymbol{K}\|\|\boldsymbol{\xi}\| + \alpha\rho + B_M \Phi(\|\boldsymbol{\xi}\|) < \rho \end{aligned}$$

so

$$\rho \geq \frac{1}{1-\alpha} \left(\alpha Q_M + \alpha\|\boldsymbol{K}\|\|\boldsymbol{\xi}\| + B_M \Phi(\|\boldsymbol{\xi}\|) \right)$$

under the assumption that $\Phi(\|\boldsymbol{\xi}\|)$ has the form $\alpha_0 + \alpha_1\|\boldsymbol{\xi}\| + \alpha_2\|\boldsymbol{\xi}\|^2$ we can chose

$$\rho(\|\boldsymbol{\xi}\|) = \beta_0 + \beta_1\|\boldsymbol{\xi}\| + \beta_2\|\boldsymbol{\xi}\|^2$$

with

$$\beta_0 \geq \frac{\alpha Q_M + \alpha_0 B_M}{1-\alpha}, \qquad \beta_1 \geq \frac{\alpha\|\boldsymbol{K}\| + \alpha_1 B_M}{1-\alpha}, \qquad \beta_2 \geq \frac{\alpha_2 B_M}{1-\alpha}$$

this control law guarantees a globally asymptotically stability of the system.

So the overall control law is composed by three terms

- $\hat{\boldsymbol{B}}(\boldsymbol{q})\boldsymbol{y} + \hat{\boldsymbol{n}}(\boldsymbol{q}, \dot{\boldsymbol{q}})$
  approximately compensated for the nonlinear terms
- $\boldsymbol{K}_P\tilde{\boldsymbol{q}} + \boldsymbol{K}_D\dot{\tilde{\boldsymbol{q}}} + \ddot{\bar{\boldsymbol{q}}}$
  stabilizes the nominal dynamic system in the error
- $\boldsymbol{w} = \rho(\|\boldsymbol{\xi}\|)\frac{\boldsymbol{z}}{\|\boldsymbol{z}\|}$
  gives robustness, counteracting the uncertainty

A further improvement can be done, in order to avoid high frequency switching of the control variable, called **chattering**, the third term can be changed to

$$\boldsymbol{w} = \begin{cases} \rho(\|\boldsymbol{\xi}\|)\frac{\boldsymbol{z}}{\|\boldsymbol{z}\|} & \|\boldsymbol{z}\| \geq \epsilon \\ \rho(\|\boldsymbol{\xi}\|)\frac{\boldsymbol{z}}{\epsilon} & \|\boldsymbol{z}\| < \epsilon \end{cases}$$

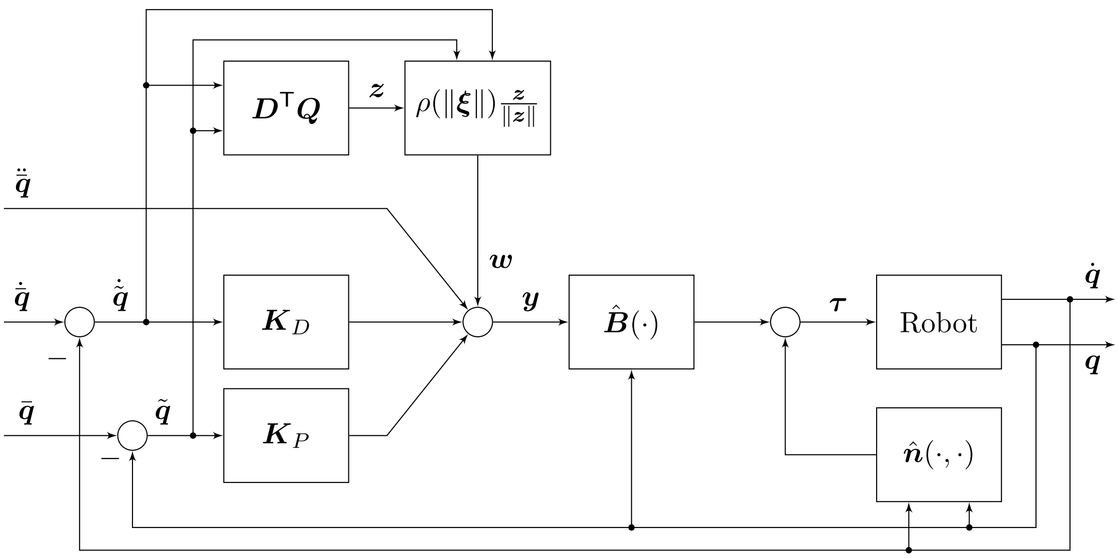

Figure 5.4: Closed loop with inverse dynamics robust control

5.1.3.2 Adaptive control

In additional to the inverse dynamics robust control (subsubsection 5.1.3.1) we can consider an adaptive control to compensate the uncertain dynamic parameters. As we saw in subsection 2.1.7 the dynamic model can write as

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\boldsymbol{\pi} = \boldsymbol{\tau}$$

where $\boldsymbol{\pi}$ is a suitable constant vector of uncertain dynamic parameters.

Let us consider the control law

$$\boldsymbol{\tau} = \boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}}_r + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_r + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{K}_D\boldsymbol{\sigma}$$

where

$$\dot{\boldsymbol{q}}_r = \dot{\tilde{\boldsymbol{q}}} + \boldsymbol{\Lambda}\tilde{\boldsymbol{q}} \qquad \ddot{\boldsymbol{q}}_r = \ddot{\tilde{\boldsymbol{q}}} + \boldsymbol{\Lambda}\dot{\tilde{\boldsymbol{q}}}$$

with $\boldsymbol{\Lambda}$ symmetrical positive define matrix (usually diagonal). If we impose

$$\boldsymbol{\sigma} = \dot{\boldsymbol{q}}_r - \dot{\boldsymbol{q}} = \dot{\tilde{\boldsymbol{q}}} + \boldsymbol{\Lambda}\tilde{\boldsymbol{q}}$$

we get as overall system

$$\begin{aligned}
\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) &= \boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}}_r + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_r + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{K}_D\boldsymbol{\sigma} \\
\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} &= \boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}}_r + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_r + \boldsymbol{K}_D\boldsymbol{\sigma} \\
\boldsymbol{B}(\boldsymbol{q})\left(\ddot{\boldsymbol{q}} - \ddot{\boldsymbol{q}}_r\right) + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\left(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_r\right) &= \boldsymbol{K}_D\boldsymbol{\sigma} \\
-\boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{\sigma}} - \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{\sigma} &= \boldsymbol{K}_D\boldsymbol{\sigma} \\
\boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{\sigma}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{\sigma} + \boldsymbol{K}_D\boldsymbol{\sigma} &= \boldsymbol{0} \qquad (5.4)
\end{aligned}$$

So let us check the stability of this system using the Lyapunov function

$$V(\boldsymbol{\sigma}, \tilde{\boldsymbol{q}}) = \frac{1}{2}\boldsymbol{\sigma}^{\mathsf{T}}\boldsymbol{B}(\boldsymbol{q})\boldsymbol{\sigma} + \frac{1}{2}\tilde{\boldsymbol{q}}^{\mathsf{T}}\boldsymbol{M}\tilde{\boldsymbol{q}}$$

$\boldsymbol{M}$ is (as always) a positive definite matrix, so the derivative is

$$\dot{V} = \boldsymbol{\sigma}^{\mathsf{T}}\boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{\sigma}} + \frac{1}{2}\boldsymbol{\sigma}^{\mathsf{T}}\dot{\boldsymbol{B}}(\boldsymbol{q})\boldsymbol{\sigma} + \tilde{\boldsymbol{q}}^{\mathsf{T}}\boldsymbol{M}\dot{\tilde{\boldsymbol{q}}}$$

using $\boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{\sigma}}$ from Equation 5.4

$$\begin{aligned}
&= \boldsymbol{\sigma}^{\mathsf{T}}\left(-\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \boldsymbol{K}_D\right)\boldsymbol{\sigma} + \frac{1}{2}\boldsymbol{\sigma}^{\mathsf{T}}\dot{\boldsymbol{B}}(\boldsymbol{q})\boldsymbol{\sigma} + \tilde{\boldsymbol{q}}^{\mathsf{T}}\boldsymbol{M}\dot{\tilde{\boldsymbol{q}}} \\
&= -\boldsymbol{\sigma}^{\mathsf{T}}\boldsymbol{K}_D\boldsymbol{\sigma} + \frac{1}{2}\boldsymbol{\sigma}^{\mathsf{T}}\left(\dot{\boldsymbol{B}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\right)\boldsymbol{\sigma} + \tilde{\boldsymbol{q}}^{\mathsf{T}}\boldsymbol{M}\dot{\tilde{\boldsymbol{q}}}
\end{aligned}$$

remembering the property of the matrix $\dot{\boldsymbol{B}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ seen in subsection 2.1.5

$$\begin{aligned}
&= -\boldsymbol{\sigma}^{\mathsf{T}}\boldsymbol{K}_D\boldsymbol{\sigma} + \tilde{\boldsymbol{q}}^{\mathsf{T}}\boldsymbol{M}\dot{\tilde{\boldsymbol{q}}} \\
&= -(\dot{\tilde{\boldsymbol{q}}} + \boldsymbol{\Lambda}\tilde{\boldsymbol{q}})^{\mathsf{T}}\boldsymbol{K}_D(\dot{\tilde{\boldsymbol{q}}} + \boldsymbol{\Lambda}\tilde{\boldsymbol{q}}) + \tilde{\boldsymbol{q}}^{\mathsf{T}}\boldsymbol{M}\dot{\tilde{\boldsymbol{q}}} \\
&= -\dot{\tilde{\boldsymbol{q}}}^{\mathsf{T}}\boldsymbol{K}_D\dot{\tilde{\boldsymbol{q}}} - \tilde{\boldsymbol{q}}^{\mathsf{T}}\boldsymbol{\Lambda}\boldsymbol{K}_D\boldsymbol{\Lambda}\tilde{\boldsymbol{q}} - 2\tilde{\boldsymbol{q}}^{\mathsf{T}}\boldsymbol{\Lambda}\boldsymbol{K}_D\dot{\tilde{\boldsymbol{q}}} + \tilde{\boldsymbol{q}}^{\mathsf{T}}\boldsymbol{M}\dot{\tilde{\boldsymbol{q}}}
\end{aligned}$$

choosing $\boldsymbol{M} = 2\boldsymbol{\Lambda}\boldsymbol{K}_D$

$$= -\dot{\tilde{\boldsymbol{q}}}^\mathsf{T}\boldsymbol{K}_D\dot{\tilde{\boldsymbol{q}}} - \tilde{\boldsymbol{q}}^\mathsf{T}\boldsymbol{\Lambda}\boldsymbol{K}_D\boldsymbol{\Lambda}\tilde{\boldsymbol{q}}$$

$\dot{V}$ is negative define thus the system is globally asymptotically stable with the equilibrium $\tilde{\boldsymbol{q}} = 0, \dot{\tilde{\boldsymbol{q}}} = 0$

Now let us consider a control law where the parameters are estimated and exploiting the formulation with the parametric vector we saw above

$$\begin{aligned}\boldsymbol{\tau} &= \hat{\boldsymbol{B}}(\boldsymbol{q})\ddot{\boldsymbol{q}}_r + \hat{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_r + \hat{\boldsymbol{g}}(\boldsymbol{q}) + \boldsymbol{K}_D\boldsymbol{\sigma} \\ &= \boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\dot{\boldsymbol{q}}_r,\ddot{\boldsymbol{q}}_r)\hat{\boldsymbol{\pi}} + \boldsymbol{K}_D\boldsymbol{\sigma}\end{aligned}$$

the system dynamic model Equation 5.4 become

$$\begin{aligned}\boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{\sigma}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\boldsymbol{\sigma} + \boldsymbol{K}_D\boldsymbol{\sigma} &= -\tilde{\boldsymbol{B}}(\boldsymbol{q})\ddot{\boldsymbol{q}}_r - \tilde{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_r - \tilde{\boldsymbol{g}}(\boldsymbol{q}) \\ &= -\boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\dot{\boldsymbol{q}}_r,\ddot{\boldsymbol{q}}_r)\tilde{\boldsymbol{\pi}}\end{aligned}$$

where $\tilde{\boldsymbol{B}}, \tilde{\boldsymbol{C}}, \tilde{\boldsymbol{g}}$ are the residual error given by the estimate of the model parameters, and thus the $\boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\dot{\boldsymbol{q}}_r,\ddot{\boldsymbol{q}}_r)\tilde{\boldsymbol{\pi}}$ can be seen as the model residual error.

So consider a modified version of the Lyapunov function saw before

$$V(\boldsymbol{\sigma},\tilde{\boldsymbol{q}},\tilde{\boldsymbol{\pi}}) = \frac{1}{2}\boldsymbol{\sigma}^\mathsf{T}\boldsymbol{B}(\boldsymbol{q})\boldsymbol{\sigma} + \tilde{\boldsymbol{q}}^\mathsf{T}\boldsymbol{\Lambda}\boldsymbol{K}_D\tilde{\boldsymbol{q}} + \frac{1}{2}\tilde{\boldsymbol{\pi}}^\mathsf{T}\boldsymbol{K}_\pi\tilde{\boldsymbol{\pi}}$$

where $\boldsymbol{M}$ is already substituted and $\boldsymbol{K}_\pi$ is a positive define matrix

$$\dot{V} = \boldsymbol{\sigma}^\mathsf{T}\boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{\sigma}} + \frac{1}{2}\boldsymbol{\sigma}^\mathsf{T}\dot{\boldsymbol{B}}(\boldsymbol{q})\boldsymbol{\sigma} + 2\tilde{\boldsymbol{q}}^\mathsf{T}\boldsymbol{\Lambda}\boldsymbol{K}_D\dot{\tilde{\boldsymbol{q}}} + \tilde{\boldsymbol{\pi}}^\mathsf{T}\boldsymbol{K}_\pi\dot{\tilde{\boldsymbol{\pi}}}$$

using $\boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{\sigma}}$ from uncertain dynamics model

$$\begin{aligned}&= \boldsymbol{\sigma}^\mathsf{T}\left(-\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) - \boldsymbol{K}_D\right)\boldsymbol{\sigma} - \boldsymbol{\sigma}^\mathsf{T}\boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\dot{\boldsymbol{q}}_r,\ddot{\boldsymbol{q}}_r)\tilde{\boldsymbol{\pi}} + \frac{1}{2}\boldsymbol{\sigma}^\mathsf{T}\dot{\boldsymbol{B}}(\boldsymbol{q})\boldsymbol{\sigma} \\ &\qquad + 2\tilde{\boldsymbol{q}}^\mathsf{T}\boldsymbol{\Lambda}\boldsymbol{K}_D\dot{\tilde{\boldsymbol{q}}} + \tilde{\boldsymbol{\pi}}^\mathsf{T}\boldsymbol{K}_\pi\dot{\tilde{\boldsymbol{\pi}}} \\ &= -\boldsymbol{\sigma}^\mathsf{T}\boldsymbol{K}_D\boldsymbol{\sigma} + \frac{1}{2}\boldsymbol{\sigma}^\mathsf{T}\left(\dot{\boldsymbol{B}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\right)\boldsymbol{\sigma} - \boldsymbol{\sigma}^\mathsf{T}\boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\dot{\boldsymbol{q}}_r,\ddot{\boldsymbol{q}}_r)\tilde{\boldsymbol{\pi}} \\ &\qquad + 2\tilde{\boldsymbol{q}}^\mathsf{T}\boldsymbol{\Lambda}\boldsymbol{K}_D\dot{\tilde{\boldsymbol{q}}} + \tilde{\boldsymbol{\pi}}^\mathsf{T}\boldsymbol{K}_\pi\dot{\tilde{\boldsymbol{\pi}}} \\ &= -\boldsymbol{\sigma}^\mathsf{T}\boldsymbol{K}_D\boldsymbol{\sigma} - \boldsymbol{\sigma}^\mathsf{T}\boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\dot{\boldsymbol{q}}_r,\ddot{\boldsymbol{q}}_r)\tilde{\boldsymbol{\pi}} + 2\tilde{\boldsymbol{q}}^\mathsf{T}\boldsymbol{\Lambda}\boldsymbol{K}_D\dot{\tilde{\boldsymbol{q}}} + \tilde{\boldsymbol{\pi}}^\mathsf{T}\boldsymbol{K}_\pi\dot{\tilde{\boldsymbol{\pi}}} \\ &= -\boldsymbol{\sigma}^\mathsf{T}\boldsymbol{K}_D\boldsymbol{\sigma} + 2\tilde{\boldsymbol{q}}^\mathsf{T}\boldsymbol{\Lambda}\boldsymbol{K}_D\dot{\tilde{\boldsymbol{q}}} + \tilde{\boldsymbol{\pi}}^\mathsf{T}(\boldsymbol{K}_\pi\dot{\tilde{\boldsymbol{\pi}}} - \boldsymbol{Y}^\mathsf{T}(\boldsymbol{q},\dot{\boldsymbol{q}},\dot{\boldsymbol{q}}_r,\ddot{\boldsymbol{q}}_r)\boldsymbol{\sigma})\end{aligned}$$

using $\boldsymbol{\sigma}$

$$
\begin{aligned}
&= -(\dot{\tilde{\boldsymbol{q}}} + \boldsymbol{\Lambda}\tilde{\boldsymbol{q}})^{\mathsf{T}}\boldsymbol{K}_D(\dot{\tilde{\boldsymbol{q}}} + \boldsymbol{\Lambda}\tilde{\boldsymbol{q}}) + 2\tilde{\boldsymbol{q}}^{\mathsf{T}}\boldsymbol{\Lambda}\boldsymbol{K}_D\dot{\tilde{\boldsymbol{q}}} + \tilde{\boldsymbol{\pi}}^{\mathsf{T}}(\boldsymbol{K}_\pi\dot{\tilde{\boldsymbol{\pi}}} - \boldsymbol{Y}^{\mathsf{T}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \dot{\boldsymbol{q}}_r, \ddot{\boldsymbol{q}}_r)\boldsymbol{\sigma}) \\
&= -\dot{\tilde{\boldsymbol{q}}}^{\mathsf{T}}\boldsymbol{K}_D\dot{\tilde{\boldsymbol{q}}} - \tilde{\boldsymbol{q}}^{\mathsf{T}}\boldsymbol{\Lambda}\boldsymbol{K}_D\boldsymbol{\Lambda}\tilde{\boldsymbol{q}} - 2\tilde{\boldsymbol{q}}^{\mathsf{T}}\boldsymbol{\Lambda}\boldsymbol{K}_D\dot{\tilde{\boldsymbol{q}}} + 2\tilde{\boldsymbol{q}}^{\mathsf{T}}\boldsymbol{\Lambda}\boldsymbol{K}_D\dot{\tilde{\boldsymbol{q}}} \\
&\qquad\qquad\qquad\qquad\qquad + \tilde{\boldsymbol{\pi}}^{\mathsf{T}}(\boldsymbol{K}_\pi\dot{\tilde{\boldsymbol{\pi}}} - \boldsymbol{Y}^{\mathsf{T}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \dot{\boldsymbol{q}}_r, \ddot{\boldsymbol{q}}_r)\boldsymbol{\sigma}) \\
&= -\dot{\tilde{\boldsymbol{q}}}^{\mathsf{T}}\boldsymbol{K}_D\dot{\tilde{\boldsymbol{q}}} - \tilde{\boldsymbol{q}}^{\mathsf{T}}\boldsymbol{\Lambda}\boldsymbol{K}_D\boldsymbol{\Lambda}\tilde{\boldsymbol{q}} + \tilde{\boldsymbol{\pi}}^{\mathsf{T}}(\boldsymbol{K}_\pi\dot{\tilde{\boldsymbol{\pi}}} - \boldsymbol{Y}^{\mathsf{T}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \dot{\boldsymbol{q}}_r, \ddot{\boldsymbol{q}}_r)\boldsymbol{\sigma})
\end{aligned}
$$

if we update the parameters' estimate as $\dot{\hat{\boldsymbol{\pi}}} = \boldsymbol{K}_\pi^{-1}\boldsymbol{Y}^{\mathsf{T}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \dot{\boldsymbol{q}}_r, \ddot{\boldsymbol{q}}_r)\boldsymbol{\sigma} = \dot{\tilde{\boldsymbol{\pi}}}$

$$
= -\dot{\tilde{\boldsymbol{q}}}^{\mathsf{T}}\boldsymbol{K}_D\dot{\tilde{\boldsymbol{q}}} - \tilde{\boldsymbol{q}}^{\mathsf{T}}\boldsymbol{\Lambda}\boldsymbol{K}_D\boldsymbol{\Lambda}\tilde{\boldsymbol{q}}
$$

$\dot{V} < 0$ thus, the system is globally asymptotically stable with the equilibrium $\tilde{\boldsymbol{q}} = 0, \dot{\tilde{\boldsymbol{q}}} = 0$

So, in summary the overall control law (which you can see in the Figure 5.5) is described by the system

$$
\begin{aligned}
\dot{\hat{\boldsymbol{\pi}}} &= \boldsymbol{K}_\pi^{-1}\boldsymbol{Y}^{\mathsf{T}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \dot{\boldsymbol{q}}_r, \ddot{\boldsymbol{q}}_r)(\dot{\tilde{\boldsymbol{q}}} + \boldsymbol{\Lambda}\tilde{\boldsymbol{q}}) \\
\boldsymbol{\tau} &= \boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \dot{\boldsymbol{q}}_r, \ddot{\boldsymbol{q}}_r)\hat{\boldsymbol{\pi}} + \boldsymbol{K}_D(\dot{\tilde{\boldsymbol{q}}} + \boldsymbol{\Lambda}\tilde{\boldsymbol{q}})
\end{aligned}
$$

in which we can identify three terms

- $\boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \dot{\boldsymbol{q}}_r, \ddot{\boldsymbol{q}}_r)\hat{\boldsymbol{\pi}}$
  can be interpreted as an approximate inverse dynamic control
- $\boldsymbol{K}_D(\dot{\tilde{\boldsymbol{q}}} + \boldsymbol{\Lambda}\tilde{\boldsymbol{q}})$
  has the behaviour of a PD acting on the tracking error
- $\boldsymbol{K}_\pi^{-1}\boldsymbol{Y}^{\mathsf{T}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \dot{\boldsymbol{q}}_r, \ddot{\boldsymbol{q}}_r)(\dot{\tilde{\boldsymbol{q}}} + \boldsymbol{\Lambda}\tilde{\boldsymbol{q}})$
  update the estimate of parameters in according to a gradient technique. The matrix $\boldsymbol{K}_\pi$ defines the speed of the convergence of the estimate to its asymptotic value.

n.b. asymptotically we will not get $\hat{\boldsymbol{\pi}} \to \boldsymbol{\pi}$, but only $\boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \dot{\boldsymbol{q}}_r, \ddot{\boldsymbol{q}}_r)(\hat{\boldsymbol{\pi}} - \boldsymbol{\pi}) \to \boldsymbol{0}$

In conclusion the **adaptive control** offers worse performance than the **robust control** in presence of model errors, but this first guarantees a more regular control action than the **robust control**, which are characterized by potentially dangerous high frequency switching.

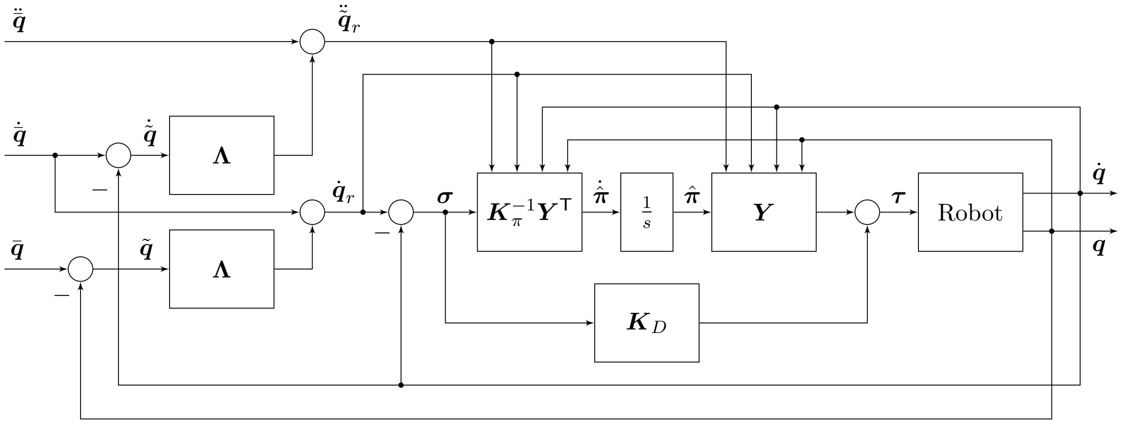

Figure 5.5: Adaptive control scheme

5.2 Control in operational space

In opposite to the control in the joints space, we can implement the control directly in the operational space (a Cartesian space).

As first difference by the control in the joints space, we will not apply the kinematic inversion, but the measures are actually the result of the direct kinematics computations based (generally) on the joint measures.

5.2.1 PD + gravity compensation

In a cartesian space we can define the error as

$$\tilde{\boldsymbol{x}} = \bar{\boldsymbol{x}} - \boldsymbol{x}$$

Let us consider a Lyapunov function

$$V(\tilde{\boldsymbol{x}}, \dot{\boldsymbol{q}}) = \frac{1}{2}\dot{\boldsymbol{q}}^\top \boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{q}} + \frac{1}{2}\tilde{\boldsymbol{x}}^\top \boldsymbol{K}_P \tilde{\boldsymbol{x}}$$

with $\boldsymbol{K}_P$ symmetric positive definite matrix

$$\dot{V} = \dot{\boldsymbol{q}}^\top \boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \frac{1}{2}\dot{\boldsymbol{q}}^\top \dot{\boldsymbol{B}}(\boldsymbol{q})\dot{\boldsymbol{q}} + \dot{\tilde{\boldsymbol{x}}}^\top \boldsymbol{K}_P \tilde{\boldsymbol{x}}$$

exploiting the **analytic Jacobian** we have $\dot{\tilde{\boldsymbol{x}}} = \dot{\boldsymbol{x}} = -\boldsymbol{J}_A(\boldsymbol{q})\dot{\boldsymbol{q}}$

$$= \dot{\boldsymbol{q}}^\top \boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \frac{1}{2}\dot{\boldsymbol{q}}^\top \dot{\boldsymbol{B}}(\boldsymbol{q})\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}^\top \boldsymbol{J}_A^\top(\boldsymbol{q})\boldsymbol{K}_P \tilde{\boldsymbol{x}}$$

replacing $\boldsymbol{B}\ddot{\boldsymbol{q}}$ with the Equation 4.1

$$= \dot{\boldsymbol{q}}^{\mathsf{T}}\left(\boldsymbol{\tau} - \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \boldsymbol{g}(\boldsymbol{q})\right) + \frac{1}{2}\dot{\boldsymbol{q}}^{\mathsf{T}}\dot{\boldsymbol{B}}(\boldsymbol{q})\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}^{\mathsf{T}}\boldsymbol{J}_A^{\mathsf{T}}(\boldsymbol{q})\boldsymbol{K}_P\tilde{\boldsymbol{x}}$$
$$= \frac{1}{2}\dot{\boldsymbol{q}}^{\mathsf{T}}\left(\dot{\boldsymbol{B}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\right)\dot{\boldsymbol{q}} + \dot{\boldsymbol{q}}^{\mathsf{T}}\left(\boldsymbol{\tau} - \boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{J}_A^{\mathsf{T}}(\boldsymbol{q})\boldsymbol{K}_P\tilde{\boldsymbol{x}}\right)$$

with the property we saw in subsection 2.1.5 we can eliminate the first term

$$= \dot{\boldsymbol{q}}^{\mathsf{T}}\left(\boldsymbol{\tau} - \boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{J}_A^{\mathsf{T}}(\boldsymbol{q})\boldsymbol{K}_P\tilde{\boldsymbol{x}}\right)$$

consider the control law $\boldsymbol{\tau} = \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}_A^{\mathsf{T}}(\boldsymbol{q})\boldsymbol{K}_P\tilde{\boldsymbol{x}} - \boldsymbol{J}_A^{\mathsf{T}}(\boldsymbol{q})\boldsymbol{K}_D\boldsymbol{J}_A(\boldsymbol{q})\dot{\boldsymbol{q}}$

$$= \dot{\boldsymbol{q}}^{\mathsf{T}}\left(\boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}_A^{\mathsf{T}}(\boldsymbol{q})\boldsymbol{K}_P\tilde{\boldsymbol{x}} - \boldsymbol{J}_A^{\mathsf{T}}(\boldsymbol{q})\boldsymbol{K}_D\boldsymbol{J}_A(\boldsymbol{q})\dot{\boldsymbol{q}} - \boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{J}_A^{\mathsf{T}}(\boldsymbol{q})\boldsymbol{K}_P\tilde{\boldsymbol{x}}\right)$$
$$= -\dot{\boldsymbol{q}}^{\mathsf{T}}\boldsymbol{J}_A^{\mathsf{T}}(\boldsymbol{q})\boldsymbol{K}_D\boldsymbol{J}_A(\boldsymbol{q})\dot{\boldsymbol{q}}$$

we can state $\dot{V} \leq 0$ ($\dot{V} = 0$ for $\dot{\boldsymbol{q}} = 0, \forall\tilde{\boldsymbol{x}} \in \mathbb{R}$) so let us check the system trajectory. The dynamics model (Equation 4.1) with the control law become

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = \boldsymbol{J}_A^{\mathsf{T}}(\boldsymbol{q})\boldsymbol{K}_P\tilde{\boldsymbol{x}} - \boldsymbol{J}_A^{\mathsf{T}}(\boldsymbol{q})\boldsymbol{K}_D\boldsymbol{J}_A(\boldsymbol{q})\dot{\boldsymbol{q}}$$

evaluated in the equilibrium $\dot{\boldsymbol{q}} = 0, \ddot{\boldsymbol{q}} = 0$

$$\boldsymbol{J}_A^{\mathsf{T}}(\boldsymbol{q})\boldsymbol{K}_P\tilde{\boldsymbol{x}} = \boldsymbol{0}$$

so, if the Jacobian is full rank the only solution is $\tilde{\boldsymbol{x}} = 0$, thus the system is globally asymptotically stable with the equilibrium $\bar{\boldsymbol{x}} = \boldsymbol{x}$. You can see the corresponding scheme in Figure 5.6

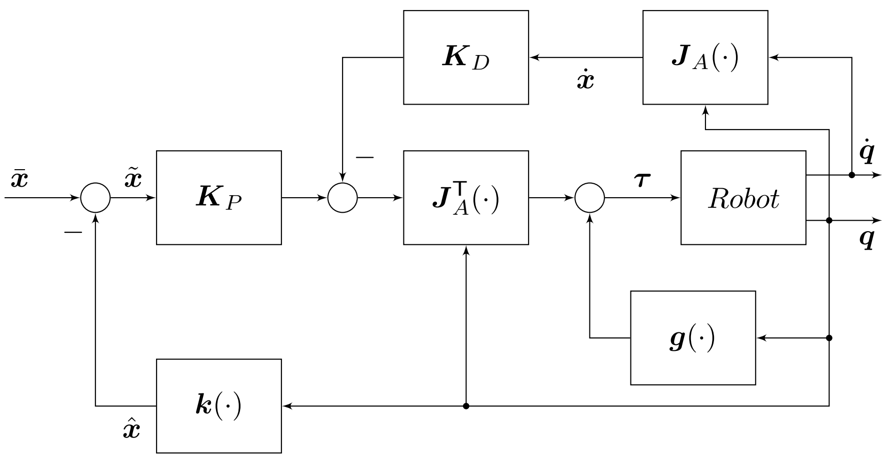

Figure 5.6: PD and gravity compensation in the operational space

5.2.2 Inverse dynamics control

To tracking a trajectory the **PD + gravity compesation** is not usable (it is constrained by $\dot{\bar{\boldsymbol{x}}} = 0$), so we have to implement a different control system, in particular we can adapt the **inverse dynamics control** designed to the control in the **joints space** saw in subsection 5.1.3.

As we saw if we impose the control law in Equation 5.2 we get that $\ddot{\boldsymbol{q}} = \boldsymbol{y}$, so we have to design a new input $\boldsymbol{y}$ in such a way to allow the trackinf of the trajectory $\bar{\boldsymbol{x}}(t)$. A way to do this is notice that

$$\ddot{\boldsymbol{x}} = \frac{d}{dt}\dot{\boldsymbol{x}} = \frac{d}{dt}\left(\boldsymbol{J}_A(\boldsymbol{q})\dot{\boldsymbol{q}}\right) = \boldsymbol{J}_A(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_A(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}$$

so we can invert the function and get

$$\ddot{\boldsymbol{q}} = \boldsymbol{J}_A^{-1}(\boldsymbol{q})\left(\ddot{\boldsymbol{x}} - \dot{\boldsymbol{J}}_A(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}\right)$$

this suggests that a possible control law for $\boldsymbol{y}$ can be

$$\boldsymbol{y} = \boldsymbol{J}_A^{-1}(\boldsymbol{q})\left(\ddot{\bar{\boldsymbol{x}}} + \boldsymbol{K}_D\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_P\tilde{\boldsymbol{x}} - \dot{\boldsymbol{J}}_A(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}\right) \tag{5.5}$$

so the overall control law become

$$\boldsymbol{\tau} = \boldsymbol{B}(\boldsymbol{q})\boldsymbol{J}_A^{-1}(\boldsymbol{q})\left(\ddot{\bar{\boldsymbol{x}}} + \boldsymbol{K}_D\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_P\tilde{\boldsymbol{x}} - \dot{\boldsymbol{J}}_A(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}\right) + \boldsymbol{n}(\boldsymbol{q},\dot{\boldsymbol{q}})$$

Let us find the error dynamics beginning from the dynamic model subject the above control law

$$\begin{aligned}
\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{n}(\boldsymbol{q},\dot{\boldsymbol{q}}) &= \boldsymbol{B}(\boldsymbol{q})\boldsymbol{J}_A^{-1}(\boldsymbol{q})\left(\ddot{\bar{\boldsymbol{x}}} + \boldsymbol{K}_D\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_P\tilde{\boldsymbol{x}} - \dot{\boldsymbol{J}}_A(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}\right) + \boldsymbol{n}(\boldsymbol{q},\dot{\boldsymbol{q}}) \\
\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} &= \boldsymbol{B}(\boldsymbol{q})\boldsymbol{J}_A^{-1}(\boldsymbol{q})\left(\ddot{\bar{\boldsymbol{x}}} + \boldsymbol{K}_D\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_P\tilde{\boldsymbol{x}} - \dot{\boldsymbol{J}}_A(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}\right) \\
\boldsymbol{0} &= \ddot{\bar{\boldsymbol{x}}} + \boldsymbol{K}_D\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_P\tilde{\boldsymbol{x}} - \dot{\boldsymbol{J}}_A(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \boldsymbol{J}_A(\boldsymbol{q})\ddot{\boldsymbol{q}}
\end{aligned}$$

substitute $\ddot{\boldsymbol{q}}$

$$\begin{aligned}
\boldsymbol{0} &= \ddot{\bar{\boldsymbol{x}}} + \boldsymbol{K}_D\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_P\tilde{\boldsymbol{x}} - \dot{\boldsymbol{J}}_A(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \ddot{\boldsymbol{x}} - \dot{\boldsymbol{J}}_A(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} \\
\boldsymbol{0} &= \ddot{\bar{\boldsymbol{x}}} + \boldsymbol{K}_D\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_P\tilde{\boldsymbol{x}} - \ddot{\boldsymbol{x}}
\end{aligned}$$

recognize $\ddot{\tilde{\boldsymbol{x}}} = \ddot{\bar{\boldsymbol{x}}} - \ddot{\boldsymbol{x}}$

$$\boldsymbol{0} = \ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_D\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_P\tilde{\boldsymbol{x}}$$

So the dynamics of the error in the operative space is characterized by a second order function with $\boldsymbol{K}_P, \boldsymbol{K}_D$ arbitrarly assigned. You can see the schema in Figure 5.7.

*n.b. to compute the control law we have to invert the Jacobian $\boldsymbol{J}_A^{-1}$, thus this method can not been used for the control of the **redundant robots** or for robots in **singular configurations***

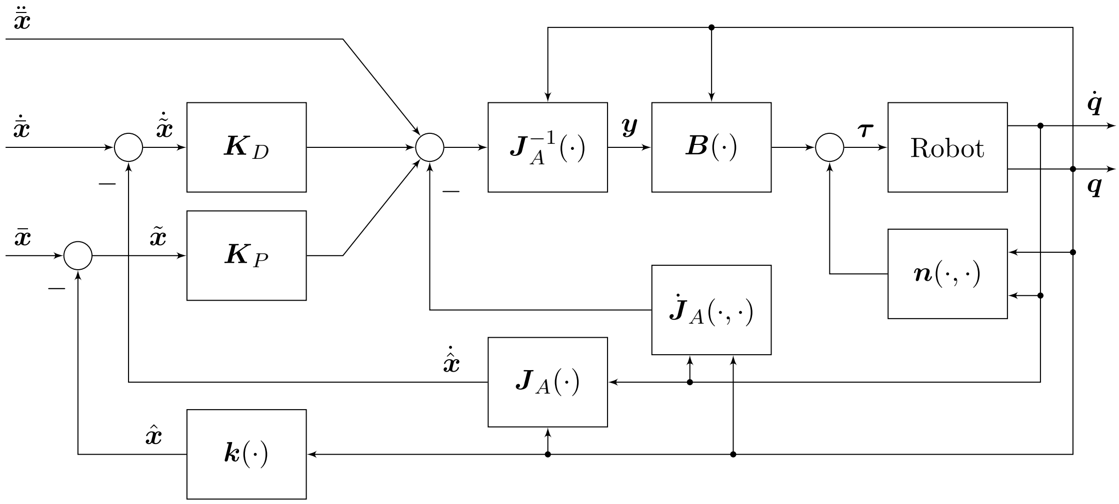

Figure 5.7: Inverse dynamics control in the operational space

Chapter 6

Control of the interaction

In this chapter we will analyze how the robot interact with the environment, and we will develop techniques to perform this task based on the control of the forces acting on the robot (generally on the end effector).

This kind of study is essential for the robots will interact with humans (collaborative robots).

Passive control of compliance In some tasks to the robots are requested a certain degree of flexibility in contrast to the request of the trajectory requested is strictly followed ($\boldsymbol{x}(t) = \bar{\boldsymbol{x}}(t)$), for example in an assembly task where it is required to insert an element in a hole (the alignment of end effector with the hole might be not perfect), in this case the problem can be bypassed equipping the robot with an **Remote Center of Compliance**[1]. The RCC placed between the robot's wrist and the gripper introduces a form of compliance for the axis perpendicular to the approach one, correcting misalignment in a passive way.

More flexibility behaviour can be gotten with an active control.

Forces measurements The measurements of forces and moments are provided by forces sensors that return the measurements of these alone the three axis bound on a local frame (generally in the proximity of the end effector).

The forces sensors are generally based on **strain gauges**[2] (a device that changes its conductance in function of strain). The strain gauges are suitable mounted in way to allow the measure of the six component of forces and moments.

[1] `https://en.wikipedia.org/wiki/Remote_Center_Compliance`
[2] `https://en.wikipedia.org/wiki/Strain_gauge`

6.1 Forces in statics

Let us start from the study of the robots' statics subjected to forces (and moments) acting on the end-effector. We will use the principle of the virtual works. For the torques joints we find the contribution as

$$dW_\tau = \boldsymbol{\tau}^\mathsf{T} d\boldsymbol{q}$$

and for the forces on the end effector

$$dW_\gamma = \boldsymbol{f}^\mathsf{T} d\boldsymbol{p}_e + \boldsymbol{\mu}^\mathsf{T} \boldsymbol{\omega}_e dt$$

with $\boldsymbol{f}$ and $\boldsymbol{\mu}$ are respectively the resulting force and moment act on the end effector. Exploiting the **geometrical Jacobian** we can write dW_γ as a function of $\boldsymbol{q}$

$$dW_\gamma = \boldsymbol{f}^\mathsf{T} \boldsymbol{J}_P(\boldsymbol{q}) d\boldsymbol{q} + \boldsymbol{\mu}^\mathsf{T} \boldsymbol{J}_O(\boldsymbol{q}) d\boldsymbol{q} = \boldsymbol{\gamma}^\mathsf{T} \boldsymbol{J}(\boldsymbol{q}) d\boldsymbol{q}$$

where the resulting force and moment are written in the vector $\boldsymbol{\gamma} = \begin{bmatrix} \boldsymbol{f}^\mathsf{T} & \boldsymbol{\mu}^\mathsf{T} \end{bmatrix}^\mathsf{T}$. The elementary movement and the virtual movement coincide, so we can write

$$\begin{aligned} \delta W_\tau &= \boldsymbol{\tau}^\mathsf{T} \delta\boldsymbol{q} \\ \delta W_\gamma &= \boldsymbol{\gamma}^\mathsf{T} \boldsymbol{J}(\boldsymbol{q}) \delta\boldsymbol{q} \end{aligned}$$

The robot is in a static state if $\delta W_\tau = \delta W_\gamma$, that is satisfied with

$$\boldsymbol{\tau} = \boldsymbol{J}^\mathsf{T}(\boldsymbol{q}) \boldsymbol{\gamma}$$

*n.b. the static relation shows a duality with the robot kinetics that can be written in the form $\boldsymbol{v}_e = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}$ **(kineto-static duality)***

n.b. if $\boldsymbol{\gamma} \in \ker(\boldsymbol{J}^\mathsf{T})$, then it does not require any joint torque to balance $\boldsymbol{\gamma}$

6.2 The concept of impedance

With a generalized approach we can state that for a dynamical system a power **flow** tent to change a generalized **effort**. We can easily see this behaviour in an electrical system, where the flow is the current and the effort is the voltage, the relation between these two measure is expressed by the impedance. In a mechanical system we can find this duality with **force** (flow) and the **position** (effort). So we can define the mechanical impedance and design a control system based on it.

Let us consider a 1 dof mass (M) on which acting two forces (u, f) (where a is the mass' acceleration)

$$Ma = u + f$$

we introduce a couple spring-dumper acting on the mass through u

$$u = -k_d v - K_e p$$

where we indicted with v, p respectively the mass' speed and position. So the system become

$$Ma + k_d v + K_e p = f$$

So we defined a relation between the force and the position (and its derivation) in a **mass-spring-damper** system; we will call this relation mechanical impedance.

6.2.1 Dynamics model with external forces

Let us try to extend the concept of mechanical impedance to the robots control. We consider the dynamical system with a $\boldsymbol{\gamma}$ force (and moment) acting on the end effector

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} - \boldsymbol{J}^{\mathsf{T}}(\boldsymbol{q})\boldsymbol{\gamma}$$

and we consider the control law based on the inverse dynamics

$$\boldsymbol{\tau} = \boldsymbol{B}(\boldsymbol{q})\boldsymbol{y} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q})$$

if we substitute it we get

$$\ddot{\boldsymbol{q}} = \boldsymbol{y} - \boldsymbol{B}^{-1}(\boldsymbol{q})\boldsymbol{J}^{\mathsf{T}}(\boldsymbol{q})\boldsymbol{\gamma}$$

now, assume for $\boldsymbol{y}$ the control law we saw for the inverse dynamic control in operative space (Equation 5.5), for the closed loop we get

$$\ddot{\boldsymbol{q}} = \boldsymbol{J}_A^{-1}(\boldsymbol{q})\left(\ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_D\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_P\tilde{\boldsymbol{x}} - \dot{\boldsymbol{J}}_A(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}\right) - \boldsymbol{B}^{-1}(\boldsymbol{q})\boldsymbol{J}^{\mathsf{T}}(\boldsymbol{q})\boldsymbol{\gamma}$$

$$\boldsymbol{J}_A(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_A(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = \ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_D\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_P\tilde{\boldsymbol{x}} - \boldsymbol{J}_A(\boldsymbol{q})\boldsymbol{B}^{-1}(\boldsymbol{q})\boldsymbol{J}^{\mathsf{T}}(\boldsymbol{q})\boldsymbol{\gamma}$$

recognizing $\boldsymbol{J}_A\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_A(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = \ddot{\boldsymbol{x}}$

$$\ddot{\boldsymbol{x}} = \ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_D\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_P\tilde{\boldsymbol{x}} - \boldsymbol{J}_A(\boldsymbol{q})\boldsymbol{B}^{-1}(\boldsymbol{q})\boldsymbol{J}^{\mathsf{T}}(\boldsymbol{q})\boldsymbol{\gamma}$$

$$\ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_D\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_P\tilde{\boldsymbol{x}} = \boldsymbol{J}_A(\boldsymbol{q})\boldsymbol{B}^{-1}(\boldsymbol{q})\boldsymbol{J}^{\mathsf{T}}(\boldsymbol{q})\boldsymbol{\gamma}$$

defining $\boldsymbol{J}^{\mathsf{T}}(\boldsymbol{q})\boldsymbol{\gamma} = \boldsymbol{J}_A^{\mathsf{T}}(\boldsymbol{q})\boldsymbol{\gamma}_A$

$$\ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_D\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_P\tilde{\boldsymbol{x}} = \boldsymbol{J}_A(\boldsymbol{q})\boldsymbol{B}^{-1}(\boldsymbol{q})\boldsymbol{J}_A^{\mathsf{T}}(\boldsymbol{q})\boldsymbol{\gamma}_A$$

and introducing $\boldsymbol{B}_A(\boldsymbol{q}) = \boldsymbol{J}_A^{-\mathsf{T}}(\boldsymbol{q})\boldsymbol{B}(\boldsymbol{q})\boldsymbol{J}_A^{-1}(\boldsymbol{q})$

$$\ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_D\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_P\tilde{\boldsymbol{x}} = \boldsymbol{B}_A^{-1}(\boldsymbol{q})\boldsymbol{\gamma}_A$$

we get an impedance relation, which is coupled and only partially assignable. But if we have also the forces' measurement we can change the control laws to exploit them

$$\begin{aligned} \boldsymbol{\tau} &= \boldsymbol{B}(\boldsymbol{q})\boldsymbol{y} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}^{\mathsf{T}}(\boldsymbol{q})\gamma \\ \boldsymbol{y} &= \boldsymbol{J}_A^{-1}(\boldsymbol{q})\boldsymbol{M}_d^{-1}\left(\boldsymbol{M}_d\ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{D}_d\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_d\tilde{\boldsymbol{x}} - \boldsymbol{M}_d\dot{\boldsymbol{J}}_A(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \boldsymbol{\gamma}_A\right) \end{aligned}$$

with $\boldsymbol{M}_d, \boldsymbol{D}_d, \boldsymbol{K}_d$ diagonal positive definite matrices; the closed loop become

$$\boldsymbol{M}_d\ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{D}_d\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_d\tilde{\boldsymbol{x}} = \boldsymbol{\gamma}_A$$

so, a completely decoupled system.

This form defines a **mechanical impedance** between the forces (and moments) and the position error in the operational space.

n.b. the ***mechanical impedance*** *have the same shape of a mass-spring-damper system with* $\boldsymbol{M}_d$ *mass,* $\boldsymbol{D}_d$ *damping and* $\boldsymbol{K}_d$ *stiffness*

Unluckily this method impose several constraints:

- it requires a complete knowledge of the dynamic model, it does not guarantee a compensation of the model errors
- it requires complete access to the robot, the control law is design directly on joints torques
- the system become inherently compliant to external disturbances, which is conflicting with the typical stiffness required to the industrial robots

6.2.2 Admittance control

Defining the **admittance filter** function (the inverse of the impedance)

$$H(s) = \frac{1}{\boldsymbol{M}_d s^2 + \boldsymbol{D}_d s + \boldsymbol{K}_d}$$

we can make the control system in Figure 6.1. In which

$$\bar{\boldsymbol{x}} - \boldsymbol{x} \approx \frac{1}{\boldsymbol{M}_d s^2 + \boldsymbol{D}_d s + \boldsymbol{K}_d} \boldsymbol{f}_{ext}$$

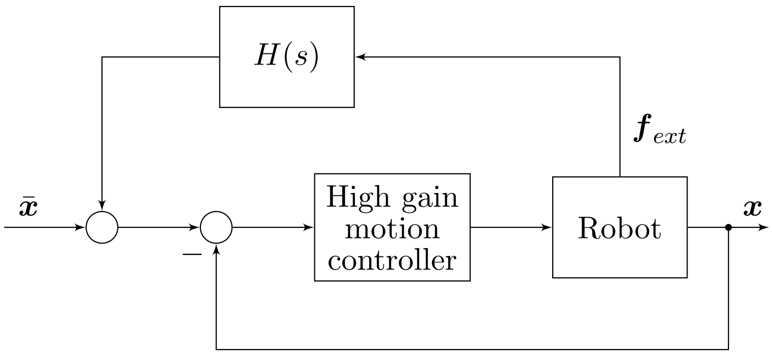

Figure 6.1: Admittance control

6.3 Natural constraints and artificial constraints

It is generally impossible to assign simultaneously an arbitrary position and force to a system, but in some task a control both on position and force is required, to handle this requirement we define a new concept based on constraints.

In the 3d space (with reference to a frame) we can identify 6 dof for the generalized movement velocity, 3 for the lineal speed (v_x, v_y, v_z) and 3 for the rotational speed $(\omega_x, \omega_y, \omega_z)$ to which we can associate 6 generalized forces (or moments), 3 linear forces (f_x, f_y, f_z) and 3 torques (μ_x, μ_y, μ_z).

So, we can define two kinds of constraints on this 12 variables:

We will call **natural constraints** ones that impose a limitation on a generalized movement velocity and forces, these are the constraints imposed by the environment. *e.g. a rigid plane impose a limit on the linear speed v_n in the direction defines by the plane normal, instead the absence of an obstacle in the generic direction of vector unit $\boldsymbol{u}$ (that is opposed to the motion) can not allow the imposition of the associated linear force f_u.*

We will call **artificial constraints** ones we can impose to the system arbitrarily *e.g. a pipe in a hole allows to impose an artificial constraints on the rotation ve-*

locity ω_l of the pipe in the hole (where $\boldsymbol{l}$ is vector unit in the direction of the pipe longitudinal axis)

It is evident that the artificial and natural constraints are complementary, and they suggest the control approach where the desire functions are defined only for the **artificial constraints**, while the state subject to the **natural constraints** are free to evolve.

So we can define the useful diagonal matrix $\boldsymbol{\Sigma}$ (6×6) with only zeros or ones as elements called **selection matrix**.

The $\boldsymbol{\Sigma}$ matrix is defined to weight the error on the reference for controlled forces the element (i,i) of the matrix is 1 for the others 0.

An approximate scheme of the control based on this approach can see in Figure 6.2, it is called **hybrid force/position control** and it is characterized by a high flexibility. It is possible identify some inconsistencies can appear due to nominal model used:

- a detected force in nominally free direction caused by a friction at the contact
- a detected movement speed in nominally constrained direction caused by compliance in the robot structure or contact
- uncertainty in the environment geometry at the contact

The first two are automatically filtered by the select matrix, the third can be mitigated by real time estimation process.

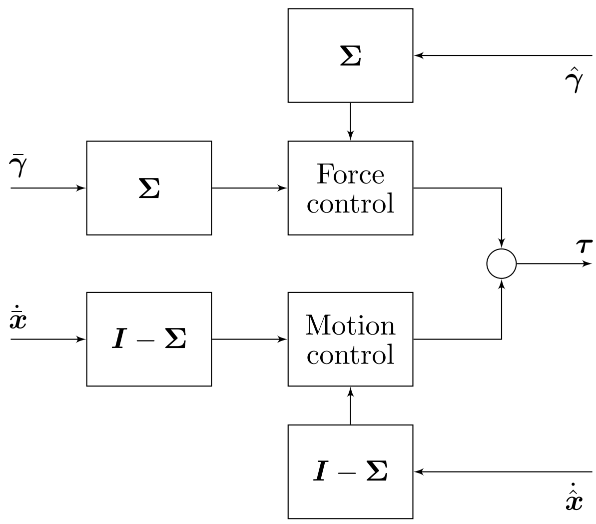

Figure 6.2: Hybrid force/position control

6.4 Explicit control

Consider a 1dof system with a mass m subject to 2 forces f, u subjected to the relation (where x is the position of the mass)

$$w + u = m\ddot{x}$$

if we introduce a spring with law $-kx$ on w we get

$$-kx + u = m\ddot{x}$$

in the **Laplace** form

$$x(s) = \frac{1}{ms^2 + k} u(s)$$

so if we evaluated the reaction force on the other side f of the spring we can get the relation

$$f(s) = \frac{k}{ms^2 + k} u(s)$$

an equivalent block diagram can be seen in the Figure 6.3

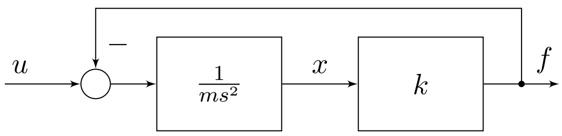

Figure 6.3: Spring-mass-environment interaction equivalent block diagram

The transfers function $\frac{F(s)}{U(s)}$ can be also written as

$$G_f(s) = \frac{\frac{k}{m}}{s^2 + \frac{k}{m}} = \frac{\omega_n^2}{s^2 + \omega_n^2}$$

highlighting it presents 2 imaginary poles in $\pm i\sqrt{k/m}$ and for $k \to \infty$ then $G_f(s) \to 1$

So we can close a loop around the force to regulate it (Figure 6.4)

a good choice for $R_f(s)$ is a pure integrator

$$R_f(s) = \frac{K_{if}}{s}$$

so we can directly assign ω_{cf} changing $K_{if} \approx \omega_{cf}$ if $\omega_{cf} < \omega_n$; with this choice the resonant frequency ω_n is out of the system bandwidth.

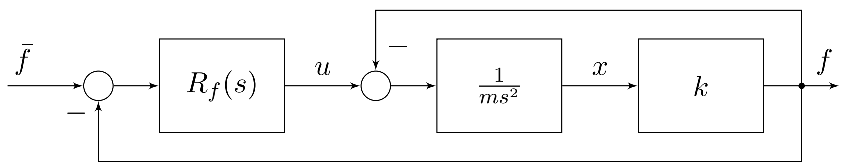

Figure 6.4: Explicit force control block diagram

6.5 Implicit control

Now let us start to analyze the same system saw for the explicit control when a control loop on position is already closed (Figure 6.5).

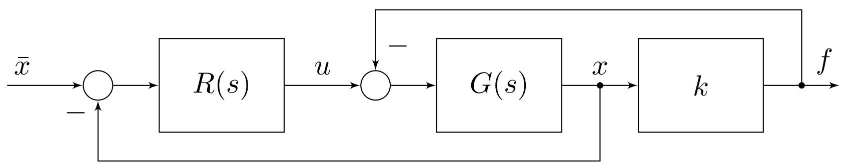

Figure 6.5: Closed loop on position

The corresponding closed loop is

$$\frac{f(s)}{\bar{x}(s)} = \frac{kR(s)G(s)}{1 + kG(s) + R(s)G(s)}$$

which for infinitely contact stiffness ($k \to \infty$) become $\frac{f(s)}{\bar{x}(s)} = R(s)$ and $f \approx u$

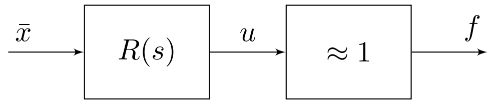

Figure 6.6: Closed loop on position with infinite contact stiffness

now, let us try to close a loop on the force with another controller. In particular, we can design a regulator composed by two action, one to compensate the position regulator influence ($C(s)$), and another one to impose the desired dynamics to f ($R_f(s)$), you can see this system in Figure 6.7.

Under the hypothesis that the position regulator is a PID

$$R(s) = k_P + k_I\frac{1}{s} + k_D s = \frac{k_D s^2 + k_P s + k_I}{s}$$

we can design the compensator as

$$C(s) = \frac{1}{k_D s^2 + k_P s + k_I}$$

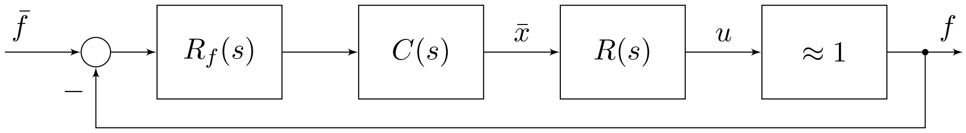

Figure 6.7: Closed loop on position with infinite contact stiffness

so the open loop function become (Figure 6.8)

$$L_f(s) = R_f(s)C(s)R(s) = \frac{1}{s}R_f(s)$$

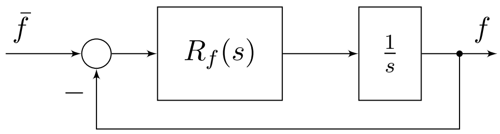

Figure 6.8: Compensated implicit force control

We can take $R_f(s)$ as PI controller

$$R_f(s) = k_{P_f} + k_{I_f}\frac{1}{s}$$

so finally the open loop function become

$$L_f(s) = \frac{sk_{P_f} + k_{I_f}}{s^2}$$

the ω_{c_f} can be set as $\omega_{c_f} \approx k_{P_f}$

Chapter 7

Control with vision sensors

7.1 Camera

A camera performs a 2D projection of a 3D space, this projection involves an information loss. To determinate coordinates of a 3D point from the 2D projection additional information are needed.

7.1.1 Perspective projection

If we identify a 3D point in the space with the tuple (X, Y, Z) and the corresponding point in the projection space with the tuple (x, y) we can identify the following relations

$$x = \frac{\lambda}{Z}X \quad y = \frac{\lambda}{Z}Y \tag{7.1}$$

Where λ is the focal length(in pixel) of the camera.

n.b. the above relations is valid only if 3D space and projection space share the same reference frame

7.1.2 Calibration

Before using the camera its need to be calibrated; we need to find two kinds of parameters

- Intrinsic parameters

  The parameters that characterize the camera as the focal length, and some kind of aberration in the lens that have to be considered in computer vision task. All these parameters are fixed for the camera, and it is not necessary to recalculate when the task changes.

- Extrinsic parameters

  All the other parameters used to map the 3D point into the projective space, like camera's orientation and position respect the global reference frame. If the camera is moved the extrinsic parameters have to be recalculated.

7.1.3 Camera configuration

There are two main camera configuration in vision control of manipulator

7.1.3.1 Eye-to-hand

The camera is fixed to the global reference frame, pointed toward the task space.

- Advantages
  - The movements of the robot does not affect the camera field of view
  - The geometric relation between projective space and the 3D space does not change
- Disadvantages
  - The movements of the robot may occlude the camera field of view

7.1.3.2 Eye-in-hand

The camera is attached to the robot, usually on the robot wrist.

- Advantages
  - The camera field of view are fixed on the end effector
  - The camera field of view is never occluded by the robots
- Disadvantages
  - The geometric relation between projective space and the 3D space changed with the robot movements
  - The camera field of view continuously change even for small movements

7.2 Control

The main control schemes based on computer vision are given by the combination of two measure methods and two actuation approaches.

7.2.1 Measure: Position based vs image based

The camera image can be use in two-way to produce measurements for the control

7.2.1.1 Position based

From the camera images a partial 3D representation of the task space is recreated exploiting technics of computer vision. Algorithms to estimate the pose are computably intense and sensitive to errors in camera calibration.

7.2.1.2 Image based

No transformation between projective space of the camera image to the real 3D worlds is attempted, the image is directly used to provide the measurements to the control system. The error for the controller is defined on a quantities directly measurable from the image.

7.2.2 Actuation: Dynamic look and move vs visual servoing

The action control based on the visual can be applied to different level of the control stack

7.2.2.1 Dynamic look and move

The visual information are used at high level of the control; the speed control is not based on visual data and only the position control exploiting the visual measurements. The camera can work at low framerate without compromising the overall performance of the position control system. Generally, in industrial application is the only way to go because the only accessible control variable on the robot is the position set point.

7.2.2.2 Visual servoing

The visual information are used also at the low level control; so, the access to the actuators input of the robots is necessary to exploiting the visual control to achieve

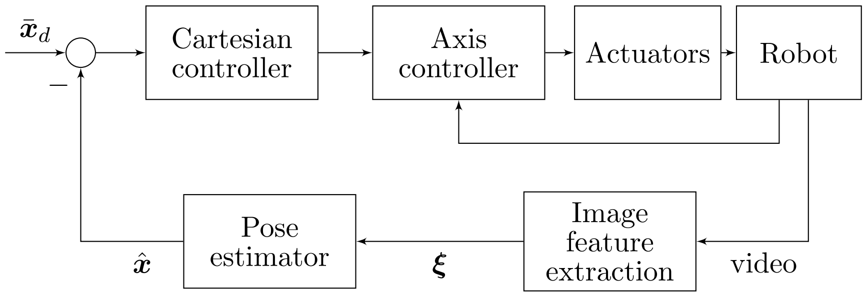

Figure 7.1: Dynamic look and move position based

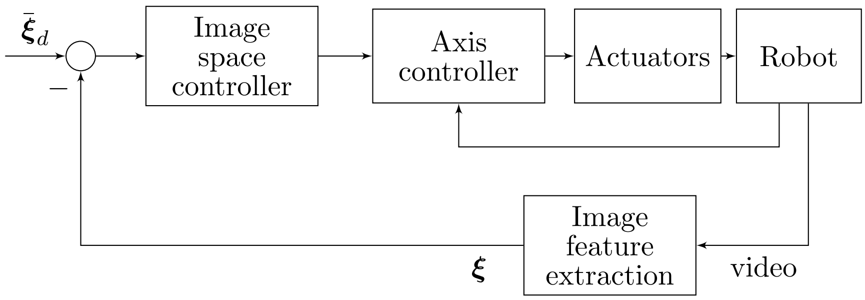

Figure 7.2: Dynamic look and move image based

high control performances. It is also required a high framerate to achieve high performances.

7.2.3 Image based schema

Design an image based control is a critical task; we need to relate motion of the camera with motion of the features in the image.

Let us consider a fixed point $\boldsymbol{P}$ in the space and the camera frame c, we can state

$$\boldsymbol{P}^w = \boldsymbol{O}_c^w(t) + \boldsymbol{R}_c^w(t)\boldsymbol{P}^c(t)$$

differentiating in time

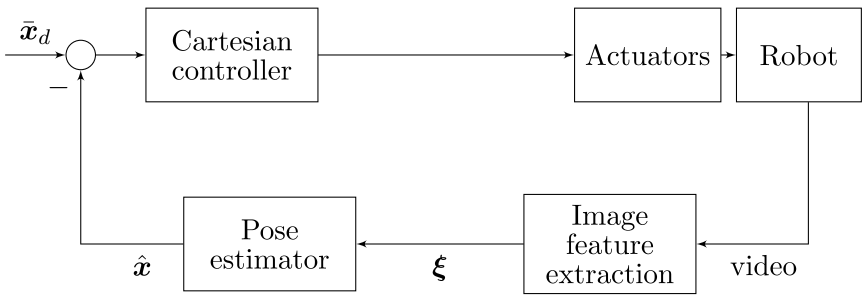

Figure 7.3: Visual servoing position based

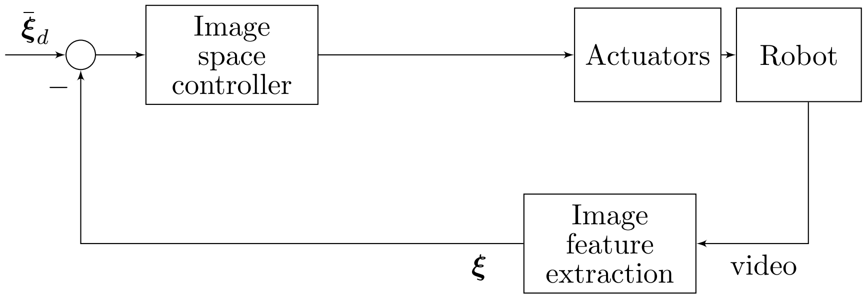

Figure 7.4: Visual servoing image based

$$\mathbf{0} = \dot{\boldsymbol{O}}_c^w + \dot{\boldsymbol{R}}_c^w \boldsymbol{P}^c + \boldsymbol{R}_c^w \dot{\boldsymbol{P}}^c$$

and find the solution for $\dot{\boldsymbol{P}}^c$

$$\dot{\boldsymbol{P}}^c = -\dot{\boldsymbol{O}}_c^c - \dot{\boldsymbol{R}}_c^c \boldsymbol{P}^c$$

remembering that $\dot{\boldsymbol{R}}\boldsymbol{P} = \boldsymbol{\omega} \times \boldsymbol{P}$ we get

$$\dot{\boldsymbol{P}}^c = -\dot{\boldsymbol{O}}_c^c - \boldsymbol{\omega}_c^c \times \boldsymbol{P}^c$$

where all the vector are expressed in the camera frame.
Define

$$\boldsymbol{P}^c = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad \boldsymbol{\omega}_c^c = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad \dot{\boldsymbol{O}}_c^c = \begin{bmatrix} \dot{O}_x \\ \dot{O}_y \\ \dot{O}_z \end{bmatrix}$$

we can write

$$\begin{aligned} \dot{X} &= Y\omega_z - Z\omega_y - \dot{O}_x \\ \dot{Y} &= Z\omega_x - X\omega_z - \dot{O}_y \\ \dot{Z} &= X\omega_y - Y\omega_x - \dot{O}_z \end{aligned}$$

remembering Equation 7.1 we get

$$\begin{aligned} \dot{X} &= \frac{x}{\lambda} Z\omega_z - Z\omega_y - \dot{O}_x \\ \dot{Y} &= Z\omega_x - \frac{y}{\lambda} Z\omega_z - \dot{O}_y \\ \dot{Z} &= \frac{x}{\lambda} Z\omega_y - \frac{y}{\lambda} Z\omega_x - \dot{O}_z \end{aligned}$$

Differentiating in time projective coordinate from Equation 7.1

$$\dot{x} = \lambda \frac{d}{dt}\frac{X}{Z} = \lambda \frac{\dot{X}Z - X\dot{Z}}{Z^2}$$
$$\dot{y} = \lambda \frac{d}{dt}\frac{Y}{Z} = \lambda \frac{\dot{Y}Z - Y\dot{Z}}{Z^2}$$

now we can finally write the relation between motion of the camera to the motion of the image's features

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \boldsymbol{L}(\lambda, x, y, Z) \begin{bmatrix} \dot{\boldsymbol{O}}_c^c \\ \boldsymbol{\omega}_c^c \end{bmatrix}$$

where

$$\boldsymbol{L}(\lambda, x, y, Z) = \begin{bmatrix} -\frac{\lambda}{Z} & 0 & \frac{x}{Z} & \frac{xy}{\lambda} & -\frac{\lambda^2+x^2}{\lambda} & y \\ 0 & -\frac{\lambda}{Z} & \frac{y}{Z} & \frac{\lambda^2+y^2}{\lambda} & -\frac{xy}{\lambda} & -x \end{bmatrix}$$

and it is called **interaction matrix**.

7.2.3.1 Interaction matrix

The **interaction matrix** is a 2×6 matrix, it depends on variable values x, y and Z. If it decomposed

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \boldsymbol{L}_O(\lambda, x, y, Z)\dot{\boldsymbol{O}}_c^c + \boldsymbol{L}_\omega(\lambda, x, y)\boldsymbol{\omega}_c^c$$

we can see that the rotation part does not depend on depth Z.

Due to the shape of it, we can state that exist an associated null space with dimension of 4, so there are four types of camera movement that not produce any change of the feature in the projective image.

7.2.3.2 Image Jacobian

If we try to bind camera speed we write

$$\begin{bmatrix} \dot{\boldsymbol{O}}_c^c \\ \boldsymbol{\omega}_c^c \end{bmatrix} = \boldsymbol{T}_w^c(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}$$

where $\boldsymbol{T}_w^c(\boldsymbol{q})$ is the coordinate transformation between world and camera frame, and $\boldsymbol{J}(\boldsymbol{q})$ is the robot **geometrical Jacobian**. So, we can write

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \boldsymbol{L}(\lambda, x, y, Z)\boldsymbol{T}_w^c(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{J}_I(\lambda, x, y, Z, \boldsymbol{q})\dot{\boldsymbol{q}}$$

where $\boldsymbol{J}_I$ is the **image Jacobian**.

The Z parameter is clearly not available, but it can be estimated based on apriori information.

Now the **image Jacobian** can be used to design the control law in the image based scheme.

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_I^\dagger \left(\dot{\xi}_d + k(\xi_d - \xi)x \right) + \left(\boldsymbol{I} - \boldsymbol{J}_I^\dagger \boldsymbol{J}_I \right) \dot{\boldsymbol{q}}_0$$

Chapter 8

Summary

8.1 Centralized control

8.1.1 Control in joint space

8.1.1.1 PD + gravity compensation

Control law

$$\boldsymbol{\tau} = \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{K}_P\tilde{\boldsymbol{q}} - \boldsymbol{K}_D\dot{\boldsymbol{q}}$$

Lyapunov's function

$$V(\tilde{\boldsymbol{q}}, \dot{\boldsymbol{q}}) = \frac{1}{2}\dot{\boldsymbol{q}}^\mathsf{T}\boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{q}} + \frac{1}{2}\tilde{\boldsymbol{q}}^\mathsf{T}\boldsymbol{K}_P\tilde{\boldsymbol{q}}$$

Closed loop

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = \boldsymbol{K}_P\tilde{\boldsymbol{q}} - \boldsymbol{K}_D\dot{\boldsymbol{q}}$$

Block diagram in Figure 5.2

8.1.1.2 Inverse dynamic control

Control law

$$\begin{aligned} \boldsymbol{\tau} &= \boldsymbol{B}(\boldsymbol{q})\boldsymbol{y} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \\ \boldsymbol{y} &= \boldsymbol{K}_P\tilde{\boldsymbol{q}} + \boldsymbol{K}_I\dot{\tilde{\boldsymbol{q}}} + \ddot{\tilde{\boldsymbol{q}}} \end{aligned}$$

Open loop

$$\ddot{\boldsymbol{q}} = \boldsymbol{y}$$

Closed loop

$$\ddot{\tilde{\boldsymbol{q}}} + \boldsymbol{K}_D\dot{\tilde{\boldsymbol{q}}} + \boldsymbol{K}_P\tilde{\boldsymbol{q}} = 0$$

Block diagram in Figure 5.3

8.1.2 Control in operational space

8.1.2.1 PD + gravity compensation

Control law

$$\boldsymbol{\tau} = \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}_A^\top(\boldsymbol{q})\boldsymbol{K}_P\tilde{\boldsymbol{x}} - \boldsymbol{J}_A^\top(\boldsymbol{q})\boldsymbol{K}_D\boldsymbol{J}_A(\boldsymbol{q})\dot{\boldsymbol{q}}$$

Lyapunov's function

$$V(\tilde{\boldsymbol{x}}, \dot{\boldsymbol{q}}) = \frac{1}{2}\dot{\boldsymbol{q}}^\top\boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{q}} + \frac{1}{2}\tilde{\boldsymbol{x}}^\top\boldsymbol{K}_P\tilde{\boldsymbol{x}}$$

Closed loop

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = \boldsymbol{J}_A^\top(\boldsymbol{q})\boldsymbol{K}_P\tilde{\boldsymbol{x}} - \boldsymbol{J}_A^\top(\boldsymbol{q})\boldsymbol{K}_D\boldsymbol{J}_A(\boldsymbol{q})\dot{\boldsymbol{q}}$$

Block diagram in Figure 5.6

8.1.2.2 Inverse dynamic control

Control law

$$\begin{aligned}\boldsymbol{\tau} &= \boldsymbol{B}(\boldsymbol{q})\boldsymbol{y} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}})\\ \boldsymbol{y} &= \boldsymbol{J}_A^{-1}(\boldsymbol{q})\left(\ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_D\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_P\tilde{\boldsymbol{x}} - \dot{\boldsymbol{J}}_A(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}\right)\end{aligned}$$

Open loop

$$\ddot{\boldsymbol{q}} = \boldsymbol{y}$$

Closed loop

$$\ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_D\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_P\tilde{\boldsymbol{x}} = 0$$

Block diagram in Figure 5.7

n.b. needs inversion of $\boldsymbol{J}_A$, then it cannot be used with redundant robots

8.2 Control of the interaction

8.2.1 Impedance control

System

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} - \boldsymbol{J}^\top(\boldsymbol{q})\gamma$$

Control law

$$\begin{aligned}\boldsymbol{\tau} &= \boldsymbol{B}(\boldsymbol{q})\boldsymbol{y} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}^\top(\boldsymbol{q})\gamma\\ \boldsymbol{y} &= \boldsymbol{J}_A^{-1}(\boldsymbol{q})\boldsymbol{M}_d^{-1}\left(\boldsymbol{M}_d\ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{D}_d\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_d\tilde{\boldsymbol{x}} - \boldsymbol{M}_d\dot{\boldsymbol{J}}_A(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \gamma_A\right)\end{aligned}$$

Closed loop

$$\boldsymbol{M}_d\ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{D}_d\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_d\tilde{\boldsymbol{x}} = \boldsymbol{\gamma}_A$$

8.2.2 Admittance control

Dynamics

$$\bar{\boldsymbol{x}} - \boldsymbol{x} \approx \frac{1}{\boldsymbol{M}_d s^2 + \boldsymbol{D}_d s + \boldsymbol{K}_d} \boldsymbol{f}_{ext}$$

Block diagram in Figure 6.1

n.b. requires a high bandwidth position controller

Index