ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

SCUOLA DI INGEGNERIA E ARCHITETTURA

DEI - DIPARTIMENTO DELL'INGEGNERIA ELETTRICA E DELL'INFORMAZIONE

Corso di Laurea in Ingegneria dell'Automazione

TESI DI LAUREA
in
CONTROLLI AUTOMATICI E TEORIA DEI SISTEMI M

Passive based control on a KUKA arm

Candidato:
MARCO LAGHI

Relatore:
Chiar.mo Prof.
LORENZO MARCONI

Correlatori:
Prof. STEFANO STRAMIGIOLI
Dott. MATTEO FUMAGALLI

Anno Accademico 2013/2014
Sessione III

A mio padre

Abstract

L'interazione in maniera sicura e compliante è una caratteristica sempre più richiesta per i sistemi robotici. La modellazione di sistemi eseguita tramite l'uso dei sistemi port-Hamiltoninani permette di comprendere cosa avviene a livello energetico durante l'interazione e aiuta nella progettazinoe di un controllore tale che il comportamento del sistema controllato sia passivo e sicuro durante essa. Ciò sfocia nel cosiddetto *Controllore Intrinsicamente Passivo* (IPC). Dal momento che questo controllo impone la rigidezza desiderata al sistema controllato, è possibile, tra le altre cose, replicare il comportamento del dispositivo RCC (Centro Remoto di Complianza) e di migliorarlo in modo tale che durante l'azione di *peg-in-hole* il buco sia meno sollecitato dal robot.

Interaction with the environment in a safety and compliant way is more often a capacity required to a robotic system. Modeling systems with port-Hamiltonian frameworks allows to understand what happens during this interaction in an energetic way and helps to design a controller such that the behaviour of the controlled system is passive and safety during interaction. This leads to the so called *Intrinsically Passive Controller* (IPC). Since this is a control that imposes a desired stiffness to the controlled plant, it is also possible to replicate the behavior of an RCC (Remote Center of Compliance) device and to improve it, such that during *peg-in-hole* tasks the hole is less stressed by the robot.

Acknowledgements

First of all, this thesis would not have been possible without the availability of Professor Lorenzo Marconi. Therefore, the first thanks is for him. On the other side, I want to thank Professor Stefano Stramigioli, to host me in the RaM group of the University of Twente during all the six months requested for this work. He didn't just host me and helped me understanding the theory on which this thesis is based, that he developed during his Ph.D, but he also treated me as a friend. A special thanks also to Matteo Fumagalli. With his suggestions, he always bring me in the right direction and his questions always pushed me deeply in the understanding of the work. I think I would not be able to finish the work without his supervision. A big thanks goes to all my tenants of Calslaan 24, Kevin, Harri, Vasilis, Xiang, Giulia, Elena, Taciana, that always tried to make me party and tolerate my incapacity to disconnect the mind from the study.

This thesis is the final step of the last two years of study. In these years I would not be able to work as hard as I did without the support of all my best friends Davide, Luca, Mattia, Michele and of a lot more (sorry if I don't write all your names, guys!!).

I want to dedicate the biggest thanks to my mother, her strength allowed me to continue and finish my studies without any worries but the university courses. An intimate thanks goes to my father, who deserves the credit for all the effort I had put to complete this Master Degree. I hope you are proud of me, dad.

Finally, the most deep and sincere thanks goes to my love, Elisabetta. She is always ready to comfort me during hard times and makes me laugh when I need it and, most important, she always believed in me. Without her smile and her support things would be surely harder.

Contents

Introduction

This work was suggested to me by Professor Lorenzo Marconi and Professor Stefano Stramigioli. The issue that I tried to solve is the interaction between a controlled system and surrounding environment and to verify the validity of the solution, applying it to the KUKA LWR 4+ robotic arm.
It is well known that interaction rises the real problem of which variable has to be controlled in order to achieve the desired behavior. Indeed, a velocity control is good in case of free motion, while in case of interaction (coupling with environment) maybe it would be better to control the force applied instead of the velocity. The interaction problem arises from this duality.
The most known technique to solve the problem is the one invented by Neville Hogan, called *impedance control*. Anyway, it is difficult to interpret in a physical way this type of control. An energetic approach is necessary, in order to model and interpret what really happens during interaction of two physical systems. The port-Hamiltonian framework allows us to have such an approach which, combined with bond-graph formalism (Appendix B), fully explains interconnection between systems. It turns out that systems are completely characterized by the energy that they have in a certain instant. Then, it has to be used a control technique based on this framework and approach, so that it works on the energy of the system that has to be controlled. Concerning this, the most important concept of the presented work is the one of *passivity*, always linked to energy. Developing and applying it to the controller, it is possible to ALWAYS achieve a desired passive behavior with respect to ANY passive environment. This leads to what Prof. Stramigioli calls *Intrinsically Passive Controller* (IPC) [11].

The path of this thesis is the following. First, an explanation of the so called **Skew Theory** is given in **Chapter 1**. This theory is necessary to understand the geometry that is behind robots. Some used geometric elements are defined and explained in **Appendix A**. It will be shown the link between the workspace of a robot and Lie Group, and all the things that follows from this. In **Chapter 2** Hamiltonian and **port-Hamiltonian system** are introduced. **Chapter 3** explains the **Passivity Theory** and the concept of **energy shaping**, from which **control by interconnection** and finally the **IPC** will follow. In **Chapter 4** the bond-graph model of the KUKA LWR 4+ is built and explained, as well as the **FRI** library, which is necessary to comunicate with the arm. In **Chapter 5** the implementation of the control technique found in Chapter 3 is applied to the KUKA LWR 4+ arm. Finally, in **Chaplter 6** an analysis of the replication of an RCC device is done, showing some improvement about the stress applied to the hole in the classic task **peg-in-hole**.

Chapter 1

Screw Theory

The Screw theory, developed by Sir Robert Stawell Ball in 1876, provides a mathematical formulation for the geometry of lines which is central to rigid body dynamics, where lines form the screw axes of spatial movement and the lines of action of forces. Since it is widely used in robotic applications, it is the base of this work. Then, in this chapter a full explanation of its main concepts and properties is given, starting from the basic geometrical definition of euclidean space. Furthermore, the affinity of each object to a particular vector space and algebra is given in order to understand the equivalence to Lie groups and algebras (Appendix A).

1.1 Euclidean Space

An *Euclidean Space* is a space of *free vectors* $\mathscr{E}$ with an extra structure, called *inner product*, that is a quadratic form and it defined as:

$$\langle , \rangle : \mathscr{E}_* \times \mathscr{E}_* \to \mathbb{R}; (v, \omega) \mapsto \langle v, \omega \rangle \tag{1.1}$$

With the inner product it is possible to define differents operations:

Definition 1.1.1. Norm (distance)

$$\| \cdot \| : \mathscr{E}_* \to \mathbb{R}; v \mapsto \sqrt{\langle v, v \rangle} \tag{1.2}$$

Definition 1.1.2. Orthogonality of $v, \omega \in \mathscr{E}_*$

$$v \perp \omega \iff \langle v, \omega \rangle = 0 \tag{1.3}$$

Definition 1.1.3. Angle bewteen vectors $v, \omega \in \mathcal{E}_*$

$$\cos v\angle\omega := \frac{\langle v, \omega\rangle}{\| v \| \cdot \| \omega \|} \tag{1.4}$$

The Euclidean Space of interest in this thesis is the 3-dimensional $\mathcal{E}(3)$, whose set of free vector is denoted as $\mathcal{E}_*(3)$.
It is convenient to identify a coordinates system (or *coordinates frame*, to uniquely determine the position of a point or other element on $\mathcal{E}(3)$:

$$\Psi_o := (0, e_1, e_2, e_3) \in \mathcal{E}(3) \times \mathcal{E}_*(3) \times \mathcal{E}_*(3) \times \mathcal{E}_*(3)$$

A coordinate system is *ortho-normal* iff

- $\| e_i \| = 1 \quad \forall i$ (unit vectors);
- $\langle e_i, e_j\rangle = 0 \quad \forall i \neq j$ (orthogonality);

Either for points and vectors, their coordinates are real number:

- For $p \in \mathcal{E}$ (point) $\quad x_i = \langle (p - o), e_i\rangle \quad i = 1, 2, 3$
- For $v \in \mathcal{E}_*$ (vector) $\quad x_i = \langle v, e_i\rangle \quad i = 1, 2, 3$

Usually the coordinates vectors are indicated as $\hat{x} := e_1$, $\hat{y} := e_2$, $\hat{z} := e_3$. Then it is possible to define a *coordinates mapping* on a coordinate system $\Psi_i = (o_i, \hat{x}_i, \hat{y}_i, \hat{z}_i)$ as:

$$\psi_i : \mathcal{E}(3) \to \mathbb{R}^3; p \mapsto \begin{pmatrix} \langle (p - o_i), \hat{x}_i\rangle \\ \langle (p - o_i), \hat{y}_i\rangle \\ \langle (p - o_i), \hat{z}_i\rangle \end{pmatrix} \tag{1.5}$$

Definition 1.1.4. Change of coordinates Given two coordinates frames Ψ_1 and Ψ_2 the following mapping is called *change of coordinates* between Ψ_1 and Ψ_2:

$$\mathbb{R}^3 \overset{\psi_1^{-1}}{\to} \mathcal{E}(3) \overset{\psi_2}{\to} \mathbb{R}^3; \quad p^1 \mapsto p \mapsto p^2 \tag{1.6}$$

that lead to an expression like:

$$p^2 = (\psi_2 o \psi_1^{-1})(p^1) \tag{1.7}$$

Definition 1.1.5. Vector product The *vector product* $\wedge$ is an operator defined as

$$\wedge : \mathscr{E}_*(3) \times \mathscr{E}_*(3) \rightarrow \mathscr{E}_*(3); (v, \omega) \mapsto v \wedge \omega \tag{1.8}$$

such that satisfies the following properties:

- $\langle (a \wedge b), a \rangle = 0 \quad \forall a, b \in \mathscr{E}_*(3);$
- $\langle (a \wedge b), b \rangle = 0 \quad \forall a, b \in \mathscr{E}_*(3);$
- $\| a \wedge b \| = \| a \| \| b \| \sin a \angle b.$

Defining the operator $\tilde{}$ such that:

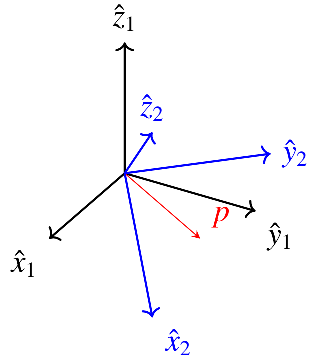
$$\tilde{} : \mathbb{R}^3 \rightarrow \mathbb{R}^{3\times3}; x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \mapsto y := \tilde{x} = \begin{pmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{pmatrix} \tag{1.9}$$

This equivalence hold:

$$v \wedge \omega = \tilde{v}\omega \tag{1.10}$$

1.2 3D Rotations

The point p of the figure to the right expressed in the two diffrent frames Ψ_1 and Ψ_2 is:



$$p^1 = \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = \psi_1(p) = \begin{pmatrix} \langle (p - o_1), \hat{x}_1 \rangle \\ \langle (p - o_1), \hat{y}_1 \rangle \\ \langle (p - o_1), \hat{z}_1 \rangle \end{pmatrix}; \quad p^2 = \begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} = \psi_2(p) = \begin{pmatrix} \langle (p - o_2), \hat{x}_2 \rangle \\ \langle (p - o_2), \hat{y}_2 \rangle \\ \langle (p - o_2), \hat{z}_2 \rangle \end{pmatrix} \tag{1.11}$$

ore equivalently $(p - o_i) = x_i\hat{x}_i + y_i\hat{y}_i + z_i\hat{z}_i$ with $x_i, y_i, z_i \in \mathbb{R}$ and $\hat{x}_i, \hat{y}_i, \hat{z}_i \in \mathscr{E}_*(3)$. Since $o_1 = o_2$, using the expression of $(p - o_1)$ in the ones of p^2, it turns out:

$$p^2 = \begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} = \psi_2(p) = \begin{pmatrix} \langle x_1\hat{x}_1 + y_1\hat{y}_1 + z_1\hat{z}_1, \hat{x}_2\rangle \\ \langle x_1\hat{x}_1 + y_1\hat{y}_1 + z_1\hat{z}_1, \hat{y}_2\rangle \\ \langle x_1\hat{x}_1 + y_1\hat{y}_1 + z_1\hat{z}_1, \hat{z}_2\rangle \end{pmatrix} = \begin{pmatrix} x_1\langle\hat{x}_1, \hat{x}_2\rangle + y_1\langle\hat{y}_1, \hat{x}_2\rangle + z_1\langle\hat{z}_1, \hat{x}_2\rangle \\ x_1\langle\hat{x}_1, \hat{y}_2\rangle + y_1\langle\hat{y}_1, \hat{y}_2\rangle + z_1\langle\hat{z}_1, \hat{y}_2\rangle \\ x_1\langle\hat{x}_1, \hat{z}_2\rangle + y_1\langle\hat{y}_1, \hat{z}_2\rangle + z_1\langle\hat{z}_1, \hat{z}_2\rangle \end{pmatrix} \tag{1.12}$$

Then a relation between the elements of p^2 and p^1 can be found:

$$p^2 = \begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} = \begin{pmatrix} \langle\hat{x}_1, \hat{x}_2\rangle & \langle\hat{y}_1, \hat{x}_2\rangle & \langle\hat{z}_1, \hat{x}_2\rangle \\ \langle\hat{x}_1, \hat{y}_2\rangle & \langle\hat{y}_1, \hat{y}_2\rangle & \langle\hat{z}_1, \hat{y}_2\rangle \\ \langle\hat{x}_1, \hat{z}_2\rangle & \langle\hat{y}_1, \hat{z}_2\rangle & \langle\hat{z}_1, \hat{z}_2\rangle \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = R_1^2 p^1 \tag{1.13}$$

where the matrix R_1^2 is a matrix that maps the coordinates of p in Ψ_1 in Ψ_2 and its called **rotation matrix**. Each rotation matrix satisfies the following proprieties:

- $\det(R_i^j) = 1$
- $R_j^i = (R_i^j)^{-1} = (R_i^j)^T$
- the columns and rows vector of R_i^j have length 1 and they are orthogonal

A square matrix $R \in \mathbb{R}^{3\times3}$ such that $R^{-1} = R^T$ is called *orthonormal*. The group of orthonormal matrices with determinant 1 is called *Special Orthonormal Group* of $\mathbb{R}^3$ and indicated as:

$$SO(3) := \{R \in \mathbb{R}^{3\times3}; \quad R^{-1} = R^T, \det R = 1\} \tag{1.14}$$

Furthermore, if $R(t) \in SO(3)$ is a differentiable function of time, $\dot{R}R^T$ and $R^T\dot{R}$ are skew-symmetric and belong to $so(3)$ defined as:

$$so(3) := \{\tilde{\omega} \in \mathbb{R}^{3\times3}; \quad -\tilde{\omega} = \tilde{\omega}^T\} \tag{1.15}$$

This means that $\exists \omega_1, \omega_2 \in \mathbb{R}^3$ such that $\tilde{\omega}_1 = \dot{R}R^t$ and $\tilde{\omega}_2 = R^T\dot{R}$. $\omega_i \in \mathbb{R}^3$ is the vector of the angular velocities around the three axes.
Is it possible to demonstrate that $so(3)$ is a Lie algebra.

Given the n rotation matrices $R_1^2, R_2^3, \ldots, R_{n-1}^n \in SO(3)$ it is possible to calculate the rotational matrix R_1^n using the "chain rule":

$$R_1^n = R_{n-1}^n R_{n-2}^{n-1} \dots R_2^3 R_1^2 \tag{1.16}$$

Then $SO(3)$ is a Group, since the associative property holds in it and it can be found an identity element (the identity matrix $I_{3\times3}$) with which the identity property and inverse property hold too.

Furthermore, it is a Lie group. Infact, being $(R, \dot{R}) \in TSO(3)$ and $\dot{R} \in T_R SO(3)$

- $(L_{R^{-1}})(R, \dot{R}) = (R^{-1}R, R^{-1}\dot{R}) = (I, \tilde{\omega}_L) \rightarrow \tilde{\omega}_L \in T_I SO(3) := so(3)$
- $(R_{R^{-1}})(R, \dot{R}) = (RR^{-1}, \dot{R}R^{-1}) = (I, \tilde{\omega}_R) \rightarrow \tilde{\omega}_R \in T_I SO(3) := so(3)$

This is quite important because it is then possible to talk about motion ω without knowing the pos R of an object.

Since $so(3)$ define the space of all the angular velocities ω its dual space $so^*(3)$ defines all the vectors $\tau \in \mathbb{R}^3$ that represent the torques. Furthermore, the dual product of ω and τ is:

$$\langle \tau | \omega \rangle := P \in \mathbb{R} \tag{1.17}$$

where P is a real number and its physical meaning is the Power.

1.3 General motion

If in the previous figure the origins of the two frames Ψ_1 and Ψ_2 do not coincide a term must be added in the Equation (1.12). This term is the vector o_1^2 representing the position of the origin of frame Ψ_1 with respect to the frame Ψ_2. Then, the mapping of the point p between the two frames leads to the following equation:

$$p^2 = R_1^2 p^1 + o_1^2 \tag{1.18}$$

The change of coordinates for general motor is:

$$\psi_j o \psi_i^{-1} : \mathbb{R}^3 \rightarrow \mathbb{R}^3; \quad p^i \mapsto R_i^j p^i + o_i^j \tag{1.19}$$

where $R_i^j \in SO(3)$ and $o_i^j \in \mathbb{R}^3$.

1.3.1 Homogeneous matrix

It would be useful to apply the ”chain rule” that holds between rotation matrices also in the case of general motion. To do so, it is first necessary to express the general point p in projective coordinates:

$$p := \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} \to P := \begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix} \tag{1.20}$$

Since $p^i \mapsto R_i^j p^i + o_i^j$, it follows that the change of coordinates for P is:

$$P^i \mapsto \begin{pmatrix} R_i^j & o_i^j \\ 0_{1\times 3} & 1 \end{pmatrix} P^i = H_i^j P^i \tag{1.21}$$

where H_i^j is called **Homogeneous transformation matrix**.
It is then possible to apply the chain rule such that:

$$H_1^n = H_{n-1}^n \dots H_2^3 H_1^2 \tag{1.22}$$

The inverse of an homogeneous matrix is:

$$H_2^1 = (H_1^2)^{-1} = \begin{pmatrix} (R_1^2)^T & -(R_1^2)^T o_1^2 \\ 0_{1\times 3} & 1 \end{pmatrix} \tag{1.23}$$

The set of all the homogeneous matrices is called **Special euclidean group** of order 3:

$$SE(3) := \left\{ \begin{pmatrix} R & o \\ 0_{1\times 3} & 1 \end{pmatrix} \text{ s.t } R \in SO(3), o \in \mathbb{R}^3 \right\} \tag{1.24}$$

Furthermore, $SE(3)$ is a Lie group as $SO(3)$ and if $H(t) \in SE(3)$ is a differentiable function of time $\dot{H}H^{-1}$ and $H^{-1}\dot{H}$ belong to $se(3)$ where:

$$se(3) := \left\{ \begin{pmatrix} \tilde{\omega} & v \\ 0 & 0 \end{pmatrix} \text{ s.t } \tilde{\omega} \in so(3), v \in \mathbb{R}^3 \right\} \tag{1.25}$$

$se(3)$ is a Lie algebra and its elements of $se(3)$ are called **twists** and usually are indicated a $\tilde{T}$. Furthermore, they identify a six dimensional vector T such that

$$T := \begin{pmatrix} \omega \\ v \end{pmatrix} \Longrightarrow \tilde{T} = \begin{pmatrix} \tilde{\omega} & v \\ 0 & 0 \end{pmatrix} \tag{1.26}$$

1.4 Twists

Twists are phisically interpreted as the generalization of velocities of a rigid body. Consider two bodies on which two frames i and j are attached. Consider then a third frame k. A twist describe the velocity of the frame (and so the body) i with respect to the frame j expressed in k and is written as:

$$T_i^{k,j}$$

Furthermore (Eq. 1.25),

$$\tilde{T}_i^{j,j} = \dot{H}_i^j H_j^i \tag{1.27}$$

$$\tilde{T}_i^{i,j} = H_j^i \dot{H}_i^j. \tag{1.28}$$

and then

$$\dot{H}_i^j = \tilde{T}_i^{j,j} H_i^j \tag{1.29}$$

$$\dot{H}_i^j = H_i^j \tilde{T}_i^{i,j} \tag{1.30}$$

that corresponds to the different choice of left or right translation. Consider a point p fixed in Ψ_i.

$$P^j = H_i^j P^i$$

$$\dot{P}^j = \dot{H}_i^j P^i$$

Then, depending on align (1.30):

$$\dot{P}^j = \tilde{T}_i^{j,j} H_i^j P^i \qquad \text{or} \qquad \dot{P}^j = H_i^j \tilde{T}_i^{i,j} P^i \tag{1.31}$$

1.4.1 Geometric interpretation of Twists

Theorem 1.4.1. ***Mozzi's theorem*** *Any rigid body motion can be expressed as a*

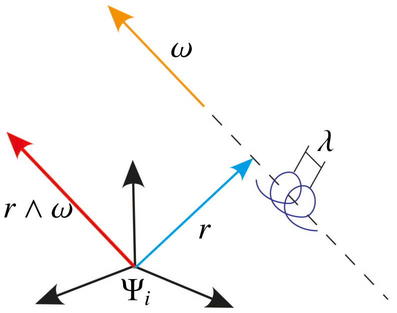


Figure 1.1. Skew representation of a twist

rotation around an axis and a translation along the same axis (Figure 1.1):

$$T := \underbrace{\begin{pmatrix} \omega \\ r \wedge \omega \end{pmatrix}}_{rotation} + \underbrace{\lambda \begin{pmatrix} 0 \\ \omega \end{pmatrix}}_{translation} \tag{1.32}$$

Furthermore, it is possible to calculate the values of r and λ starting from ω and v:

$$r = \frac{\omega \wedge v}{\| \omega \|^2} \qquad \text{and} \qquad \lambda = \frac{\omega^T v}{\| \omega \|^2} \tag{1.33}$$

1.4.2 Change of coordinates for Twists

From align (1.28):

$$\tilde{T}_i^{j,j} = H_i^j \tilde{T}_i^{i,j} H_j^i \tag{1.34}$$

$$\begin{pmatrix} \tilde{\omega}_i^{j,j} & v_i^{j,j} \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} R_i^j & p_i^j \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \tilde{\omega}_i^{i,j} & v_i^{i,j} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} R_j^i & -R_j^i p_j^i \\ 0 & 1 \end{pmatrix} \tag{1.35}$$

that is equal to:

$$\tilde{T}_i^{j,j} = \begin{pmatrix} \tilde{\omega}_i^{j,j} & v_i^{j,j} \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} \widetilde{(R_i^j \omega_i^{i,j})} & \tilde{p}_i^j R_i^j \omega_i^{i,j} + R_i^j v_i^{i,j} \\ 0 & 0 \end{pmatrix} = H_i^j \tilde{T}_i^{i,j} H_j^i \tag{1.36}$$

The same change of coordinate for twist in vector form is:

$$T_i^{j,j} = \begin{pmatrix} w_i^{j,j} \\ v_i^{j,j} \end{pmatrix} = \begin{pmatrix} R_i^j & 0 \\ \tilde{p}_i^j R_i^j & R_i^j \end{pmatrix} \begin{pmatrix} w_i^{i,j} \\ v_i^{i,j} \end{pmatrix} = Ad_{H_i^j} T_i^{i,j} \tag{1.37}$$

The matrix $Ad_{H_i^j}$ is called **Adjoin matrix of the homogeneous matrix H_i^j**.

1.4.3 Exponential map

From align (1.30):

$$\dot{H}_j^i = \tilde{T}_i^{j,j} H_j^i \tag{1.38}$$

If $\tilde{T}_i^{j,j}$ is constant, the solution of the previous differential align is:

$$H_i^j(t) = e^{\tilde{T}_i^{j,j} t} H_i^j(0) \tag{1.39}$$

So, this exponential map is:

$$e : se(3) \rightarrow SE(3); \qquad \tilde{T} \mapsto e^{\tilde{T}} \tag{1.40}$$

1.4.4 Direct Kinematic

A **unit twist** is a Twist in one of the two following forms:

$$\hat{T} = \begin{pmatrix} \hat{\omega} \\ \bullet \end{pmatrix} \qquad \text{or} \qquad \hat{T} = \begin{pmatrix} 0 \\ \hat{v} \end{pmatrix} \tag{1.41}$$

where $\hat{\omega}$ and $\hat{v}$ are unit vectors and are called **unit twists**.

A one degree of fredom kinematic pair or joint constrains the relative motion between two objects a, b with a unit twist.

$$T_a^{j,b} = \hat{T}_a^{j,b} q, \qquad q \in \mathbb{R} \tag{1.42}$$

where q is the joint position (linear or angular) and $\hat{T}_a^{j,b}$ is a constant unit twist if Ψ_j is fixed either in a or b. Then, the exponential map between two frames i and j is:

$$H_i^j(t) = e^{\tilde{\hat{T}}_i^{j,j} t} H_i^j(0) \tag{1.43}$$

$$H_i^j(q_j) = e^{\tilde{\hat{T}}_i^{j,j} q_j} H_i^j(0) \tag{1.44}$$

Consider now a robotic manipulator with a serial structure of n 1-dof joints and n links, each with a frame i attached on it. The chain rule says that the position of the end effector (link n) with respect to the inertial frame 0is given by:

$$H_n^0 = H_1^0 H_2^1 \dots H_n^{n-1} \tag{1.45}$$

Since all the joints are 1-dof kinematic pairs, using the previous aligns:

$$H_n^0(q_1, q_2, \dots, q_n) = \underbrace{e^{\tilde{\hat{T}}_1^{0,0} q_1} H_1^0(0)}_{H_1^0(q_1)} \underbrace{e^{\tilde{\hat{T}}_2^{1,1} q_2} H_2^1(0)}_{H_2^1(q_2)} \dots \underbrace{e^{\tilde{\hat{T}}_n^{n-1,n.1} q_n} H_n^{n-1}(0)}_{H_n^{n-1}(q_n)} \tag{1.46}$$

that after several calculus leads to the **Brockett's product of exponentials formula** (Direct kinematic):

$$H_n^0(q_1, q_2, \dots, q_n) = e^{\tilde{\hat{T}}_1^{0,0} q_1} e^{\tilde{\hat{T}}_2^{1,1} q_2} H_2^1(0) \dots e^{\tilde{\hat{T}}_n^{n-1,n.1} q_n} H_n^0(0) \tag{1.47}$$

1.5 Wrenches

The expression of Power P in the mechanical translational domain is:

$$P = Fv \in \mathbb{R} \tag{1.48}$$

where v is the column vector of linear velocities and F is the row vector of forces. In the skew theory a Twist a six dimensional column vector that is the generalization of velocities. Then a 6 six dimensional row vector representing the generalized forces is called Wrench:

$$W = \begin{pmatrix} \tau & F \end{pmatrix} \tag{1.49}$$

Where τ is a 3 dimensional row vector representing torques and F a 3 dimensional row vector representing forces. Then, the power is given by:

$$P = WT = \tau\omega + fv \tag{1.50}$$

In the last expression W and T must be expressed in the same coordinate system. To specify in which coordinate system a wrench is expressed the number of the frame is added as a superscript, like W^j. The tilde representation of a wrench is:

$$\tilde{W} = \begin{pmatrix} \tilde{f} & \tau^T \\ 0 & 0 \end{pmatrix} \tag{1.51}$$

Geometrically a wrench W is a linear map of a twist T in $\mathbb{R}$:

$$W : se(3) \to \mathbb{R}; \qquad W(T) = WT \tag{1.52}$$

This is (*Appendix A*) the same definition of the dual product. It follows that wrenches are the DUAL of twist and, then, $W \in se^*(3)$. Wrenches are co-vectors and not vectors.

1.5.1 Geometrical interpretation of Wrenches

Theorem 1.5.1. ***Pinsot's theorem*** *Any system of forces can be expressed as a pure linear force along a line plus a pure moment around it (Figure 1.2):*

$$W := \underbrace{\begin{pmatrix} r \wedge F \\ F \end{pmatrix}}_{force} + \underbrace{\lambda \begin{pmatrix} F \\ 0 \end{pmatrix}}_{moment} \tag{1.53}$$

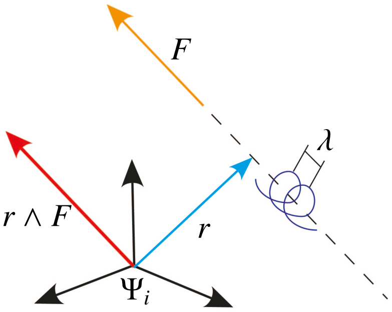


Figure 1.2. Skew representation of a wrench

1.5.2 Change of coordinates for Wrenches

The change of coordinate for twists is expressed in align 1.37. Furhtermore, the power continuity imposes:

$$W^j T_j^{j_i} = W^j Ad_{H_i^j} T_j^{i,i} = (Ad^T_{H_i^j} W^j)^T T_j^{i,i} = W^i T_j^{i,i} \tag{1.54}$$

That lead to the expression of the change of coordinates for wrenches:

$$(W^i)^T = Ad^T_{H_i^j} (W^j)^T \tag{1.55}$$

A summury of the change of coordinates for twists and wrenches is shown in Figure 1.3.

$$\begin{tikzcd}
\Psi_i \arrow[r, "H_i^j"] & \Psi_j \\
T_k^{i,l} \arrow[r, "Ad_{H_i^j}"] & T_k^{j,l} \\
W^i & W^j \arrow[l, "Ad^T_{H_i^j}"']
\end{tikzcd}$$

Figure 1.3. Change of coordinates between two frames

1.5.3 The Geometric Jacobian

As for the direct kinematic, it is really useful to know the end effector twist knowing the joint velocities and to know the joint torques starting from the wrench. Let's start with the first task. By definition:

$$\tilde{T}_n^{0,0} = \dot{H}_n^0 H_0^n \tag{1.56}$$

$$H_n^0 = H_1^0 H_2^1 \dots H_n^{n-1} \tag{1.57}$$

that leads to:

$$\begin{aligned}
\tilde{T}_n^{0,0} &= \dot{(H_1^0 H_1^2 \ldots H_n^{n-1})}(H_{n-1}^n H_{n-2}^{n-1} \ldots H_0^1) \\
&= \underbrace{\dot{H}_1^0 H_0^1}_{\tilde{T}_1^{0,0}} + H_1^0 \underbrace{\dot{H}_2^1 H_1^2}_{\tilde{T}_2^{1,1}} H_0^1 \ldots H_{n-1}^0 \underbrace{\dot{H}_n^{n-1} H_{n-1}^n}_{\tilde{T}_n^{n-1,n-1}} H_0^{n-1} \\
T_n^{0,0} &= T_1^{0,0} + Ad_{H_1^0} T_2^{1,1} + \ldots + Ad_{H_{n-1}^0} T_n^{n-1,n-1}
\end{aligned}$$

or in a more compact form

$$T_n^{0,0} = T_1^{0,0} + T_2^{0,1} + \ldots + T_n^{0,n-1} \tag{1.58}$$

This last align says that twists expressed in the same frame can be summed like scalars. A twist of a 1-dof joint is a unit twist times the velocity of the joint:

$$T_{i+1}^{i,i} = \hat{T}_{i+1}^{i,i} \dot{q}_{i+1} \tag{1.59}$$

Then, the align (1.58) can be expressed as:

$$T_n^{0,0} = J(q)\dot{q} \tag{1.60}$$

where

$$\begin{aligned}
J(q) &= \begin{pmatrix} T_1 & T_2 & \ldots & T_n \end{pmatrix} \\
T_i :&= Ad_{H_{i-1}^0} \hat{T}_i^{i-1,i-1} \\
q :&= (q_1, q_2, \ldots, q_n)^T
\end{aligned}$$

The matrix $J(q)$ is called **geometric jacobian** and is the map between the joint velocities space and the cartesian twist space. It follows also that:

$$\tau = (\tau_1, \tau_2, \ldots, \tau_n)^T = J^T(q)(W^n)^T \tag{1.61}$$

Chapter 2

Port-Hamiltonian Systems

The theory of port-Hamiltonian systems provides a framework for the geometric description of network models of physical systems. It turns out that port-based network models of physical systems immediately lend themselves to a Hamiltonian description. Then, it offers a systematic framework for analysis, control and simulation of complex physical systems. This chapter will briefly introduce this theory.

2.1 Port-Hamiltonian systems

A classical Hamiltonian system is a mathematical formalism to describe the evolution equations of a physical system. It is completely described by a scalar function $H(q, p)$ called *Hamiltonian* that represents the total energy of the system, where $q = (q_1, \dots, q_n)^T$ is the vector of the generalized coordinates for the mechanical system with n degrees of freedom and $p = (p_1, \dots, p_n)^T$ is the vector of the generalized momenta. The standard Hamiltonian equations are:

$$\begin{aligned} \dot{q} &= \frac{\partial H}{\partial p}(q, p) \\ \dot{p} &= -\frac{\partial H}{\partial q}(q, p) + F \end{aligned} \tag{2.1}$$

where the input F is the vector of the external generalized forces. The state space of Eq. (2.1) (q, p) is called *phase state*. The *energy balance* is immediatly derived:

$$\frac{d}{dt}H = \frac{\partial^T H}{\partial q}(q, p)\dot{q} + \frac{\partial^T H}{\partial p}(q, p)\dot{p} = \frac{\partial^T H}{\partial p}(q, p)F = \dot{q}^T F \tag{2.2}$$

that express the *conservation of energy* (the variation of the total energy is equal to

the work). Due to the energy balance, the output of the system is $f = \dot{q}$ (vector of generalized velocities).
A more general form of the system (2.1) is:

$$\begin{aligned}
\dot{q} &= \frac{\partial H}{\partial p}(q, p), \\
\dot{p} &= -\frac{\partial H}{\partial q}(q, p) + B(q)e, \qquad e \in \mathbb{R}^m \\
f &= B^T(q)\frac{\partial H}{\partial p}(q, p), \qquad\quad f \in \mathbb{R}^m
\end{aligned} \tag{2.3}$$

with $B(q)e$ denoting the generalized forces resulting from the input $e \in \mathbb{R}^m$. e and f are two dual variables. This means that they belongs to two dual vector spaces. Furthermore they represent the *power port* of the system.

Definition 2.1.1. Power Port A power port is defined as $P = \mathcal{F} \times \mathcal{E}$ where $\mathcal{F}$ is the flow vector space and $\mathcal{E} = \mathcal{F}^*$ is the effort vector space. Given $f \in \mathcal{F}$ and $e \in \mathcal{E}$, the product $\langle f, e \rangle$ is called the power traversing the power port.

If $m < n$ the system is called *underactuated*. Similary to the previous system, the power balance of system (2.3)

$$\frac{dH}{dt}(q, p) = e^T f \tag{2.4}$$

A further generalization is to consider the system described in local coordinates:

$$\begin{aligned}
\dot{x} &= J(x)\frac{\partial H}{\partial x}(x) + g(x)e, \qquad x \in \mathcal{X}, e \in \mathbb{R}^m \\
f &= g^T(x)\frac{\partial H}{\partial x}(x), \qquad f \in \mathbb{R}^m
\end{aligned} \tag{2.5}$$

where $J(x)$ is an $n \times n$ matrix with entries depending on x, which is assumed to be *skew-symmetric*, and $x = (x_1, \ldots, x_n)$ are the local n-dimensional state-space manifold $\mathcal{X}$. Because of skew-symmetry of J the energy balance $\frac{dH}{dt}(x) = e^T f$ holds. The system (2.5) is a ***port-Hamiltonian system*** with ***structure matrix*** J, input matrix $g(x)$ and *Hamiltonian* H.

2.2 Dirac Structure

The dynamics of a system are determined by the storage elements in bond-graph representation as well as the resistive elements, while the geometric structure results from the generalized junction structure of the bond-graph. These interconnections usually give rise to algebraic constraints between the state space variables of the sub-systems; thus leading to implicit systems. Therefore it is important to extend the framework of port-Hamiltonian systems to the context of *implicit systems*, that are systems with algebraic constraints. The formalization of the geometric function of the generalized junction as a *Dirac structure* is the key to define an *implicit* port-Hamiltonian system, together with the formalization of the concept of a power-conserving interconnection and the generalization of the notion of a structure matrix $J(x)$ as encountered before.

Consider a vector space $\mathcal{F}$ and its dual $\mathcal{F}^*$ and the dual product (Def. A.1.10, Sec. A.1).

Definition 2.2.1. Dirac Structure A *Dirac structure* on $\mathcal{F}$ is a subspace $\mathcal{D} \subset \mathcal{F} \times \mathcal{F}^*$ such that

- $\langle e|f\rangle = 0$, for all $(f, e) \in \mathcal{D}$,
- $\dim \mathcal{D} = \dim \mathcal{F}$

Where the first property correspond to the *power-conservation* and expresses the fact that the total power entering (or leaving) a Dirac structure is zero.

Definition 2.2.2. Bi-linear form Related to the definition of power, there exist a canonically defined *bi-linear form* $\ll \cdot, \cdot \gg$ on the space $\mathcal{F} \times \mathcal{F}^*$, defined as:

$$\ll (f^a, e^a), (f^b, e^b) \gg := \langle e^a|f^b\rangle + \langle e^b|f^a\rangle \tag{2.6}$$

with $(f^a, e^a), (f^b, e^b) \in \mathcal{F} \times \mathcal{F}^*$.

The bi-linear form is *indefinite* ($\ll (f^a, e^a), (f^b, e^b) \gg$ can be positive or negative) but is not *non-degenerate* ($\ll (f^a, e^a), (f^b, e^b) \gg = 0$ for all (f^a, e^a) implies that $(f^b, e^b) = 0$. A fundamental definition follows:

Definition 2.2.3. Constant Dirac structure A *constant* Dirac structure on $\mathcal{F}$ is a subspace $\mathcal{D} \subset \mathcal{F} \times \mathcal{F}^*$ such that

$$\mathcal{D} = \mathcal{D}^{\perp} \tag{2.7}$$

where $\perp$ denotes the orthogonal complement with respect to the bi-linear form $\ll \cdot, \cdot \gg$.

From this last definition it follows that the dimension of any Dirac structure $\mathcal{D}$ on an n-dimensional linear space is equal to n. Furthermore, considering $(f, e) \in \mathcal{D}$, by Definition (2.6)

$$0 = \langle (f, e), (f, e) \rangle_{\mathcal{F}\times\mathcal{F}^*} = 2\langle e|f \rangle \tag{2.8}$$

Thus for all $(f, e) \in \mathcal{D}$. It follows that a Dirac structure $\mathcal{D}$ on $\mathcal{F}$ defines a power-conserving relation between the power variables $(f, e) \in \mathcal{F} \times \mathcal{F}^*$, which moreover has maximal dimension.

For systems which have 3-D mechanical components (that are of interest in this thesis), the Dirac structure is *modulated* by the geometric variables: the state space $SO(3)$ is a manifold and the flow $f_S = -\dot{x}$ correspondig to the energy-storage are elements of the tangent space $T_x SO(3)$ at the state $x \in SO(3)$, while the efforts e_S are elements of the co-tangent space $T_x^* SO(3)$. This Dirac structures often arises as a result of *kinematic constraints*. Since often this constraints are configuration dependent the Dirac structure become modulated by the configuration variables. Note that bond-graph's junction elements correspond to Dirac structures.

2.3 Implicit port-Hamiltonian systems

The general representation of a port-Hamiltonian system is shown in Figure 2.1. The port entering the Dirac structure are divided in two parts. There are two *internal* ports, corresponding to the energy-storage port (denoted as $\mathcal{S}$) and the energy-dissipation port (denoted as $\mathcal{R}$), and two *external* ports, that are, the C port, accessible for the control action, and the $\mathcal{I}$ port that represent the interaction of the sistem with the environment.

2.3.1 Energy storage port

The port variables of this port are denoted as (f_S, e_S). They interconnect the Dirac sturcture with the energy storage element defined by a finite dimensional manifold $\mathcal{X}$ with coordinates x and an Hamilotinan function $H : \mathcal{X} \to \mathbb{R}$ that represent the energy of the element. The flow variables of this element are the rate $\dot{x}$ of the energy variables x. Then, the energy variables of the element are co-energy variables $\frac{\partial H}{\partial x}(x)$,

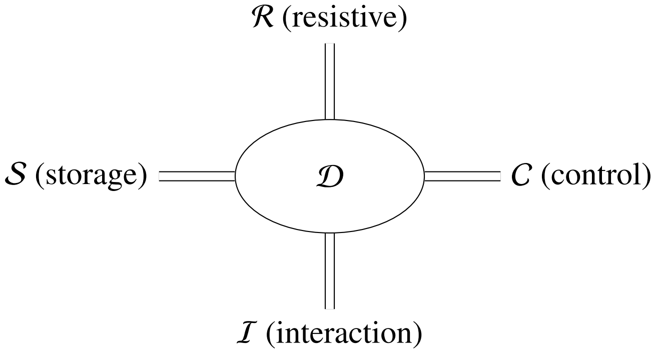


Figure 2.1. Implicit port-Hamiltonian System

given by the energy balance:

$$\frac{d}{dt}H = \left\langle \frac{\partial H}{\partial x}(x) | \dot{x} \right\rangle = \frac{\partial^T H}{\partial x}(x)\dot{x} \tag{2.9}$$

Setting

$$\begin{aligned} f_S &= -\dot{x} \\ e_S &= \frac{\partial H}{\partial x}(x) \end{aligned} \tag{2.10}$$

(that corresponds to a 0-junction) the interconnection between the energy storage element and the Dirac structure is done. Hence the energy balance of Eq. 2.9 can also be written as:

$$\frac{d}{dt}H = \frac{\partial^T H}{\partial x}(x)\dot{x} = -e_S^T f_S \tag{2.11}$$

2.3.2 Resistive port

This port includes all the energy dissipation effects of the system and its port variables are (f_R, e_R). These port variables are determined on a static resistive relation R, generally of the form:

$$R(f_R, e_R) = 0, \tag{2.12}$$

with the property that for all (f_R, e_R)

$$\langle e_R | f_R \rangle \leq 0 \tag{2.13}$$

In many cases the resistive relation is *linear*. This means that the port variables has to satisfy a linear relation of the form:

$$R_f f_R + R_e e_R = 0. \tag{2.14}$$

If there are no external port, the Dirac structure of the port-Hamiltonian system satisfy the relation $e_S^T f_S \; + e_R^T f_R = 0$ that leads to

$$\frac{d}{dt} H = -e_S^T f_S \; = e_R^T f_R \leq 0. \tag{2.15}$$

2.3.3 External ports

The variables of the external port C are denoted as (f_C, e_C) while the ones of the external port $\mathcal{I}$ are denoted as (f_I, e_I). Considering also these two ports, the power balance is:

$$e_S^T f_S \; + e_R^T f_R + e_C^T f_C + e_I^T f_I = 0. \tag{2.16}$$

where $e_S^T f_S$ can be substitutes with $-\frac{d}{dt} H$ and then:

$$\frac{d}{dt} H = e_R^T f_R + e_C^T f_C + e_I^T f_I \tag{2.17}$$

2.3.4 port-Hamiltonian dynamics

THe port-Hamiltonian system with state space $\mathcal{X}$, hamiltonian H and corresponding to the structure just explained (comprehensive of the Dirac structure $\mathcal{D}$ and the power ports $\mathcal{S}, \mathcal{R}, C, \mathcal{I}$ is denoted by $\Sigma = (\mathcal{X}, H, \mathcal{R}, C, \mathcal{I}, \mathcal{D})$.
The dynamic of this port-Hamiltonian system is given considering the constraints imposed to the port variables by the Dirac structure

$$(f_S \, , e_S \, , f_R, e_R, f_C, e_C, f_I, e_I,) \in \mathcal{D} \tag{2.18}$$

and substituting in these relation the equations (2.10). This lead to the *implicit* dynamics:

$$(-\dot{x}(t), \frac{\partial H}{\partial x}(x(t)), f_R(t), e_R(t), f_C(t), e_C(t), f_I(t), e_I(t)) \in \mathcal{D} \tag{2.19}$$

with $f_R(t)$ and $e_R(t)$ satisfying the Equation (2.12).

2.4 Explicit port-Hamiltonian systems

Definition 2.4.1. Input-state-output dynamical system An input-state-output dynamical system is defined as a 5-tuple $\Sigma = (\mathbb{R}, \mathcal{U}, \mathcal{Y}, \mathcal{X}, \mathcal{B}_f)$ where $\mathbb{R}$ represent the time axis, $\mathcal{U}$ is the input signal space, $\mathcal{Y}$ is the output signal space, $\mathcal{X}$ the state space and $\mathcal{B}_f \subseteq (\mathcal{U} \times \mathcal{Y}, \mathcal{X}^{\mathbb{R}})$ the full behavior.

An *explicit port-Hamiltonian systems* is an input-state-output continuous time dynamical system where:

- $\mathcal{X}$ is an n dimensional manifold representing the state space; the states are energy variables;
- $\mathcal{U}$ is the input vector space and the input is a power conjugate variable;
- $\mathcal{Y} = \mathcal{U}$ is the output vector space and the output is the power variable dual to the input;
- $\mathcal{B}_f$ is the full behavior of the system represented by:

$$\begin{aligned} \dot{x} &= [J(x) - R(x)]\frac{\partial H}{\partial x}(x) + g(x)u, \\ y &= g^T(x)\frac{\partial H}{\partial x}(x) \end{aligned} \tag{2.20}$$

where u,y are the variables of the port C. $J(x)$ is a skew-symmetric matrix and $R(x) = R^T(x) \geq 0$ is the resistive structure resulting from $R(x) = g_R^T(x)\tilde{R}g_R(x)$ for some linear relation $f_R = -\tilde{R}e_R$, with g_R the input matrix of the resistive port. The underling Dirac structure of the system is given by the graph of the skew-symmetric linear map

$$\begin{pmatrix} -J(x) & -g_R(x) & -g(x) \\ g_R(x) & 0 & 0 \\ g(x) & 0 & 0 \end{pmatrix}. \tag{2.21}$$

Chapter 3

Intrinsically Passive Control

As shown in Chapter 2, a system represented in an Hamiltonian form is mostly characterized by its Hamiltonian function, that is, the total energy stored in the system. Furthermore, its dynamics depends on it as well as the dissipation and the inputs of the system. Then, energy plays a central role in the control of a physical system, since the stability of the system depends on the shape of the energy. In fact, any configuration characterized by a (local) *minimum* of the energy has (local) stable behavior. The problem arises when the configuration that present the minimum is not the desired one.

One of the most used technique in control of physical system is the called *energy shaping*. In this strategy the controller is saw as a device that *exchange* energy with the plant in such a way that the controlled system can be still interpreted as a dynamical system but with an energy function that as his minimum in the desired configuration. The design of such a controller using the port-Hamiltonian formalism leads to a precise explaination of all the energetic properties of the system that can be used for the regulation purpose.

The link between stability and energetic properties of a system is formalized by means of the *passivity theory*. This theory allows the so called *Passive Based Controls* (PCB) and the *Intrinsically Passive Control* (IPC). In this chapter these topics are presented, starting with the passivity theory related to port-Hamiltonian systems and energy-shaping technique and than leading to PCB.

3.1 Passivity Theory

Consider an I-S-O dynamical system (Def. 2.4.1) of the form:

$$\begin{cases} \dot{x} = f(x) + g(x)u \\ y = h(x) \end{cases} \tag{3.1}$$

where $x \in \mathcal{X}$ is the state variable and $\mathcal{X}$ is the state space, $u \in \mathcal{U}$ is the input and $\mathcal{U}$ is an m-dimensional input space and $y \in \mathcal{Y}$ is the output and $\mathcal{Y}$ is an m-dimensional output space. Let w be a real value map defined on $\mathcal{U} \times \mathcal{Y}$

$$w : \mathcal{U} \times \mathcal{Y} \rightarrow \mathbb{R}; \tag{3.2}$$

called *supply rate* and assume that the system has at least one equilibrium configuration in $x = 0$.

Definition 3.1.1. Dissipative system Let $x_0 \in \mathcal{X}$ be the initial state, that is $x_0 = x(t = 0)$. A system of the form (Eq. 3.1) is said to be *dissipative* with respect to the supply rate w if, for all $u \in \mathcal{U}, x_0 \in \mathcal{X}$ and $t \geq 0$ exists a continuous function $V : \mathcal{X} \rightarrow \mathbb{R}^+$ such that the following equation, called *dissipation inequality*, holds:

$$V(x(t)) - V(x_0) \leq \int_0^t w(\tau)d\tau \tag{3.3}$$

It is possible to give a physical and energetic interpretation of dissipative system considering V as the generalized energy of the system and w the generalized power. Then, the dissipation inequality express the statement that a system is dissipative if and only if the stored generalized energy at time t, $V(x(t))$, is at most equal to the sum of the initially stored energy $V(x_0)$ and the total external supplied energy to the system $\int_0^t w(\tau)d\tau$ in the interval $[0, t]$. This means that there cannot be any internal production of energy and the pair (u, y) represents the medium through which the system can exchange energy.
If the generalized energy of a dissipative system coincides with its energy, its generalized power coincides with its power and $\mathcal{U}$ and $\mathcal{Y}$ are m-dimensional dual spaces then the following choice of the supply rate can be made:

$$w(u(t), y(t)) = \langle u(t), y(t) \rangle = y^T u \qquad u \in \mathcal{U}, y \in \mathcal{Y} \tag{3.4}$$

Thus, the supply rate represent the power flow, u and y are *flow* and *effort* and the pair (u, y) is a *power port*.

Definition 3.1.2. Passive system A system is *passive* if it is dissipative with respect to the supply rate $w(u, y) = \langle u, y\rangle = y^T u$.

Then, the dissipation equality for a passive system is always

$$V(x(t)) - V(x_0) \leq \int_0^t y^T(\tau)u(\tau)d\tau \tag{3.5}$$

So, if $u = 0$ it follows that:

$$\dot{V}(x(t)) \leq y^T(t)u(t) \leq 0 \qquad \forall t \tag{3.6}$$

and if $V(x)$ is positive definite, the equilibrium point $x = 0$ is Lyapunov stable.

Definition 3.1.3. Lossless system A passive system with storage function $V(x)$ is *lossless* if $\forall u \in \mathcal{U}, x_0 \in \mathcal{X}$ and $t \geq 0$

$$V(x(t)) - V(x_0) = \int_0^t y^T(\tau)u(\tau)d\tau \tag{3.7}$$

This means that a lossless system stores all the energy provided through the power port.

Definition 3.1.4. Strictly passive system A *strictly passive* system is a passive system with storage function $V(x)$ for which there exists a positive definite function $S : \mathcal{X} \to \mathbb{R}^+$ such that $\forall \quad u \in \mathcal{U}, x_0 \in \mathcal{X}$ and $t \geq 0$

$$V(x(t)) - V(x_0) = \int_0^t y^T(\tau)u(\tau)d\tau - \int_0^t S(x(\tau)d\tau \tag{3.8}$$

A strictly passive system dissipates part of the energy provided through the power port. The energy dissipated in an interval $[0, t]$ is equal to $\int_0^t S(x(\tau)d\tau$.
Referring to the the general system of Eq. (3.1), it is possible to verify its passivity by means of the non-linear version of Kalman-Yakubovitch-Popov lemma (KPY). Infact, if there exist a function $V : \mathcal{X} \to \mathbb{R}^+$ with $V(0) = 0$ such that

$$L_f V(x) \leq 0 \tag{3.9a}$$

$$L_g V(x) = h^T(x) \tag{3.9b}$$

then the system is passive with respect to the storage function $V(x)$. In the previous equations L_k denotes the Lie derivative with respect to a function k.
Furthermore, $L_f V(x)$ represents the dissipation of the system and it is null if the system is lossless, while $L_g V(x)$ represents the power supplied to the system.
Then it is possible to give another definition of passive system

Definition 3.1.5. For a passive system the equation

$$P = \frac{dV}{dt} + P_{diss} \tag{3.10}$$

where $P = L_g V(x) = y^T u$ is the power supplied to the system, $V(x)$ is the storage function and $P_{diss} = -L_f V(x) > 0$ is the dissipated power.

It is possible to use definitions and observations given above to undestand if a port-Hamiltonian system is passive.
Consider the generic port-Hamiltonian system with dissipation

$$\begin{aligned} \dot{x} &= [J(x) - R(x)]\frac{\partial H}{\partial x}(x) + g(x)u, \\ y &= g^T(x)\frac{\partial H}{\partial x}(x) \end{aligned} \tag{3.11}$$

It is possible to relate port-Hamiltonian systems and passivity by means of the following proposition:

Proposition 3.1.1. *A port-Hamiltonian system with dissipation is passive and its storage function is the Hamiltonian function.*

Proof. It is possible to relate a port-Hamiltonian system of Eq. (3.11) to the affine I-S-O system described by Eq. (3.1) defining the following relations:

- $f(x) = (J(x) - R(x))\frac{\partial H}{\partial x}$;
- $g(x) = g(x)$;
- $h(x) = g^T(x)\frac{\partial H}{\partial x}$.

So, it turns out that:

$$L_f H(x) = L_{(J(x)-R(x))\frac{\partial H}{\partial x}} H(x) = \frac{\partial^T H}{\partial x}(J(x) - R(x))\frac{\partial H}{\partial x} = -\frac{\partial^T H}{\partial x}R(x))\frac{\partial H}{\partial x} \leq 0 \tag{3.12a}$$

$$L_g H(x) = \frac{\partial^T H}{\partial x} g(x) = \left(g^T(x)\frac{\partial H}{\partial x}\right)^T = h^T(x) \tag{3.12b}$$

where the term with $J(x)$ in the first row disappears since $J(x)$ is skew-symmetric and the inequality holds since $R(x)$ semi-positive definite. This means the a port-Hamiltonian system with dissipation enjoys the KPY property and consequently it is passive. Furthermore, a port-Hamiltonian system with dissipation is strictly passive or lossless depending on the dissipation term $R(x)$, as shown in Eq. (3.12a). In particular, if $R(x) = 0$

$$L_{(J(x)-R(x))\frac{\partial H}{\partial x}} H(x) = 0 \tag{3.13}$$

and then the system is lossless, otherwise, since $R(x)$ is semi-positive definite, it is strictly passive.
It is possible also to rewrite Def. 3.1.5 for port-Hamiltonian system, that is

$$P = \frac{dH}{dt} + \frac{\partial^T H}{\partial x} R(x)) \frac{\partial H}{\partial x} \tag{3.14}$$

from which it follows that $P_{diss} = \frac{\partial^T H}{\partial x} R(x)) \frac{\partial H}{\partial x}$. So, the dissipated power depends on the $R(x)$. If $R(x)$ is positive definite $P_{diss} > 0$ that means that some power is dissipated by the system and, then, the system is strictly passive. If $R(x) = 0$ also $P_{diss} = 0$ and so the system is lossless. Finally, if $R(x)$ is definite negative, $P_{diss} < 0$ which means that the system produces energy internally and it is not passive.

As just shown, port-Hamiltonian system have all the properties of passive system. This means that it is possible to stabilize in an asymptotic way an equilibrium configuration corresponding to the local minimum point of the associated Hamiltonian function by means of the control low $u = -ky$. The evolution of the state of a system controlled in such a way is described by the equation:

$$\begin{aligned} \dot{x} &= [J(x) - R(x)]\frac{\partial H}{\partial x} - g(x)u = \\ &= [J(x) - R(x)]\frac{\partial H}{\partial x} - kg(x)y = \\ &= [J(x) - R(x)]\frac{\partial H}{\partial x} - kg(x)g^T(x)\frac{\partial H}{\partial x} \\ \dot{x} &= [J(x) - (R(x) + kg(x)g^T(x))]\frac{\partial H}{\partial x} \end{aligned} \tag{3.15}$$

So, this kind of control act on the system as an added power dissipation correspondig to $kg(x)g^T(x)$. This motivates the name of the control, called *stabilization by damping injection.*

3.2 Energy Shaping

As just saw it is possible to stabilize the configuration of the system corresponding to the scrict minimum of the energy function by means of damping injection. Unfortunately the cases in which this configuration correspond to the desired one are rares. It follows that the used control has to act on the system in such a way that the minimumm of the energy function slides on the desired one. This kind of action is called *energy shaping*. Furthermore, it is possible to asymptotically stabilize the new minimum by means of damping injection. The resulting control law is named *energy shaping + damping injection.*

To better undestand how this action acts on the system, here below Eq. (3.8) is written for a port-Hamiltonian system:

$$\begin{aligned} H(x(t)) - H(x(0)) &= \int_0^t u^T(\tau)y(\tau)d\tau - \int_0^t \frac{\partial^T H}{\partial x} R(x) \frac{\partial H}{\partial x} d\tau = \\ &= \int_0^t u^T(\tau)y(\tau)d\tau - d(t) \end{aligned} \tag{3.16}$$

where, obviously, $d(t)$ is a non negative function representing the energy naturally dissipated into the system in the interval $[0, t]$.
The chosen control law $u(t)$ has to act on the system such that the closed loop dynamics satisfy a new energy balance equation

$$H_d(x(t)) - H_d(x(0)) = \int_0^t v^T(\tau)y(\tau)d\tau - d_d(t) \tag{3.17}$$

where H_d is a new energy function with a strict minimum in x^*, that is the desired configuration, and $d_d(t)$ is the desierd energy dissipation assigned by damping injection. A formal solution to this problem follows:

Proposition 3.2.1. *If it is possible to find a function $\beta(x)$ such that:*

$$-\int_0^t \beta^T(x(\tau))y(\tau)d\tau = H_a(x(t)) + k \tag{3.18}$$

where $k \in \mathbb{R}^+$ is constant, the the control law $u = \beta(x) + v$ is such that the energy balance

$$H_d(x(t)) - H_d(x(0)) = \int_0^t v^T(\tau)y(\tau)d\tau - d(t) \tag{3.19}$$

is satisfied with $H_d(x) = H(x) + H_a(x)$.

Proof. By replacing $u = \beta(x) + v$ in Eq. (3.16):

$$H(x(t)) - H(x(0)) = \int_0^t \beta^T(x(\tau))y(\tau)d\tau + \int_0^t v^T(\tau)y(\tau)d\tau - d(t) \tag{3.20}$$

Then, substituting Eq. (3.18) in Eq. (3.20), it follows that:

$$H(x(t)) + H_a(x(t)) - H(x(0)) + k = \int_0^t v^T(\tau)y(\tau)d\tau - d(t) \tag{3.21}$$

From Eq. (3.18) it has to be $H_a(x(0)) = -k$ and then the previous equation can be rewritten as:

$$H(x(t)) + H_a(x(t)) - H(x(0)) - H_a(x(0)) = \int_0^t v^T(\tau)y(\tau)d\tau - d(t) \tag{3.22}$$

So, setting $H_d(x(t)) = H(x(t)) + H_a(x(t))$, Eq. (3.17) holds.

It can be seen that the closed loop energy is different than the one stored by the plant and the one supplied by the controller. This kind of control is called *energy balancing passivity based control* (or, shorter, *energy balancing PCB*).
Furthermore, since the control law is built in such a way that $H_d(x)$ has a minimum in the desired configuration, setting $v = 0$, the system will asymptotically reach x^*.
As just shown, the energy balancing PCB changes the energy of the plant by giving it the desired shape. This action cab be interpreted in a physical way. To do this a more geometric interpretation is necessary and it is given by che concept of *control as inteconnection*.

3.3 The Control as Interconnection

The control as interconnection is a new way of looking at control and control problem than the classical *signal processors*. In particular it consist in considering the controller as the simulation of a physical system, with well defined properties, connected to the plant to be controlled. Then, the control issues concern in the design of a system such that its interaction with the plant leads to the desired behavior of the latter. In this section this control paradigm will be described for dynamical system, which is the framework needed for the control of port-Hamiltonian system.

Definition 3.3.1. Interconnection of dynamical systems The interconnection of two dynamical systems $\Sigma_1 = (\mathbb{R}, \mathcal{X}, \mathcal{B}_1)$ and $\Sigma_2 = (\mathbb{R}, \mathcal{X}, \mathcal{B}_2)$ with the same signal space $\mathcal{X}$ is denoted as $\Sigma_1 \wedge \Sigma_2$ and is defined as $\Sigma_1 \wedge \Sigma_2 = (\mathbb{R}, \mathcal{X}, \mathcal{B}_1 \cap \mathcal{B}_2)$

This means that, since the two systems have the same signal space, the behavior resulting by the interconnection consists in all those trajectories $w : \mathbb{R} \to \mathcal{W}$ that are allowed for both Σ_1 and Σ_2. So, the interconnection limits the behavior of both the interconnected system.
Let $\Sigma_p = (\mathbb{R}, \mathcal{X}, \mathcal{B}_p)$ be the plant to be controlled. The set of all dynamical systems with $\mathbb{R}$ as time axis and $\mathcal{W}$ as signal space is called *family of admissible controllers* and is denoted as $\mathfrak{C}$, while an element $\Sigma_c \in \mathfrak{C}$ is called *admissible controller*. Consequently, $\Sigma_p \wedge \Sigma_c$ is called *controlled system.*
With this approach the controller is no more saw as a signal processor but as a dynamical system and it is possible to easily intepreted as a physical system. It follows that the problem on control concerns in three steps:

- Description of the family of admissible controller $\mathfrak{C}$
- Description of the desired behavior of the plant
- Design of a controller $\Sigma_c \in \mathfrak{C}$ such that $\Sigma_p \wedge \Sigma_c$ has the desired behavior.

Since port-Hamiltonian systems interact with the external environment through power ports, also the interconnection between port-Hamiltonian systems take place through respective power ports. It is useful to interconnect the two systems (the palnt and the controller) by means of *power preserving interconnection*, that allows to establish an energy transfer between the power port interconnected (Fig. 3.1).

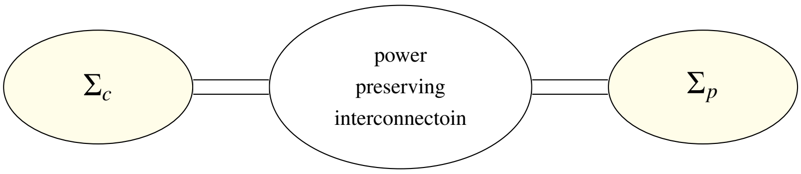


Figure 3.1. Power preserving interconnection

The most general way to represent a power preserving interconnection is through a Dirac sturcture (Sec. 2.2) but it is more useful to do it by means of a skew-symmetric matrix $J_{int}(x_p, x_c)$, where x_p is the state of the plant and x_c the one of the controller. The relation between inputs and outputs of the two systems is then:

$$\begin{pmatrix} u_p \\ u_c \end{pmatrix} = J_{int}(x_p, x_c) \begin{pmatrix} y_p \\ y_c \end{pmatrix} \tag{3.23}$$

In this way it is explicited that the energy extracted from one system and supplied to the other one is trasfered without losing or producing extra energy. An important result follows:

Proposition 3.3.1. *The power preserving interconnection between two port-Hamiltonian systems Σ_1 adn Σ_2 yelds to another port-Hamiltonian system with state space given by the product $\mathcal{X}_1 \times \mathcal{X}_2$ and with Hamiltonian function $H_1(x_1) + H_2(x_2)$.*

3.4 Control of Interaction

A robot has often to interact with the environment in order to accomplish a task. During the interaction deep changes occour. Infact, before the contact with the environment only the motion of the robot has to be controlled, while after it the whole dynamic model needs to be changed so that the coupling of the robot and the environment is taken into account. The main problem is that, even though the controlled robot during free motion is stable, instability can rise during interaction with the environment. During the contact the interacting systems influence each other reciprocally. This contact takes place through localized ports and the ”information” exchange affects forces and velocity. Therefore, the issue of the designer is to undestand which kind of strategy has to be used in order to control the interaction. Indeed, neither the force nor the velocity controls would be proper, as they both depend on the dynamics of the controlled system, which is known, and on the dynamics of the environment, that is often mostly unknown.

The most used and successful control for this tasks is the well-known *impedance control* that acts on the system by creating a *dynamical* relation between the applied force and the velocity of the end effector. Then, the purpose of this control is to define the behavior of the plant during the interaction. Impedance control is the starting point of the design of a port-Hamiltonian based *intrinsically passive controller* (IPC) of interaction.

Indeed, as just said above, the interaction between a robot and the environment can be modeled as an exchange of forces and velocities between the two systems.

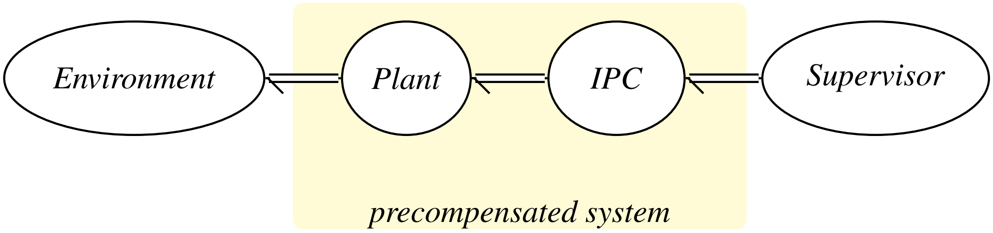


Figure 3.2. Scheme of Intrinsically Passive Control

This interaction can be easily modeled in a port-Hamiltonian framework by means of a localized ports, that on the robot side it is usually located at its end-effector. Then, since the only thing that can be intrinsically controlled is the relation between forces and velocities characteristic of this power port, it is necessary to design a controller able to modify the *behavior* of the whole system with respect to port, instead of regulating just one variable. The goal of the controller is to regulate the relationship between the two dual variables through the *interconnection port* and this can be reached using the *control by interconnection* technique (Sec. 3.3). In this way it is possible to regulate the plant by interconnecting to it another dynamical system, that is, the controller, in order to constrain the behavior of the plant at the power port through which the interconnection takes place, constraining the relative variables to a desired subset. So, this constrained relation is reached indipendentely from the environment the system interacts with.

It is naturally desirable to maintain a passive behavior also during the interaction, while controlling the way the plant interacts with environment. In order to maintain passivity, it is useful to connect to the plant a passive controller through power preserving interconnection. In this way the controlled plant preserves passivity with ANY possible passive environment. This is what it's called *Intrinsically Passive Controller* (IPC) and it is modelled as a physical system that compensates some unwanted properties of the controlled system and sets the desired ones with respect to the interaction. Anyway, such a controller doesn't allow to perform any task. In order to allow task accomplishment, a power port has to be added to the IPC. Then, an external supervisory system can inject energy through this port in the controlled system in a proper way such that a desired task is accomplished. The general scheme of the IPC is shown in Figure 3.2.

Chapter 4

The KUKA LWR 4+

In this chapter an analysis of the KUKA Lightweight Robot 4+ is done. First the kinematic proprieties and the construction of the bond-graph model will be shown. Then a brief description about the external user interface will FRI follow. If interested, a full explanation of the internal controller KRC and all the operation proggramming issues can be found in the original KUKA manuals.

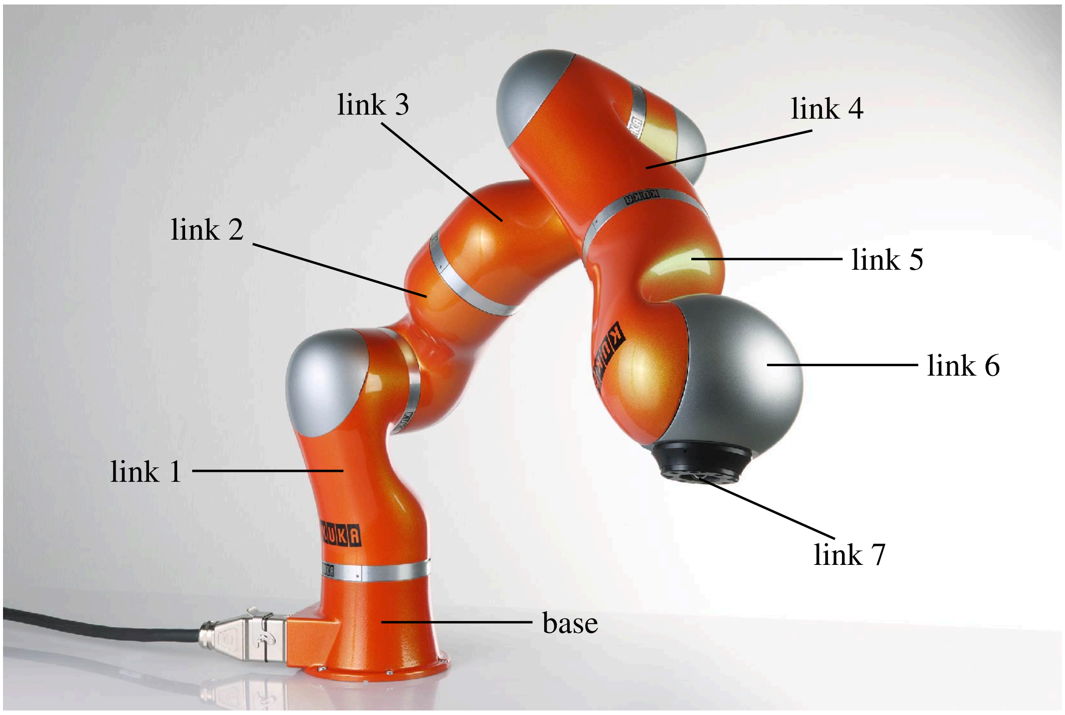
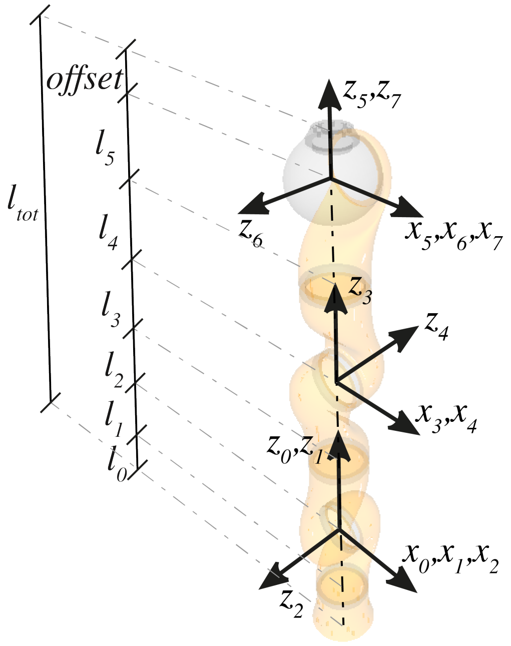


Figure 4.1. The KUKA LWR4+

4.1 The Kinematic structure

The KUKA LWR 4+ has a serial structure of 7 links (Figure 4.1). Its kinematic configuration is show in Figure 4.2. It has been built using the Modified *Denavit-Hartenberg notation* (*MDH*), whose parameters are also in Figure 4.2. This convention is built in such a way that the z axis of each frame corresponds to the rotational axis of the realtive joint. Then, the variables q_j are the rotational angles around z_j of each $j-th$ joint.



Kinematic Parameters

$l_0 = 11cm$
$l_1 = l_2 = l_3 = l_4 = 20cm$
$l_5 = 19cm$
$offset = 7.8cm$
$l_{tot} = 117.8cm$

MDH Parameters

j	α_j	d_j	θ_j	r_j
1	0	0	q_1	0
2	$\pi/2$	0	q_2	0
3	$-\pi/2$	0	q_3	$(l_2 + l_3)$
4	$-\pi/2$	0	q_4	0
5	$\pi/2$	0	q_5	$(l_4 + l_5)$
6	$\pi/2$	0	q_6	0
7	$-\pi/2$	0	q_7	0

Figure 4.2. Kinematic structure

It follows that the H-matrix $H_j^{j-1}(q_j)$ between the joint j and the joint $j-1$ is always like:

$$H_j^{j-1}(q_j) = \begin{bmatrix} \cos q_j & -\sin q_j & 0 & 0 \\ \sin q_j \cos \alpha_j & \cos q_j \cos \alpha_j & -\sin \alpha_j & -r_j \sin \alpha_j \\ \sin q_j \sin \alpha_j & \cos q_j \sin \alpha_j & \cos \alpha_j & r_j \cos \alpha_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.1}$$

4.2 The Dynamic model

In this section a bond-graph representation of the the dynamic model of the KUKA LWR4+ is shown. A good approximation of the inertia parameters of the arm can be found in [2]. The same parameters were also used in this thesis to validate the code explained in the next chapter.

4.2.1 The model of the joints

A joint establishes an energetic connection between two links. Then, it imposes a relation between the wrenches and the twists of the two bodies.

Since the wrench applied in a joint is applied to the connected links, the same wrench is applied to both bodies. The relation between the wrench of the actuator and the ones applied to the links is:

$$W_{actj} = W_{linkj} = W_{linkj-1} \tag{4.2}$$

Furthermore, the relation between the twist of the link $j-1$ with respect to the inertial frame expressed in a frame j ($T^{j,0}_{j-1}$) and the twist of the link j withe respect to the inertial frame expresed also in the same frame j ($T^{j,0}_{j}$) is:

$$T^{j,0}_{j} = T^{j,0}_{j-1} + T^{j,j-1}_{j} \tag{4.3}$$

In bond-graphs the two equations above are represented by a **0**-junction (Figure 4.3). This means that a joint is an *energetic connection* between two links. To do so all the twists and the joint must be expressed in the same frame. Here the j-th frame is chosen.
A joint performs two actions in the system:

1. It applies constraints in certain directions;

2. It allows motion in the others.

All the joints of the KUKA LWR 4+ are rotational and then only a degree of freedom is allowed. Furthermore, the choice of MDH convention implies that the direction with freedom is the rotation around the z axis of each DH frame. The relation between the torque applied by the actuator and the corresponding wrench expressed in the j-th frame is:

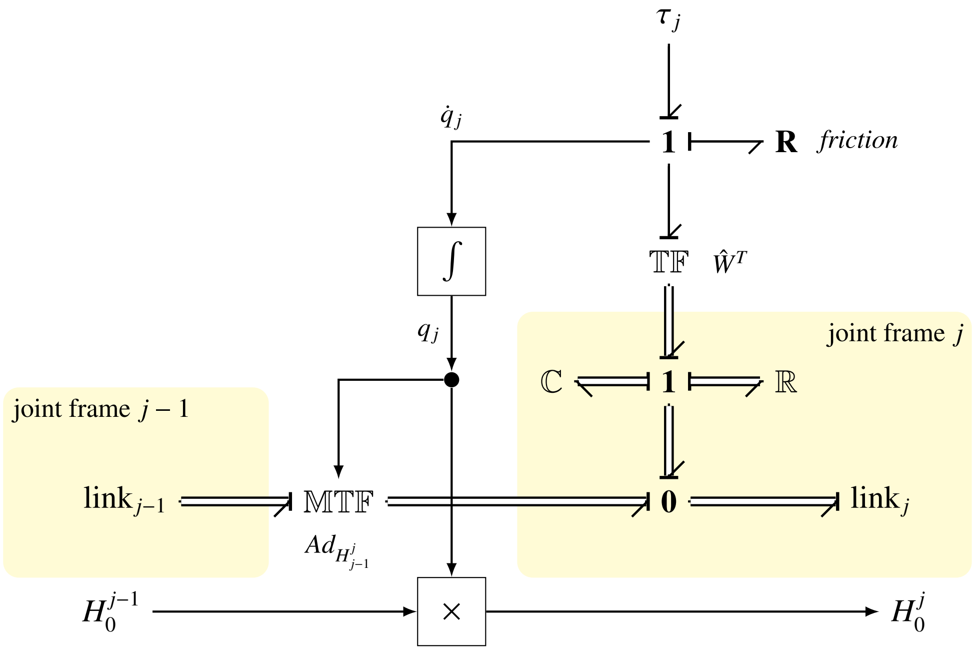


Figure 4.3. Bond-graph model of a joint

$$W^j = \hat{W}\tau_j \tag{4.4}$$

where $\hat{W} = \begin{bmatrix}\hat{\tau} & 0_{1x3}\end{bmatrix}$ and $\hat{\tau} = \begin{bmatrix}0 & 0 & 1\end{bmatrix}$. The power continuity impose a relation between the wrench W^j and the twist $T_j^{j,j-1}$of the link and the torque τ_j and the angular velocity $\dot{q}_j$ of the actuator:

$$(W^j)^T T_j^{j,j-1} = (\hat{W}\tau_j)^T T_j^{j,j-1} = \tau_j \hat{W}^T T_j^{j,j-1} = \tau_j \dot{q}_j \tag{4.5}$$

such that:

$$\dot{q}_j = (\hat{W})^T T_j^{j,j-1} \tag{4.6}$$

Equations (4.4) and (4.6) correspond to a 𝕋𝔽 element with $\hat{W}^T$ as parameter (Figure 4.3).

A six dimensional constraint can be modeled as a ℂ-element with really high gains on the constrained directions (ideally ∞) ad really low gains in the free directions (ideally $-\infty$) plus a ℝ-element in integral form with really high gains in the constrained direction (ideally ∞) and null gains in the free directions, both linked with a **1**-junction (the relative twist must be the same). The equations of a constraint are then:

$$(W_C)^T = C \cdot \int T_j^{j,j-1} dt \tag{4.7}$$

$$(W_R)^T = R \cdot T_j^{j,j-1} \tag{4.8}$$

Where $C = \text{diag}\Big(\big[\infty \quad \infty \quad -\infty \quad \infty \quad \infty \quad \infty\big]\Big)$ and $R = \text{diag}\Big(\big[\infty \quad \infty \quad 0 \quad \infty \quad \infty \quad \infty\big]\Big)$. Eventually a friction model can be added between the effort source and the TF element. The same concept of power continuity can be done between the couples of wrenches and twists respectively expressed in the frames $j-1$ and j.
The relation between the twist $T_{j-1}^{j-1,0}$ of frame $j-1$ with respect to the frame 0 expressed in the frame $j-1$ and same one expressed and the same twist expressed in the frame j $T_{j-1}^{j,0}$ is:

$$T_{j-1}^{j,0} = Ad_{H_{j-1}^{j}} T_{j-1}^{j-1,0} \tag{4.9}$$

and the dual relations between the wrench W^{j-1} expressed in the frame $j-1$ and the same frame expressed in the frame j (W^j) is:

$$(W^{j-1})^T = Ad_{H_{j-1}^{j}}^{T} (W^j)^T \tag{4.10}$$

These last two equations are again equivalent to a transformation element in bondgraph, but, since $Ad_{H_{j-1}^{j}(q_j)}$ varies with q_j, it is a $\mathbb{MTF}$ element (always Figure 4.3). The × block multiply the matrices H_{j-1}^{j} $(= (H_j^{j-1})^{-1})$ and H_0^{j-1} and gives as output the matrix H_0^j. This is needed, as it will be shown, in the model of the link to model the gravity effects.

4.2.2 The model of a link

The model of rigid-body dynamics consists of inertia and gyroscopic effects. Defining as c_j the frame attached to the center of mass of the j-th body with the axes oriented as the principal inertia axes, the balance of wrenches is:

$$I^{c_j} \dot{T}_j^{c_j,0} = \begin{pmatrix} \tilde{P}_\omega^{c_j} & \tilde{P}_v^{c_j} \\ \tilde{P}_v^{c_j} & 0 \end{pmatrix} T_j^{c_j,0} + (W^{c_j})^T \tag{4.11}$$

Where

- $I^{c_j} = \begin{pmatrix} J & 0 \\ 0 & M \end{pmatrix}$ is the inertia tensor expressed in the principal inertia frame;

- $P^{c_j} = I^{c_j} T_j^{c_j,0}$ is the momentum, divided in the angular components P_ω and the linear components P_v. These are in the skew form;

- W^{c_j} is the external wrench expressed in the principal inertia frame;

- $T_j^{c_j,0}$ is the twist of the link with respect to the inertial frame always expressed in the principal inertia frame.

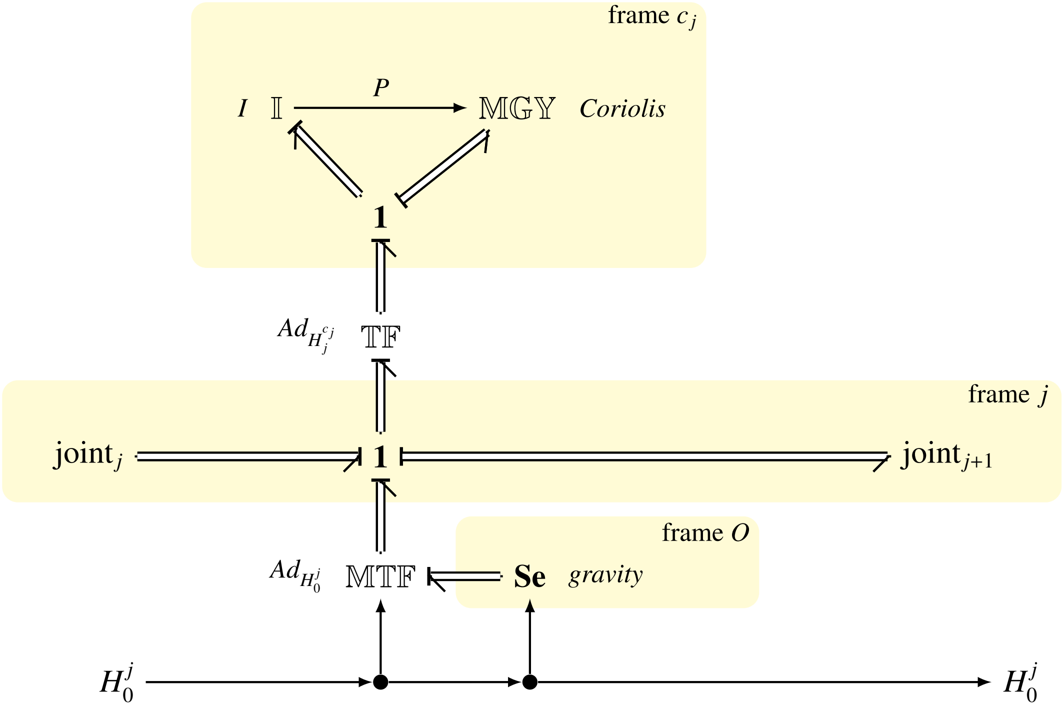


Figure 4.4. Bond-graph model of a link

The first term of Equation (4.11) corresponds to an $\mathbb{I}$ element, the second is represented as MGY while the thirds, the external force, came from the bond that links the body to the joints. This three element are linked by a **1**-junction (Figure 4.4).

The relation between the twist $T_j^{c_j,0}$ of the body with respect of the inertial frame expressed int the body pricipal inertia frame c_j and the same twist $T_j^{j,0}$ expressed in the frame j is, again:

$$T_j^{j,0} = Ad_{H^j_{c_j}} T_j^{c_j,0} \tag{4.12}$$

The one between the external wrench W^{c_j} expressed in the c_j frame and the same wrench expressed in the frame j is:

$$(W^j)^T = Ad^T_{H^{c_j}_j}(W^{c_j})^T \tag{4.13}$$

Then, as for the joint, the change of coordinates of wrenches and twists between two different frames is a $\mathbb{TF}$ with $Ad_{H^{c_j}_j}$ as parameter.

The external wrench W^j is the sum of the wrench W^j_j transmitted by the joint j, the wrench W^j_{j+1} transmitted by the joint $j+1$ and the resulting gravitational wrench W^j_g, all expressed in the frame j:

$$W^j = W^j_j + W^j_{j+1} + W^j_g \tag{4.14}$$

Since all the bonds with these efforts contains variables expressed in the frame, they also have the same twist. This is a **1**-junction, whose twist is causally imposed by the body (Figure 4.4).

The gravitational wrench W^j_g is the result of the change of coordinates of the wrench W^0_g, given as a source element. The adjoint $Ad_{H^j_0}$ is a function of q_i with $i = 1, \cdot, j$. Then a $\mathbb{MTF}$ is necessary. Alwais in Figure (4.4) the signal H^j_0 that came from the previous joint is used to calculate of the gravitational wrench and the $Ad_{H^j_0}$.

4.2.3 The entire model

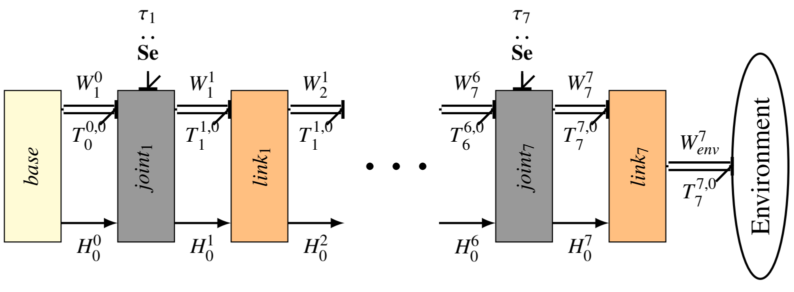


Figure 4.5. Bond-graph model of the KUKA LWR4+

The model of a joint (Figure 4.3) has three power ports (link_{j-1} and τ_j as input, link_j as output) and two signal ports (H_0^{j-1} as input and H_0^j as output) and the model of a link (Figure 4.4) has three power ports (joint_j as input, joint$j-1$ as output) and two signal ports (H_0^{j-1} as input and output). Then, the entire model of the KUKA LWR4+ is shown in Figure 4.5.

4.3 FRI Library

The Fast Research Interface (or FRI) is a C library that improves a simple user interface to the KUKA LWR 4 and hides all communication and set-up issues behind interface.
In this section its main features will be shown. An extended and complete documantation can be found in [6].

The FRI Interface Library runs on a remote PC node that is connected to the KRC (KUKA Robot Controller) via an Ethernet connection. In intervals of 1 to 100 milliseconds, UDP packages are periodically sent from the KRC unit to the remote host. These packages contain a complete set of robot control and status data (e.g., joint positions, joint torques, drive FRIDriveTemperatures, etc.). The remote host has to instantaneously send a reply message after the reception of each package. A reply message contains input data for the applied controllers (e.g., joint position set-points, joint stiffness set-points, etc.). The controller provided are:

- Joint position controller
- Cartesian impedance controller
- Joint impedance controller

Data transfer between the robot controller and an external computer is carried out using the following modes:

- **Monitor mode**: Cyclical communication with transfer of robot data to an external computer.
- **Command mode**: Cyclical communication with transmission of commands from an external computer to the robot controller. In Command mode, robot motion can also be controlled externally.

In particular, there are two status graph, one running on the KRC and one running on the remote PC. The second is ruled by the first. In Section 5.2.2 the status graph of the remote PC and its rules are explained. Here the KRC FRI status graph is shown.
The interaction between the two modes in the KRC is schematized in the status graph of Figure 4.6. The functional principle is the following:

- FRIOPEN activates the FRI connection between the robot controller and the external computer. At this point in time Monitor mode is active. In this mode, there is cyclical data exchange between the KRL and the robot. The quality of the data transfer is also determined and classified.
- FRISTART starts Command mode if the quality of the data transfer has been classified as Good or Perfect. Data exchange between the robot controller and the robot and determination of the quality of the data transfer continue. If the quality of the data transfer is not high enough (edge ”4”) the activation of Command mode is not allowed. If, instead, the comunication quality is at least good, the state machine switch to Command mode and remote control is allowed.
- FRISTOP stops Command mode and switches to Monitor mode.
- FRICLOSE closes the FRI connection.
- × indicate the faults. If there is a fault the machine is forced to switch in Monitor mode.

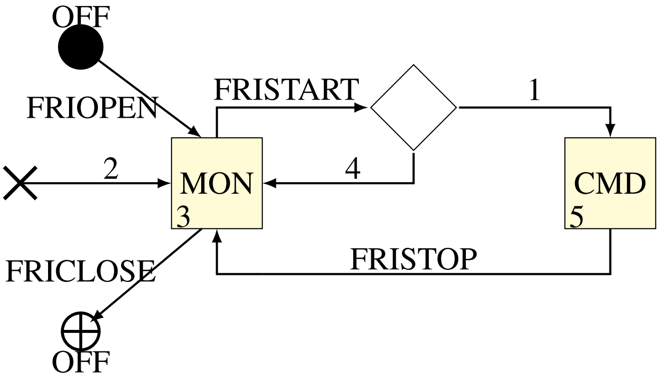


Figure 4.6. Status graph of the FRI

Chapter 5

A passive based control on the KUKA LWR 4+

In this chapter it is explained how the Intrinsically Passive Control has been implemented on the KUKA LWR4+. First a full description of the control is given, starting from what kind of passive system the IPC has to simulate (see Sec. 3.4) and the issues around it. Then, a brief explaination of the *Robotic Operation System* (ROS) is given, in order to understand how the code written for this thesis works. A description of the code and its structure will follow.

5.1 Design the IPC for the KUKA LWR4+

The aim of this control is to achieve a desired *passive* behavior, as explained in Section 3.4. This means that the IPC has to simulate a physical passive system. Figure 3.2 shows the general scheme of the IPC. In that figure the pre-compensation of undesired dynamics is exploited. These dynamics, generally, are the gravity effects and the frictions presents in each joint.

Unfortunately an accurate model of the KUKA LWR4+ is not available. This imply that is not possible to design a precise compensation of the gravity. Anyway, gravity compensation is already perfectly done by the KRC. Then, it is delegated to the internal controller of the arm and its presence in the control scheme will be neglicted from here on.

The same problem regarding gravity compensation is present also for frictions. An official documentation about them does not exist. Luckily, the brushless motors embedded in the arm are equipped with harmonic drives. The friction presentes in this kind of drives is very low, almolt neglictable. Furthermore, as it will be explained later, also an implicit internal compensation of this really low frictions is made by the KRC. Then, also the terms concerning frictions and their compensation are not considered in the following.

It has been largely explained in Cap. 3 that this control technique is based on energy shaping strategy. Indeed, this is the effect of gravity compensation. The shape of the compensated plant energy is flat for each joints configuration. Then, IPC has to ”re-shape” the energy function at the end effector in order to achieve a strict minimum in the desired configuration x^*. A possible desired energy shape is shown in Figure 5.1, that is the characteristic energy function of any *geometrical* and *winding spring*.

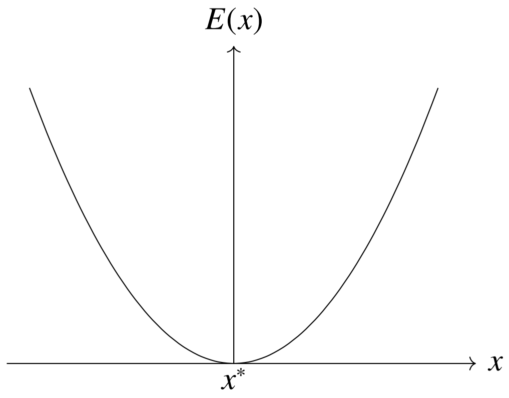


Figure 5.1. Desired energy function

To fully understand what a spring is in a mathematical and geometric point of view, a characterization of springs is given here below.

5.1.1 Springs

In mechanical domain *ideal springs* are ideal power-conserving concepts of storage of potential energy. In general there exist two kinds of springs: *geometrical springs* and *winding springs*. Winding springs are a particular kind of springs which are able to create different wrenches for exactly the same configurations of the bodies they are attached to.

5.1.1.1 Geometrical Springs

A spring is a *nodic* element. A nodic element is a physical dynamical subsystem with $n > 1$ ports whose state change does not depend on the absolute value of its ports but on their relative values.
It follows that to define a spring at least two bodies (or in a geometric point of view, two frames) are required. Then, given m bodies the spring is defined by its energy storage function.

Definition 5.1.1. Spring Given m bodies contained in m Euclidean spaces $\mathcal{E}_1, \ldots, \mathcal{E}_m$, we call *spring* a function of the following form:

$$V : SE_2^1(n) \times SE_3^2(n) \times \ldots \times SE_m^{m-1}(n) \to \mathbb{R} \tag{5.1}$$

A spring is a *passive* physical element. This means that it cannot supply infinite energy. Then, the function V must have a finite minimum.
To calculate the constitutive equations of a spring it is necessary to consider the differential of the energy function:

$$dV : SE_2^1(n) \times SE_3^2(n) \times \ldots \times SEm^{m-1}(n) \to T^*SE_2^1(n) \times T^*SE_3^2(n) \times \ldots \times T^*SE_m^{m-1}(n)$$

In a certain point $h = (h_2^1, \ldots, h_m^{m-1}) \in SE_2^1(n) \times \ldots \times SEm^{m-1}(n)$ and for a velocity $\dot{h} \in T_{h_1^2}SE_2^1(n) \times \ldots \times T_{h_m^{m-1}}SE_m^{m-1}(n)$, the differential is such that $dV(h)$ applied to $(h, \dot{h})$ represents the increase of energy due to a change of relative positions $\dot{h}$.
Then, if we define:

$$T_V := t_2^1 \times \ldots \times t_m^{m-1} \in se_1(n) \times \ldots se_{m-1}(n) \tag{5.2}$$

where t_i^j is an element in $se_j(n)$ anb represent the twist of the body i with respect the body j, and

$$W_V := w_2^1 \times \ldots \times w_m^{m-1} \in se_1^*(n) \times \ldots se_{m-1}^*(n) \tag{5.3}$$

where w_i^j is an element in $se_j^*(n)$ and represent the wrench applied by body i to the spring connecting bodies i and j and it is expressed in the space of body i, the increase of energy of the spring at a certain istant of time is:

$$\dot{V} = \langle W_V, T_V \rangle = w_2^1 t_2^1 + \ldots + w_m^{m-1} t_m^{m-1} \tag{5.4}$$

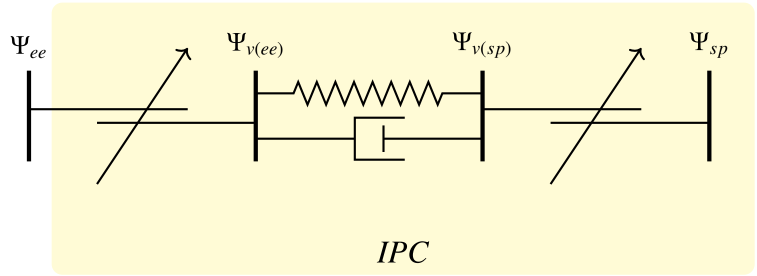


Figure 5.2. IPC scheme

In Hamiltonian terms, a spring is then described by the following equations:

$$\dot{H} = \chi_H T_V \tag{5.5}$$

$$W_V = \chi_H^* \frac{\partial V}{\partial H} \tag{5.6}$$

where $H \in SE_2^1(n) \times \ldots \times SE_m^{m-1}(n), \chi_H := \chi_{H_2^1} \times \ldots \times \chi_{H_m^{m-1}}$, with $\chi_{H_m^{m-1}} := (\pi^j_{H_i^j})^{-1}$ and $\pi^j_{H_i^j} : T_{H_i * j} SE_i^j(n) \rightarrow se_j(n)$, and χ_H^* is the dual of χ_H. The equations above corresponds to a **C**-element in its integral form in the Bond-graph formalism.
It is also possible to define V with respect to the relative motion of just two bodies i and j. It will be:

$$V_i^j(H_1^j, \ldots, H_{i-1}^j, H_{i+1}^j, \ldots, H_m^j); SE_i^j(n) \rightarrow \mathbb{R}; H_i^j \mapsto V(H_j^1 o H_2^j, \ldots H_j^{m-1} o H_m^j) \tag{5.7}$$

Consider now two frames Ψ_i and Ψ_j Since the potential energy V_i^j has a minimum it is possible to design it such that this minimum is at the identity of the frames, that is $\Psi_i \equiv \Psi_j$, so where $H_i^j = I$. The relative configuration H_i^j in which V has its minimum is called *center of stiffness* and it is possible to study the elastic behavior by using coordinates.
Furthermore the two energy functions:

$$\check{V}_i(H_i) : SE_i \rightarrow \mathbb{R}; H_i \mapsto V_i^j(H_i^{-1}) \tag{5.8}$$

$$\check{V}_j(H_j) : SE_j \rightarrow \mathbb{R}; H_j \mapsto V_i^j(H_j^{-1}) \tag{5.9}$$

have a minimum at the identity of $SE_i(n)$ and $SE_j(n)$.

Finally, considering $d\check{V}_l : SE_l(n) \rightarrow T^* SE_l(n)$ with $l = i, j$, it is possible to define the *stiffness map* K_l, that chacaterizes the spring, as the lnearization of $d\check{V}_l$ around the identity $I \in SE_l(n)$:

$$K_l : se_l(n) \in se_l^*(n), T_l \mapsto \frac{d}{dt}(d\check{V})|_{t=0} \quad l = i, j \tag{5.10}$$

and the relation between the stiffness maps expressed in the two spaces is:

$$K_j = Ad^T_{H^i_j} K_i Ad_{H^i_j} \tag{5.11}$$

5.1.1.2 Winding Springs

A winding spring is, in real world, a rotational spring. Consider such kind of spring between two bodies. If we rotate one of them by 360 degrees around a fixed axis, the final configuration is the same as the initial. Anyway, the the forces applyed between the two bodies are not the same. This is what is called a winding spring:

Definition 5.1.2. Winding Spring A *winding spring* between two bodies i and j is a spring with a lower bounded energy function having the form:

$$V_w : \tilde{Q}^j_i \rightarrow \mathbb{R}; \quad \tilde{Q}^j_i \in Q^j_i \subset SE^j_i(n) \tag{5.12}$$

such that $q_1 \sim q_2, q_1 \neq q_2$ implies $V_w(q_1) \neq V_w(q_2)$.

The energy function of a winding spring is the same shown in Fig. 5.1.

Going back to the implementation of the IPC, what our control has to simulate is a spring attached to the end effector and eventually a damper to inject some desired energy dissipation in the behaviour. It could be use full to have a variable rest length spring in order to accomplish particular tasks. A 1D graphical representation of it is given in Figure 5.2. In that figure Ψ_{ee} is the frame fixed to the edge of the end effector and Ψ_{sp} is the set-point frame, while $\Psi_{v(ee)}$ and $\Psi_{v(sp)}$ are the *virtual* positions of the end effector and the set point respectively. As the figure shows, it is possible to modify the length of the spring from both its vertices, varying the distance (and a 6D case, also the orientation) between the ”real” and the ”virtual” frames.

Figure 5.3 shows the spatial impelementation of the IPC on the KUKA LWR4+.

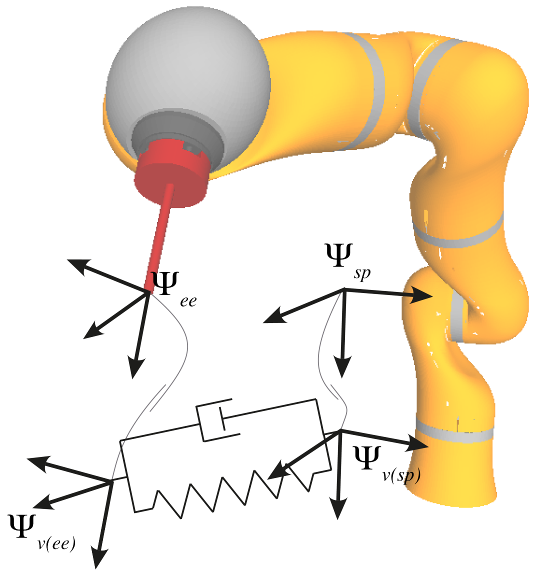


Figure 5.3. IPC implementation on the KUKA LWR4+

The bond graph of this variable length spatial spring, comprehensive of damping injection ($\mathbb{R}$-element) is presented in the figure here below.

$$\begin{tikzcd}
T^{0,0}_{ee} : \mathbf{1} \arrow[r, harpoon] & \mathbf{0} \arrow[d, harpoon] & \overset{Ad_{H^0_{ee}}}{\mathbb{MTF}} \arrow[l, harpoon] & S_f : T^{ee,ee}_{v(ee)} \arrow[l, harpoon] & & & \\
 & T^{0,0}_{v(ee)} : \mathbf{1} \arrow[d, harpoon] & & & & & \\
 & \mathbf{0} \arrow[r, harpoon] & \overset{T^{0,v(sp)}_{v(ee)}}{\mathbf{1}} \arrow[r, harpoon] & \overset{Ad_{H^{v(sp)}_0}}{\mathbb{MTF}} \arrow[r, harpoon] & \overset{T^{v(sp),v(sp)}_{v(ee)}}{\mathbf{1}} \arrow[r, harpoon] \arrow[d, harpoon] & \overset{\chi_{H^{v(sp)}_{v(ee)}}}{\mathbb{MTF}} \arrow[r, harpoon] & \mathbb{C} :: V^{v(sp)}_{v(ee)}(H^{v(sp)}_{v(ee)}) \\
 & T^{0,v(sp)}_0 : \mathbf{1} \arrow[u, harpoon] & & & \mathbb{R} :: R_{damping} & & \\
 & \mathbf{0} \arrow[u, harpoon] \arrow[d, harpoon] & & & & & \\
T^{0,0}_{sp} : S_f \arrow[r, harpoon] & \mathbf{0} & \overset{Ad_{H^0_{sp}}}{\mathbb{MTF}} \arrow[l, harpoon] & S_f : T^{sp,sp}_{v(sp)} \arrow[l, harpoon] & & &
\end{tikzcd}$$

Figure 5.4. Bond grah of the variable length spatial spring

In Figure 5.4 it is possible to notice the presence of three flow source elements. That are the power port through which the supervisor can inject energy in the controlled system. On the other hand the **1**-junction in the upper right corner is the power port that connect the spring, that is the IPC, to the robotic arm. The general bond graph scheme of the IPC is shown in Figure 5.5. Notice that such approach is completely coordinate-free.

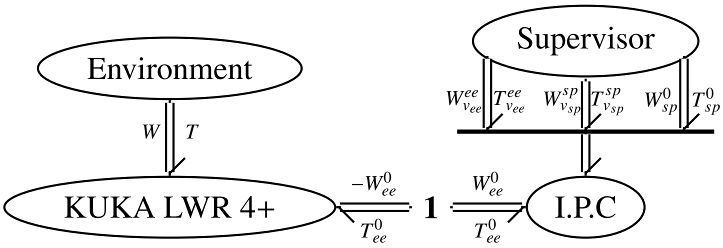


Figure 5.5. Bond-graph scheme of the IPC

The shape of energy that we would like to achieve is something really similar to the one of Fig. 5.1. This is good for the "translational" part. It has been characterized that, thinking about a rotational spring, Fig. 5.1 corresponds to the energy of a winding spring. For our purpose this will take to an undesired behavior. Indeed, if we define a diagonal 6 by 6 stiffness matrix K and simply apply the relation between displacement and forces applied to the end effector through the linear map $K : se(3) \in se^*(3); \delta T \mapsto K\delta T$, where δT is the infinitesimal twist, a rotation of 360 degrees around one axis (that puts the end effector to the same configuration before the rotation) causes a torque around the same axis different than without that rotation. This behavior is illogical.

The torques applied to the end effector depends on the energy shape with respect to the angular displacement. Then, if we want that equivalent configuration $q_i \sim q_j$ (in homogeneous matrix $H(q_i) = H(q_j)$) lead to equivalent applied torques, the energy function of the rotational part has to be closed with respect to a circle (Fig. 5.6). Here below it is shown how to design an energy function corresponding to the one of Figure 5.1 for the translational displacement and to Figure 5.6 for the angular displacement. A complete explanation of it can found in [11], Chapter 5.

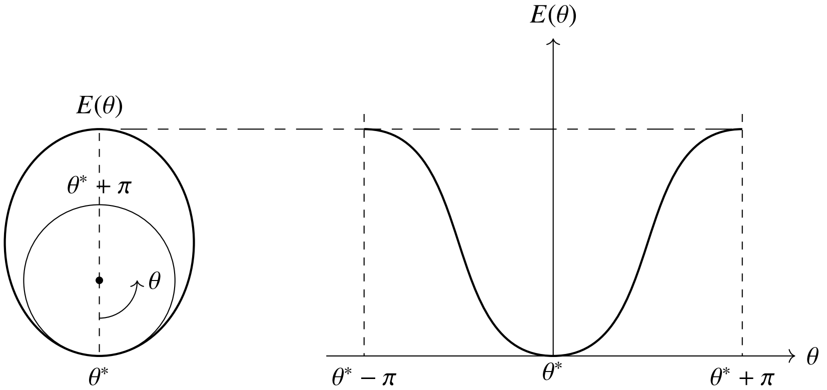


Figure 5.6. Desired rotational energy function

First of all, let's recall the tilde notation for a twist T_i^j and and its dual wrench W_i^j:

$$\tilde{T}_i^j = \begin{bmatrix} \Omega_i^j & v_i^j \\ 0 & 0 \end{bmatrix} \quad \text{with } \Omega_i^j = \tilde{\omega}_i^j = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}; \omega_i^j = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \tag{5.13}$$

$$\tilde{W}_i^j = \begin{bmatrix} \tilde{f}_i^j & m_i^j \\ 0 & 0 \end{bmatrix} \tag{5.14}$$

The rotational map can be exploited in the following way, where K is partitioned in its rotational, translational, and coupling components:

$$\begin{bmatrix} m_i^j \\ f_i^j \end{bmatrix} = \begin{bmatrix} K_o & K_c \\ K_c^T & K_t \end{bmatrix} \begin{bmatrix} \delta\theta_i^j \\ \delta p_i^j \end{bmatrix} \tag{5.15}$$

The matrix K is always symmetric and therefore K_o adn K_t, which are respectively called *rotational stiffness* adn *translational stiffness*, are also symmetric, while $K_c = K_c^T$ corresponds to the maximum decoupling to between rotation and translation. Furthermore, as said before, the point corresponding to the coinciding origins of the coordinate systems Ψ_i and Ψ_j at the equilibrium is called *center of stiffness*. Defining:

$$G_x = \frac{1}{2}\text{tr}(K_x)I - K_x; \quad x = o, t, c \tag{5.16}$$

the following equation is equivalent to Eq. (5.15)

$$\tilde{m}_i^j = 2\mathrm{as}(G_o\delta\tilde{\theta}_i^j) + 2\mathrm{as}(G_c\delta\tilde{p}_i^j) \tag{5.17}$$

$$\tilde{f}_i^j = 2\mathrm{as}(G_c\delta\tilde{\theta}_i^j) + 2\mathrm{as}(G_t\delta\tilde{p}_i^j) \tag{5.18}$$

that correspond to energy function of Fig. 5.1.
Relating the wrench and the displacement in the following way:

$$\tilde{m}_i^j = -2\mathrm{as}(G_oR_i^j) - \mathrm{as}(G_tR_j^i\tilde{p}_i^j\tilde{p}_i^jR_i^j) - 2\mathrm{as}(G_c\delta\tilde{p}_i^jR_i^j) \tag{5.19}$$

$$\tilde{f}_i^j = -R_j^i\mathrm{as}(G_t\tilde{p}_i^j) - \mathrm{as}(G_tR_j^i\tilde{p}_i^jR_i^j)R_i^j - 2\mathrm{as}(G_cR_i^j) \tag{5.20}$$

where $H_i^j = \left[\begin{smallmatrix} R_i^j & p_i^j \\ 0 & 0 \end{smallmatrix}\right]$ and the minus signs are used to create the minimum, the energy functions of Fig. 5.1 and Fig. 5.6 are achieved for the translation and orientation parts respectively. m_i^j and f_i^j are the torque and the force that applied to "body" i by the "body" j expressed in i.
It is important to underline that K, and then K_o, K_t and K_c are defined only at the identity point between Ψ_i and Ψ_j. Figure 5.7 show how the control has been implemented.

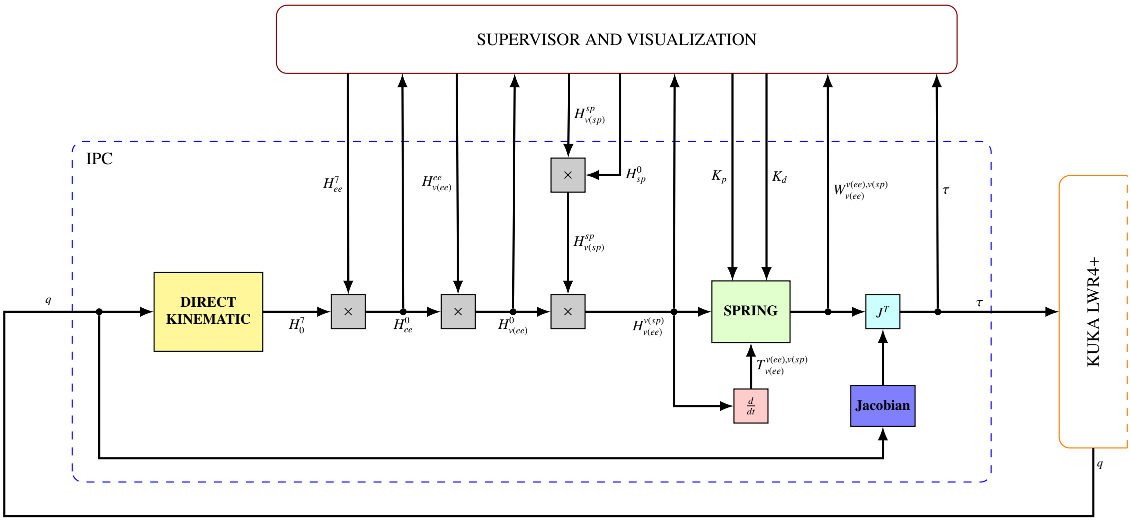


Figure 5.7. The IPC control scheme

In the block DIRECT KINEMATIC equations 1.45 are applied. Since each homogeneous transformation matrix H_j^{j-1} that express the position of link j with

respect to $j - 1$ depends on the MDH parameters and is of the form (Eq. 4.1), it calculates each H matrix as a function the position of each joint q_j received from the KUKA in the vector $q = [q_1, \ldots, q_7]^T$, applies Eq. (1.45) and returns $H_0^7 = (H_7^0)^{-1}$. The IPC also receives H_{sp}^0, $H_{v(ee)}^{ee}$ and $H_{v(sp)}^{sp}$ from the SUPERVISOR, as well as the stiffness matrix K_p and the dissipation matrix K_d. The derivative block $\frac{d}{dt}$ is used in order to obtain the twist between $\Psi_{v(ee)}$ and $\Psi_{v(sp)}$ expressed in $\Psi_{v(ee)}$ and to pass it to the SPRING block. SPRING uses $H_{v(ee)}^{v(sp)}$ to calculate the simulated spring wrench $W_{v(ee)}^{v(ee),v(sp)}$ with Eq. (5.20), taking into account also the damping effect, given by $K_d T_{v(ee)}^{v(ee),v(sp)}$, that is subtracted from the wrench resulting from Eq. (5.20). Finally, through the use of the geometric Jacobian (Sec. 1.5.3), the vector $\tau = [\tau_1, \ldots, \tau_7]$ (vector of the torques that the joint actuators have to apply in order to simulate the spring wrench) is sent to the KUKA LWR4+.
The SUPERVISOR AND VISUALIZATION block communicates with the IPC, in order to inject energy sending the matrices said before and receiving back some information to visualize on a screen, useful to the user to check what is going on. An important aspect of this control is that there is no inversion of kinematic. This means that the control is immune to singular configurations, that are often a big problem in controls like this, as the equivalent *impedance control*.

As said in Sed 4.3 it is possible to communicate, and then to control, the KUKA LWR4+ with a remote computer via Ethernet cable. This is done writing some programs that, through the FRI, control the robotic arm. In the next sections it is explained how.

5.2 The code

The implementation of the control scheme in Figure 5.7 has been done using the *Robotic Operative System* (ROS) environment, well known in the robotic world, and the programming in C++.
Anyway, to easily design the control, the IPC has been developed by using the 20-Sim [1] modeling and simulation software. The reason why 20-Sim has been used is that it allows the user to directly model systems using bond-graphs, and easily build a 3D simulation of our model in order to immediately check the correct behaviour of the controlled plant. This means that, once the bond-graph model of the KUKA LWR4+ has been designed, it was possible to duplicate it in the 20-Sim environment as it has been drawn (Fig. 4.5). Furthermore, once the controller

is tested and checked, it is possible to automatically generate a C++ class of the controller and use it in the ROS environment.
In Figures 5.8 and 5.9 the entire model of the KUKA arm, and the entire simulated sistem are shown. These are exactly the same as the ones shown and explained of Figure 4.5 and 5.7.

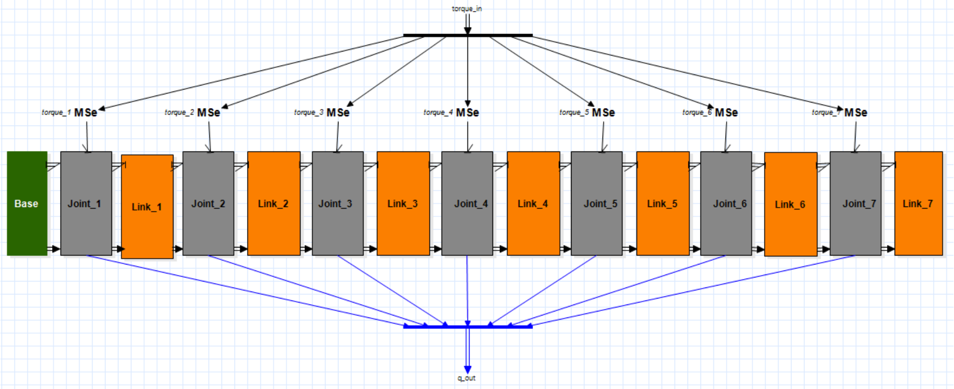


Figure 5.8. Bond-graph model of the KUKA LWR4+ in 20-Sim environment.

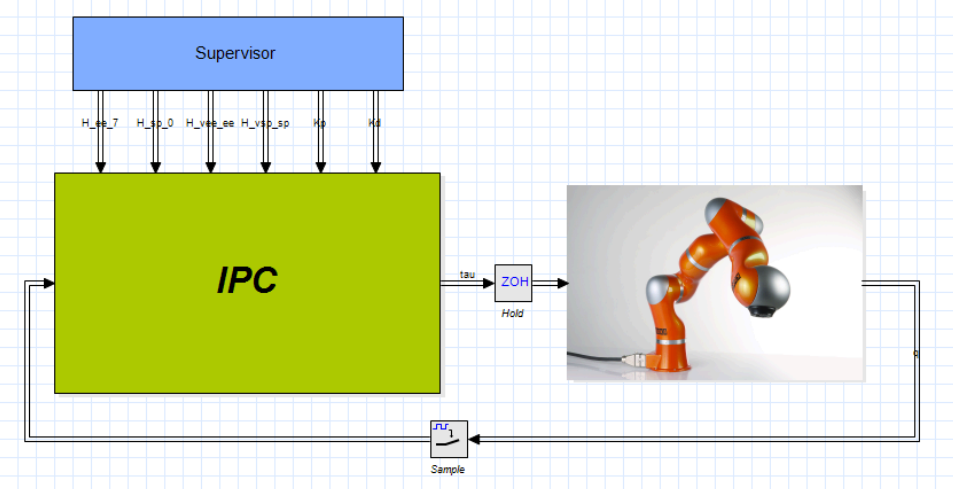


Figure 5.9. Total simulated system in 20-Sim environment.

An explanation of main features of ROS will follow. For a full explanation of how it works it is suggested to look at [8]. Then, the structure of implemented codes and data communication is given.

5.2.1 ROS

The Robot Operating System (ROS) [8] is a flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms.
ROS works with two kinds of elements:

- **Nodes**: executable files, contained into ROS packages that are collector of libraries and executable regarding a certain argument;
- **Topics**: communication channels that allow nodes to exchange data and information.

For each ROS node it is possible to specify the frequency/rate at which it runs. E.g, specify a rate of 20 means that the frequency at which the node runs is 20 Hz (1 time each 0.05 seconds). Furthermore, nodes can communicates through topics. To be able to do so a node needs to be an **advertiser** on a topic (if it has to send certain information) or a **subrisciber** to a topic (if it has to listen and receive some information to elaborate). A topic is characterized by the kind of data (**message**) that can be broadcasted through it and the size of the queue of messages that it can contain. This means that if the size of the topic is set to *n*, *n* messages are mmaintainedin the queue waiting for some node to consume them. If the queue is already full and a new message is published to the topic, the oldest message in the queue (indeed, the first) is eliminated and the new one is inserted to the bottom of the queue. The syntax to publish on a topic is:

```
ros::Publisher publisher_name = n.advertise<message_tipe>("topic_name", queue_size);
```

Once the publischer is initialized it is possible to publish the message on the topic with:

```
publisher_name.publish(msg);
```

The syntax o subscribe on a topic:

```
ros::Subscriber subscriber_name = n.subscribe("topic_name", queue_size, topic_CallBack);
```

where `topic_CallBack` is a function with which the node treat the message received. In its declaration the type of message that it has to handle has to be specified as inputs:

```
void topic_CallBack(const message_type::ConstPtr& msg){
    ...
    %code
    ...
}
```

`message_type` can be anything, from just a integer to an array of double to a structured data.

Now the explanation of all the nodes, and relative topics, used to control the KUKA LWR4+ is given.

5.2.2 FRI node

It has been shown in Sec. 4.3 that the communication between the KRC and the PC host is carried out by means the two modes MONITOR and COMMAND. The transitions between these two modes in the KRC is shown always in Section 4.3, while the operation in the remote PC, ruled by the state of KRC, is here explained.

The FRI node is a node that uses the FRI library in order to communicate with the KRC, adapted in such a way that it is able to run in the ROS environment and, then, its role is to build a *bridge* between the KRC and the PC. In this way it is possible to run some control nodes that communicate with the FRI node, which has the task to communicate data relative to the operative state of the KUKA (such that running control mode, position of the joints, setted stiffness of the joint, ecc...) to the rest of the running nodes and to receive control commands from control nodes (as desired joint position, stiffness, torque to apply..) and communicate them to the KRC. To be able to have this operation, KRC and FRI node must be syncronized. Then, a state machine guides the FRI node, ruled by the one of the KRC. The states of the FRI node are: IDLE, INIT, MON, PRECMD, ENDCMN, CMD, KILL. Interaction bewteen states is shown in Fig. 5.10.

The requirements to enable FRI and go into monitor mode are on the KRC because this happens before the first communication with the FRI host machine. To allow monitor mode (MON) in FRI node the configurations on the KRC must be checked, such as a load data and estimated external joint torque. If something is not matched it is not possible to enable FRI.
When FRI node is initialized and monitor mode is running there are also some requirements for the state transition to CMD mode. These constraints are: drives are

engaged, command flag (that is a flag that indicates the control type for the KRC, see Sec. 4.3) is setted and it is constant, the commanded position of the external controller is almost identical to the commanded position of the internal controller.
When running in command mode (CMD) it is not allowed to change control flags and the variable that corresponds to these control flags must be setted. When a change of the control flag is desired while in command mode, the mode needs to go back to monitor mode where the control flag can be changed.
The control of the KRC control mode is done by means of a state machine in FRI node. This state machine guides the KRC state machine by sending its state as an integer value.
The way the FRI node state machine works is that it starts at ”IDLE” and receives a request to initialize a FRI connection and transitions to the INIT state. Inside INIT the correct ROS publishers and subscribers are created and it then waits for the FRI connection to transit into MON state. In monitor mode information is send back over ROS topics, the cmd flags are setted in the KRC and the commanded position of the KRC is copied and send back to the KRC as commanded position by the FRI controller. When command mode is desired, the state is changed to PRECMD, where measurements continue but the command flags cannot be changed anymore, and the KRC side of the state machine attempts to FRI START to change in CMD mode. When the KRC changes to command mode the FRI node state machine follows and the robot can be controlled in the variables setted by the CMD flags. To stop FRI connection the ROS side state becomes ENDCMD where neither the commanded variables or the cmd flags can be changed, the KRC side of the state machine attempts to FRI STOP to change back to MON MODE. When the KRC is back into monitor mode the FRI node state machine will follow.

As said above, it is possible to chose the control type through the command flag. Here only the case of the *join impedance controller* is analyzed. This because this type of control allows to change the stiffness of the joints and to control them sending the desired torques. Then it is possible to set the stiffness to zero and control the robot in torques, that is exactly the purpose of the IPC (Fig. 5.7).
The topic for which the FRI is a publisher is just `FRI_MsrJnt`, through which the FRI node send the position of the seven joints using the message type `Jnt_pos` , a message containing an array `float32[7]`.
Furthermore, FRI node is subscribed to the following topics:

`FRI_Control`	message type:	`FRI_control`

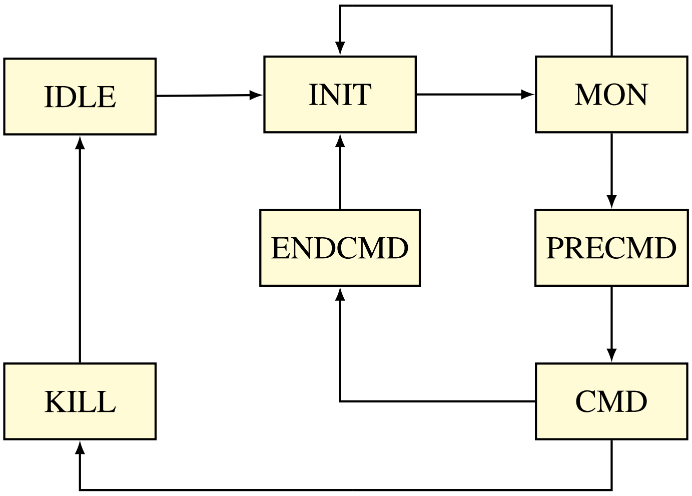


Figure 5.10. Status graph of the FRI node

`FRI_send_stiffness`	message type:	`Jnt_stiff`
`FRI_sendtorque`	message type:	`Jnt_torque`

through which the controller node (in this case the IPC node) send the command to be sent to the KRC. The three message types above contain:

- `FRI_control`: array of `bool[5]` containing five booleans to able or disable the control commands:
    - `Position`
    - `Velocity`
    - `Stiffness`
    - `Damping`
    - `Torque`
- `Jnt_stiff`: array of `float32[7]`
- `Jnt_torque`: array of `float32[7]`

An explanation of the other ROS nodes follows.

5.2.3 IPC node

The IPC node is the ROS node in which the Intrinsically Passive Controller is implemented (Sec. 5.1). The topic in which it is involved are listed below, divided in two section: topics for the communication IPC-Supervisor (and in general to all other nodes) and topic specifically for communication IPC-Plant (through FRI).

5.2.3.1 IPC-Plant topics

Subscribers:	**message type**	
FRI_MsrJnt	Jnt_pos:	floatt32[7]
Publisher:		
FRI_Control	FRI_control:	bool[5]
FRI_send_stiffness	Jnt_stiff:	float32[7]
FRI_send_torque	Jnt_torque:	float32[7]

5.2.3.2 IPC-Supervisor topics

Subscribers:	**message type**	
SetEndEffector	Cart_pose:	float32[12]
SetVee	Cart_pose:	float32[12]
SetPoint	Cart_pose:	float32[12]
SetVsp	Cart_pose:	float32[12]
SetStiffness	Twist:	standard message belonging to the class geometry_msgs provided by ROS

SetEndEffector,SetVee,SetPoint,SetVsp are used by the supervisor to set H^7_{ee}, $H^{ee}_{v(ee)}$,H^0_{sp},$H^{sp}_{v(sp)}$ while through SetStiffness it set the stiffness of the spring. H^7_{ee} is set only one time just after the launch of the node. It depends on the position and orientation of the tip of the end effector (the tool) used.
Cart_pose contains all the datas of the relative homogeneous matrix. Since the last row of each H-matrix is always [0001], the format of Cart_pose message and the correspondence with H-matrix elements is:

$$H = \begin{bmatrix} r_{xx} & r_{xy} & rxz & p_x \\ r_{yx} & r_{yy} & ryz & p_y \\ r_{zx} & r_{zy} & rzz & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$msg = [r_{xx}, r_{xy}, rxz, p_x, r_{yx}, r_{yy}, ryz, p_y, r_{zx}, r_{zy}, rzz, p_z]$$

`Twist` is a message containing two structure, `linear` and `angular`, each one composed by three `float64` named `x,y,z`. Since for the user is mainly useful to define a stiffness matrix diagonal (so defining a total decoupled stiffness map), this type of message is used setting the diagonal of K_o in `angular` and the diagonal of K_t in `linear`.

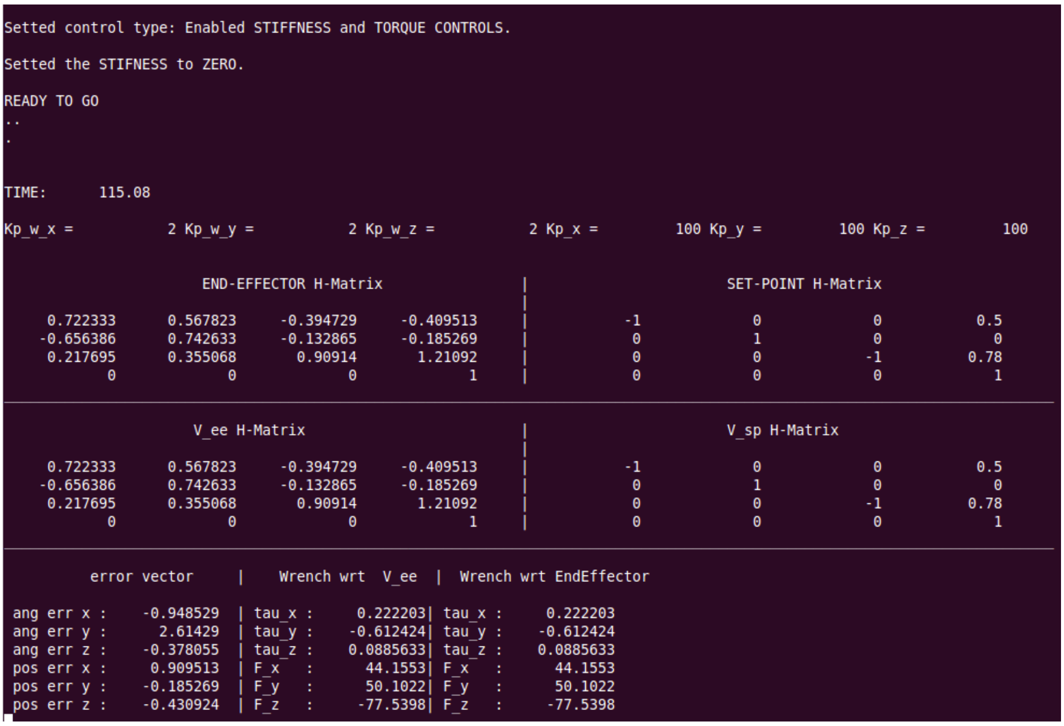
Publisher:

`EndEffector_wrench`	`WrenchStamped`:	used to communicate to RViz (Sec: 5.2.5) the spring wrench with respect to Ψ_{ee}. `WrenchStamped` is a standard message belonging to the class `geometry_msgs` provided by ROS
`Vee_wrench`	`WrenchStamped`:	used to communicate to RViz the spring wrench with respect to $\Psi_{v(ee)}$
`EndEffectorPose`	`Cart_pose`:	used to communicate to the Supervisor node H_ee^0 for its initialization
`VeePose`	`Cart_pose`:	(see above)
`VspPose`	`Cart_pose`:	(see above)

When the IPC node is lunched, it set the control types and the stiffness of each joint and waits to receive for the first time the joint position through the topic `FRI_MsrJnt`. Once received it initialize the object `IPC` (class `IPC` generated from 20-Sim code generation). During initialization H_ee^0 is calculated and then Ψ_{ee}, Ψ_{sp}, $\Psi_{v(sp)}$ are set at the identity of Ψ_{ee}. After initialization it starts to run on a while cycle at a frequency of 500Hz. The pseudo-code is here below:

```
...

%callBackFunctions
...


void main()
{
    %node initialization
    ...
    %declaration of the node frequency

    ros::Rate r(500);

    %publishers and subscribers declaration
    ...

    %creation of the IPC object

    controller IPC;

    %creation of the message FRI_Control and setting true the stiffness and torque commands.
    %Then send it to the FRI node

    Fri_Control ControlType = [false, false, true, false, false];
    toFRI_control.publish(ControlType);

    %creation of the message Empty_Jnt_stiff and then send it to the FRI node

    Jnt_stiff Empty_Jnt_stiff;
    to_FRI_stiffness.publish(Empty_Jnt_stiff);

    %waiting for the first joint position received

    while (first_recieved == false){}

    %inizialization of the IPC

    IPC.initialize();

    %start the cycle

    while (ros::ok())
    {
        %calculation of wrenches, torques to apply and frames position
        IPC.calculate();
        toFRI_torque.publish(Jnt_torque);
        toAll_ee_wrench.publish(ee_wrench);
        toAll_vee_wrench.publish(v_ee_wrench);
        toAll_ee_pose.publish(ee_pose);
        toAll_vee_pose.publish(v_ee_pose);
        toAll_vsp_pose.publish(v_sp_pose);
        print();
    }

    return 0;
}
```

Notice the command `ros::Rate r(500)`. A so high frequency is needed because we have to simulate a continuous energy interconnection between IPC and plant. Higher the frequency is more accurate this simulation is.
`print()` function is a function that print on the terminal some information as the time, the values of diagonal of K, H^0_{ee}, H^0_{sp} , $H^0_{v(ee)}$, $H^0_{v(sp)}$, the position error between $\Psi_v(ee)$ and $\Psi_v(sp)$ as a 6×1 vector $[\delta\theta^T, \delta p^T]^T$ that express the position of $\Psi_{v(ee)}$ with respect to $\Psi_{v(sp)}$ in a vector form instead of the matrix form $H^{v(sp)}_{v(ee)}$ (the calculation of $\delta\theta$ is made with particular function `from_rot_to_omega` that implement



```
Setted control type: Enabled STIFFNESS and TORQUE CONTROLS.

Setted the STIFNESS to ZERO.

READY TO GO
..
.


TIME:      115.08

Kp_w_x =          2 Kp_w_y =          2 Kp_w_z =          2 Kp_x =        100 Kp_y =        100 Kp_z =        100


                   END-EFFECTOR H-Matrix                |                    SET-POINT H-Matrix
                                                        |
    0.722333      0.567823     -0.394729     -0.409513  |           -1             0             0           0.5
   -0.656386      0.742633     -0.132865     -0.185269  |            0             1             0             0
    0.217695      0.355068       0.90914       1.21092  |            0             0            -1          0.78
           0             0             0             1  |            0             0             0             1
-----------------------------------------------------------------------------------------------------------------
                   V_ee H-Matrix                        |                    V_sp H-Matrix
                                                        |
    0.722333      0.567823     -0.394729     -0.409513  |           -1             0             0           0.5
   -0.656386      0.742633     -0.132865     -0.185269  |            0             1             0             0
    0.217695      0.355068       0.90914       1.21092  |            0             0             0          0.78
           0             0             0             1  |            0             0             0             1
-----------------------------------------------------------------------------------------------------------------

          error vector     |   Wrench wrt  V_ee  |  Wrench wrt EndEffector

 ang err x :    -0.948529  | tau_x :     0.222203| tau_x :     0.222203
 ang err y :      2.61429  | tau_y :    -0.612424| tau_y :    -0.612424
 ang err z :    -0.378055  | tau_z :    0.0885633| tau_z :    0.0885633
 pos err x :     0.909513  | F_x   :      44.1553| F_x   :      44.1553
 pos err y :    -0.185269  | F_y   :      50.1022| F_y   :      50.1022
 pos err z :    -0.430924  | F_z   :     -77.5398| F_z   :     -77.5398
```

Figure 5.11. Screen of the printed information to the terminal by IPC node

a Cayley-Hamilton method explained in [14]; this calculates $\delta\theta$ form the rotation matrix $R^{v(sp)}_{v(ee)}(t)$), the wrenches $W^{v(ee)}$ and W^{ee}.
The layout of what is printed to the terminal is Fig. 5.11.

5.2.4 Supervisor nodes

The Supervisor node has to send to the IPC nodes the matrices $H^{ee}_{v(ee)}$, H^{0}_{sp} and $H^{sp}_{v(sp)}$ using the topics `SetVee`,`SetPoint`,`SetVsp` and the stiffness matrix K through the topic `SetStiffness`. This is done by the user that communicate with the IPC node through the following four distinct nodes

5.2.4.1 v_ee node

`v_ee` node is used to set $H^{ee}_{v(ee)}$. Once lunched, it first wait to receive the actual H^{0}_{ee} and $H^{0}_{v(ee)}$.Then it calculates the initial $H^{ee}_{v(ee)}t_i$, memorize it and the instant in which it has been received and start ask to the user to set the position and the orientation of $\Psi_{v(ee)}$ with respect to Ψ_{ee} to communicate to the IPC node. Its subscribed to and publish on the following topics:

Subscribers

`VeePose`

`EndEffectorPose`

Publisher:

`SetVee`

With the function `from_rot_to_omega` [14] the node calculates the rotational part of the vector form corresponding to the initial $H^{ee}_{v(ee)}(t_i)$ that will be called $s^{ee}_{v(ee)}(t_i) = [(\delta\theta^{ee}_{v(ee)})^T(\delta p^{ee}_{v(ee)})^T]$.
The user has to give as input the coordinates of the position of $\Psi_{v(ee)}$ with respect to Ψ_{ee} (that are p_x, p_y, p_z) and the desired orientation setting the rotation angles around the axis in the order $\theta_x, \theta_y, \theta_z$. These 6 paramenters are memorized in a 6 by one vector.
Furthermore, since the movement from one configuration to another cannot happen instantaneously (it would be not physically consistent, since it would means that the supervisor inject an infinite amount of power in an infinitesimal interval), once the user finishes to set the desired position of $\Psi_{v(ee)}$ it has to set also the time T in which he want to reach it. The desired position is then $s^{ee}_{v(ee)}(t_f)$ with $t_f = t_i + T$, where t_i is updated to the instant in which the user finish to input the data. The node generate a five degrees polynomial trajectory from the initial to the final (desired) configuration (as said just above, the node calculates the vector $s^{ee}_{v(ee)}(t_i)$ and then each its elements has a corresponding one in $s^{ee}_{v(ee)}(t_f)$). The initial and final velocity and acceleration, needed by the trajectory function, are setted to zero. This function give back the vector configuration $s^{ee}_{v(ee)}(t)$ corresponding to the actual istant $t \in [t_i, t_f]$
Each cycle the node convert $s^{ee}_{v(ee)}(t)$ in $H^{ee}_{v(ee)}(t)$, using the function `fromomega_torot` (that is the inverse of `from_rot_to_omega`) to calculate $R^{ee}_{v(ee)}$ from $\delta\theta^{ee}_{v(ee)}(t)$) and publish on the topic `SetVee` the position corresponding to the one generated by the trajectory function.
The pseudo-code of this node is:

```
...
bool done=true;
ros::Time initial_time;
ros::Time present_time;
ros::Time final_time;
ros::Time duration;
...

%callBackFunctions
...
```

```
void trajectory_generation()
{
    present_time = ros::Time::now()()

    if ( present_time <= final_time){

        \% trajectory calculation
        ...

        return;
    }

    done = true;

}

void main()
{
    %node initialization
    ...

    %publishers and subscribers declaration
    ...


    while (firsts_recieved == false){}

    %calculation of the initial position of v(ee)
    ...
    from_rot_to_omega();

    %start the cycle
    while (ros::ok())
    {
        if (done == true){
            set_desired_vee();
            initial_time = ros::Time::now()();
            done = false;
        }
        if (done == false){
            trajectory_generation();
        }
        from_omega_to_rot();
        SetVee.publish(Vee_pose);
    }
    return 0;
}
```

5.2.4.2 v_sp node

v_sp node has the same task of v_ee node but with respect the frames $\Psi_{v(sp)}$ and Ψ_{sp}.

Then its working flow is the same but the topic for on which it publish and the topics to which it is subscribed are:

Subscribers

`VspPose`

`SetPoint`

Publisher:

`SetVsp`

5.2.4.3 v_ee_and_sp node

Eventually the user would like to change $H^{ee}_{v(ee)}$ and $H^{ee}_{v(ee)}$ in the same way. This is allowed by the node `v_ee_and_sp` . In this case both $H^{ee}_{v(ee)}(t_i)$ and $H^{sp}_{v(sp)}(t_i)$ has to be calculated during the initialization, and treat $H^{ee}_{v(ee)}(t)$ and $H^{sp}_{v(sp)}(t)$ separately, at least during the first trajectory generation "cycle". After this, infact, $H^{ee}_{v(ee)}$ and $H^{ee}_{v(ee)}$ are the same, and could be treated in the same way.
The structure of this node is the same as the previous, but the subscribers and publishers are the sum of the nodes `v_ee` and `v_sp` :

Subscribers

`VeePose`

`VspPose`

`EndEffectorPose`

`SetPoint`

Publisher:

`SetVee`

`SetVsp`

5.2.4.4 setpoint node

The `setpoint` node allows to move the set the matrix H^0_{sp}. Again, it works like the others "point-generation" node, but the initial $H^0_{sp}(t_i)$ is set equal to the value of H^0_{ee} at the istant in which the node is launched. The message corresponding to $H^0_{sp}(t)$ is`SetPoint`. Topics involved are:

Subscribers

`EndEffectorPose`

Publisher:

`SetPoint`

5.2.5 RViz node

To understand and literally see where the various frames are positioned in the word, the 3D visualization tool of ROS called RViz [9] is used.
RViz allows the visualization of a 3D model of a robotic system, and the replication of the movements of the real one, through the use of the urdf files. To understand their syntax see [13].

Basically, in these files the model of the robotic system is built specifying its joints (and their type, as translational or rotational) and links and their spatial relation (the displacement and RPY angles between the joint frames are required, that can be obtained from the MDH paramenters, Fig.(4.2)). Furthermore, since it is a ROS tool, it is possible to communicate to RViz through specified topics. In particular, RViz automatically open and listen to the topic `joint_states`, interpreting the default ROS message type `sensor_msgs::JointState`, to set the value of the joint position of the 3D model. This kind of message is structured with three array of the dimension equal to the number of the non-fixed joint of the model:

- `name[]` (its elements are strings that have to be equal to the names of the joints defined in the urdf file);
- `position[]` (its elements are `floats64` corresponding to the positions of the joints);
- `effort[]` (it contains the torque applied by the actuator of each joint, eventually it can be not used).

Therefore, the `name[]` array is set during the node initialization and never changed again, while `position[]` is updated every time the new position of the joints is received. To visualize frames that does not belong to the model structure, that are Ψ_{ee}, $\Psi_{v(ee)}$, Ψ_{sp} and $\Psi_{v(sp)}$, it is possible to use tools provided by the standard ROS package `tf`. The type of message used is the standard `geometry_msgs::TransformStamped` that contains:

- the `frame_id`, that is the name of the frame with respect to the frame that has to be visualized is expressed;
- the `child_frame_id`, that is the name of the published frame;
- the reference time, that is the intant in which the message is published;
- the position of the frames, expressed in a 3 by one vector for the position and a quaternion for the orientation.

These messages, one for each ”floating frame”, are published on a special topic, called `broadcaster`, always provided by the `tf` package.
It follows that a new node, here called `rviz_state_publisher` is used in order to convert the messages used by the other nodes in the format explained above. It will be subscribed to the topics:

Subscribers

`SetEndEffector`

`SetPoint`

`SetVee`

`SetVsp`

`FRI_MsrJnt`

In the CallBack functions of the first four topics the transformation is done. For example, concerning the end effector pose:

```
...
geometry_msgs::TransformStamped EndEffector_pose;
...
SetEndEffectorCallBack( const Cart_pose::ConstPtr& msg)
{
    Cart_pose SetEndEffector_cart_pose=(*msg);

    EndEffector_pose.header.stamp = ros::Time::now();

    EndEffector_pose.transform.translation.x = SetEndEffector_cart_pose.Cart_pose[3];
        EndEffector_pose.transform.translation.y = SetEndEffector_cart_pose.Cart_pose[7];
        EndEffector_pose.transform.translation.z = SetEndEffector_cart_pose.Cart_pose[11];

        R[0] = SetEndEffector_cart_pose.Cart_pose[0];
        R[1] = SetEndEffector_cart_pose.Cart_pose[1];
        R[2] = SetEndEffector_cart_pose.Cart_pose[2];
        R[3] = SetEndEffector_cart_pose.Cart_pose[4];
        R[4] = SetEndEffector_cart_pose.Cart_pose[5];
        R[5] = SetEndEffector_cart_pose.Cart_pose[6];
        R[6] = SetEndEffector_cart_pose.Cart_pose[8];
        R[7] = SetEndEffector_cart_pose.Cart_pose[9];
        R[8] = SetEndEffector_cart_pose.Cart_pose[10];

        fromRotationToQuaternion( R );

        EndEffector_pose.transform.rotation.x = q.x();
        EndEffector_pose.transform.rotation.y = q.y();
        EndEffector_pose.transform.rotation.z = q.z();
        EndEffector_pose.transform.rotation.w = q.w();
}
```

while for the callback of `FRI_MsrJnt`:

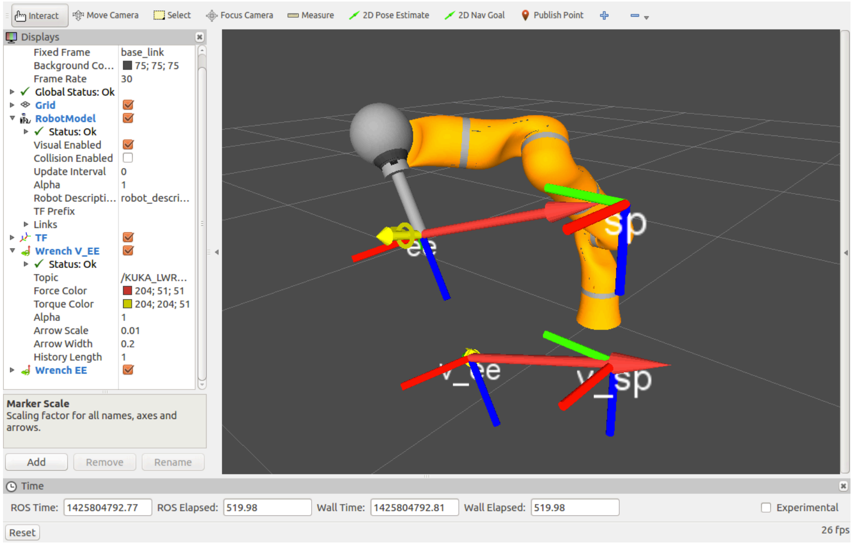
```
...
sensor_msgs::JointState Rviz_Jnt_state;
...
jointPoseCallback(const Jnt_pos::ConstPtr& msg)
{
    Jnt_pos Msr_Jnt_pos==(*msg);

    for( i = 0; i < 7; i++)
    {
        Rviz_Jnt_state.position[i] = Msr_Jnt_pos[i];
    }
}
```

The pseudo-code of the `rviz_state_publisher` node is:

```
...
%Callback functions

void main()
{
    %node initialization
    ...
    ros::Rate r(20);

    %publishers and subscribers declaration
    ...
    while(ros::ok())
    {
        ToRVIZ_jointstate.publish(Rviz_Jnt_state);

                br.sendTransform(SetPoint_pose);
                br.sendTransform(EndEffector_pose);
                br.sendTransform(v_ee_pose);
                br.sendTransform(v_sp_pose);
    }
}
```

Notice that the rate of this node is quite low, since it is used just for visualization and it is not necessary to execute it at a high frequency (and a high rate could busy the CPU too much, affecting the performances of the controller). Since the transformation of messages are done directly in the callbacks function, only publication

of the various messages are required in the while loop.
The last things visualized by RViz are the wrenches imposed by the IPC with respect the frames Ψ_{ee} and $Psi_{v(ee)}$. A screenshot that show how RViz looks like follows:



Fixed Frame	base_link
Background Co...	75; 75; 75
Frame Rate	30
✓ Global Status: Ok	
Grid	☑
RobotModel	☑
✓ Status: Ok	
Visual Enabled	☑
Collision Enabled	□
Update Interval	0
Alpha	1
Robot Descripti...	robot_descri...
TF Prefix	
Links	
TF	☑
Wrench V_EE	☑
✓ Status: Ok	
Topic	/KUKA_LWR...
Force Color	🟥 204; 51; 51
Torque Color	🟨 204; 204; 51
Alpha	1
Arrow Scale	0.01
Arrow Width	0.2
History Length	1
Wrench EE	☑

Marker Scale
Scaling factor for all names, axes and arrows.

ROS Time: 1425804792.77 ROS Elapsed: 519.98 Wall Time: 1425804792.81 Wall Elapsed: 519.98 □ Experimental

26 fps

Figure 5.12. Screenshot of the RViz interface

A scheme of the interaction between all these ROS nodes is given in Figure 5.13. Notice that is really similar to the control scheme of the IPC (Fig. 5.7)

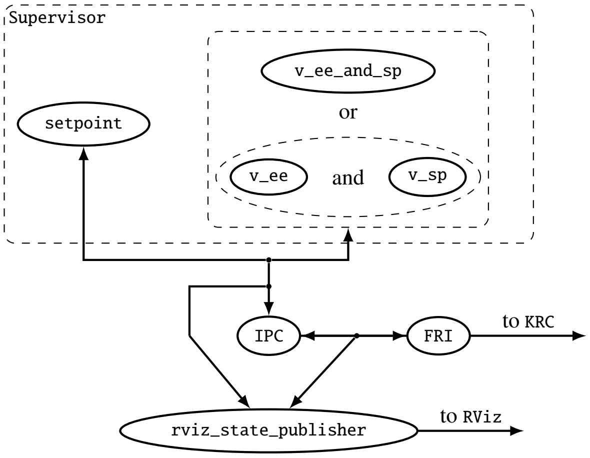


Figure 5.13. Nodes comunication

Chapter 6

Analysis of the RCC

The *Remote Center of Compliance* device (RCC) was designed in principle to assign some compliance to the end effector of robots that had to task a classical *peg in hole*. RCC was necessary because first robotic arms didn't allow to reach a compliant behaviour. Therfore, to compensate some misalignment between the peg and the hole, a passive compliant device attached between the tip of the robot and the peg was needed. A scheme of the working principle is Figure (6.1)

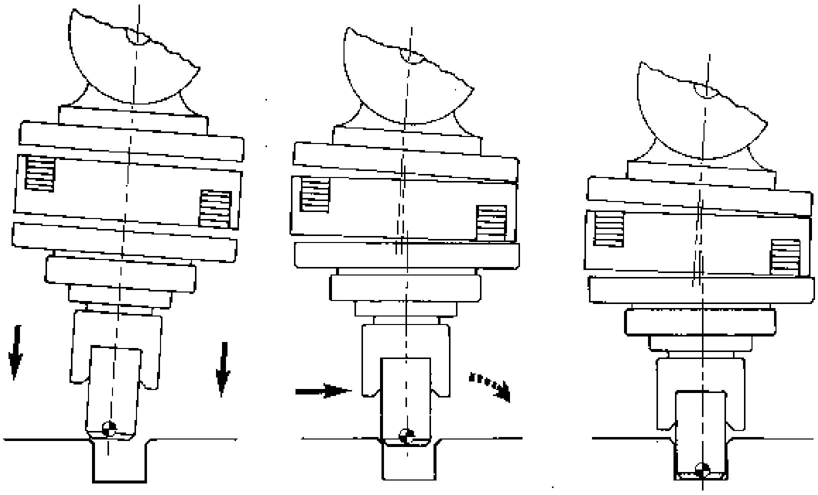


Figure 6.1. The Remote Center of Compliance

To understand how exactly it is designed, it is suggested to have a look at [3].

Basically, it consists of two parallel rigid plates, linked with a system of springs and dampers. These springs are designed and regulated in such a way that the resulting total *center of stiffness* (corresponding to the center of compliance) is coincident with the tip of the peg (see Figure 6.1. Furthermore, the stiffness matrix is diagonal with quite low lateral stiffness and high stiffness along the vertical axis of the RCC and around all the axis.
In this way, a lateral misalignment between the peg and the hole is compensated by the RCC, helped also by the slope (that in Figure 6.1 is either on the edge of the peg and on the edge of the hole), whose lower plate slides allowing the insertion. Slope can be only on one of the two edges.

6.1 Replication of the RCC

Since the IPC is passive and it is possible to set the center of stiffness and its relative stiffness matrix K, it is possible to intrinsically replicate the behaviour of the RCC. To be precise, it is possible to set the stiffness with respect to $\Psi_{v(ee)}$. Therefore, the stiffness matrix that the user set is indicated as $K^{v(ee)}$. Remember that it is defined with respect to the configuration corresponding to the identity between $\Psi_{v(ee)}$ and $\Psi_{v(sp)}$. Generally speaking, to replicate the RCC, $K^{v(ee)}$ has to be:

$$K^{v(ee)} = \begin{bmatrix} k_{o,x} & 0 & 0 & 0 & 0 & 0 \\ 0 & k_{o,y} & 0 & 0 & 0 & 0 \\ 0 & 0 & k_{o,z} & 0 & 0 & 0 \\ 0 & 0 & 0 & k_{t,x} & 0 & 0 \\ 0 & 0 & 0 & 0 & k_{t,y} & 0 \\ 0 & 0 & 0 & 0 & 0 & k_{t,y} \end{bmatrix}; \qquad k_{o,x}, k_{o,y} k_{o,z}, k_{t,z} \text{ big} \quad k_{t,x}, k_{t,y} \text{ small} \tag{6.1}$$

Set $H^{ee}_{v(ee)} = H^{sp}_{v(sp)} = I$ means that $K^{ee} = K^{v(ee)}$, and therefore the same behavior of the RCC is achieved.

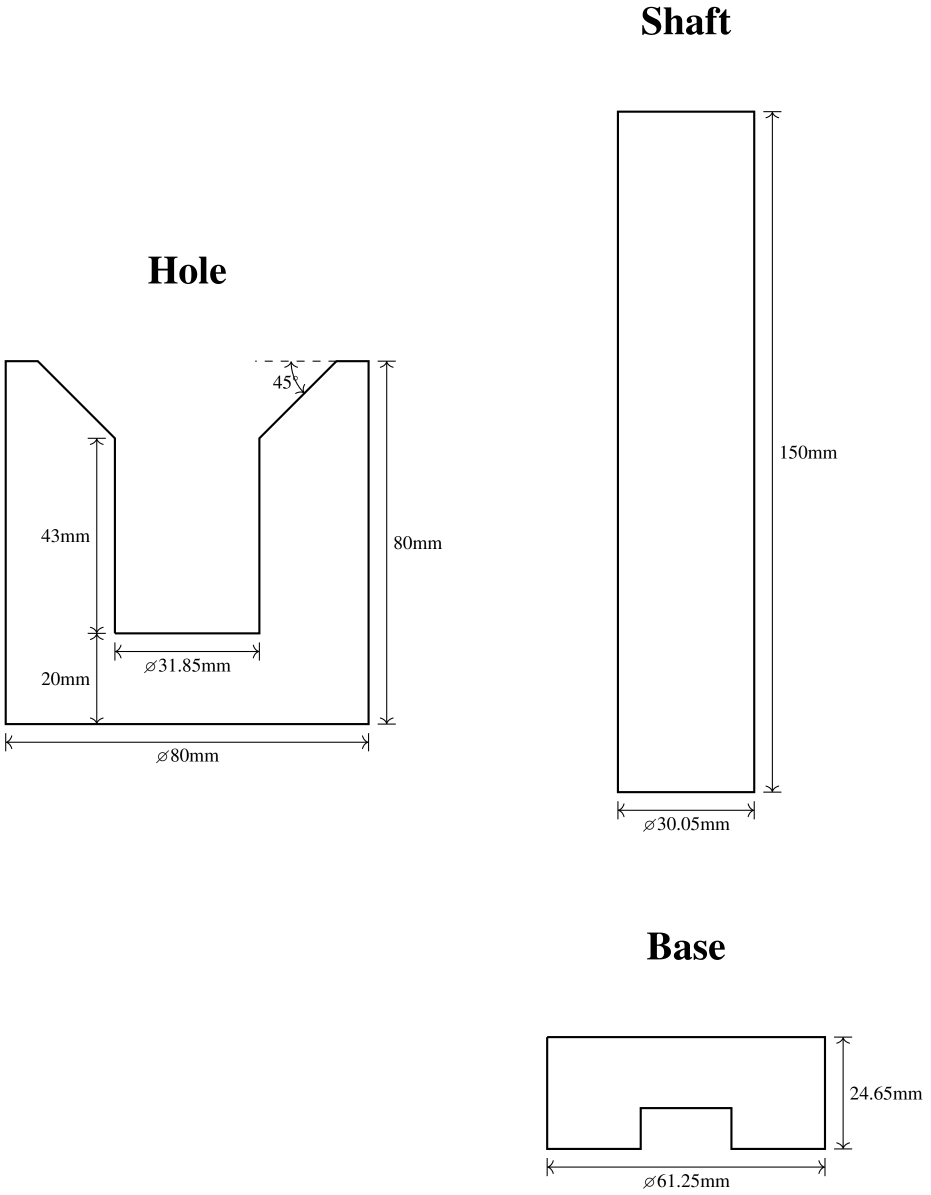


Figure 6.2. Shapes and quotes of the **hole** (left) and of the peg, composed by the **hole** and **hole** (right).

The simulation has been done using a peg attached to the tip of the KUKA LWR4+ composed by the base and the shaft corresponding to the ones shown in Figure (6.2). Always in Fig. (6.2) it is indicated the shape of the used hole.
The difference between the hole diameter and the peg diameter (to be precise, the diameter of the shaft) is 1.8mm .
The total height of the peg is $h_{peg} = 0.17465$m. Considering also the offset between

the frame of the 7-th joint and the tip of the arm (see Fig. (4.2), $H^7_{(ee)}$ was set by the supervisor at:

$$H^7_{(ee)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.17465 + 0.078 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.25265 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.2}$$

During simulation, the position of the bottom of the hole with respect to the base of the KUKA LWR+ was:

$$p^0_{hole} = \begin{bmatrix} x^0_{hole} \\ y^0_{hole} \\ z^0_{hole} \end{bmatrix} = \begin{bmatrix} -0.5783 \\ 0.0105 \\ 0.0206 \end{bmatrix} \text{m} \tag{6.3}$$

To create the desired misagnment the positions along x^0 and y^0 of the setpoint frame Ψ_{sp} were constant and set to:

$$x^0_{sp} = 0.02\text{m}$$
$$y^0_{sp} = 0.57\text{m}$$

Simulations has been done creating a cyclic "up and down", with a trajectory on z going from 0.08m to 0.02m in the following way:

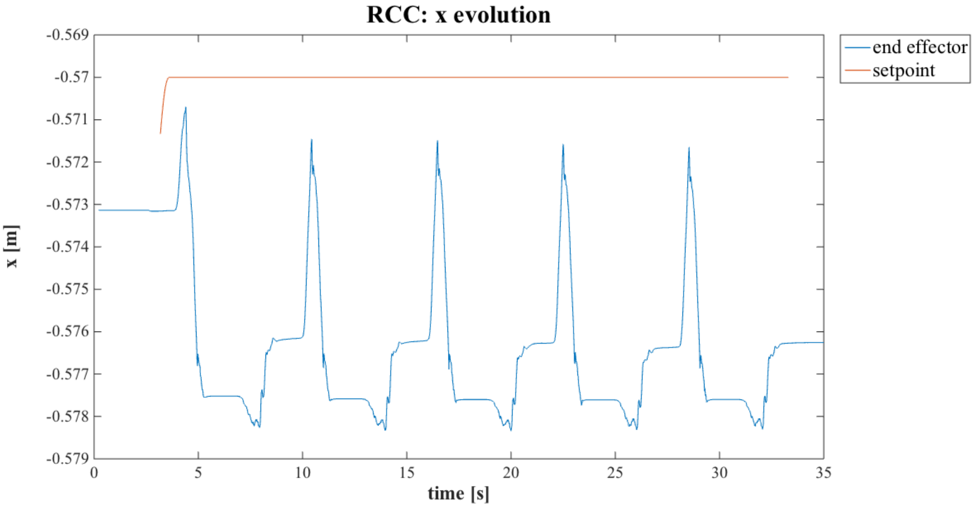
```
%start cycle

1 second to 0.08m;
from 0.08m to 0.02m in 2 seconds with a 5th degree polynomial;
1 second to 0.02m;
from 0.02m to 0.08m in 2 seconds with a 5th degree polynomial;

%end cycle
```

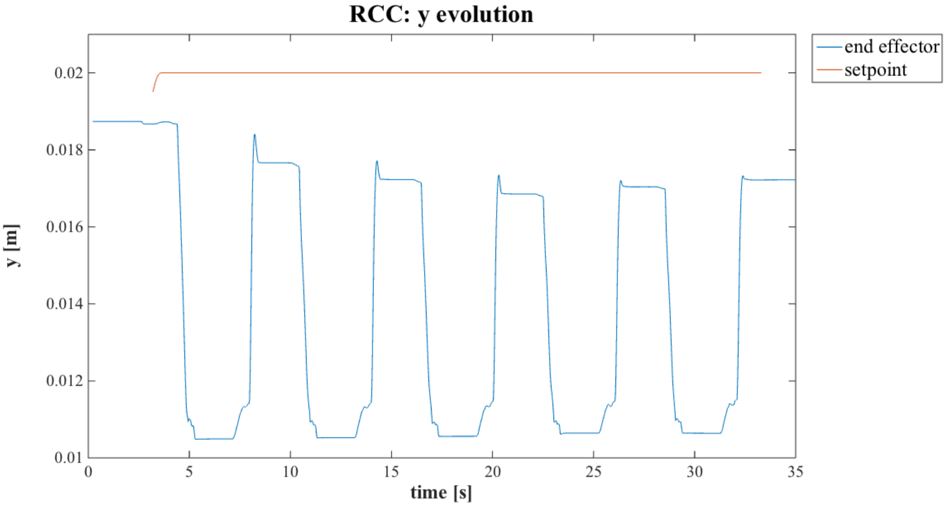
Notice that the end value of the trajectory is a little bit lower than the real bottom altitude of the peg. This is done to be sure that the arm always push to go down.

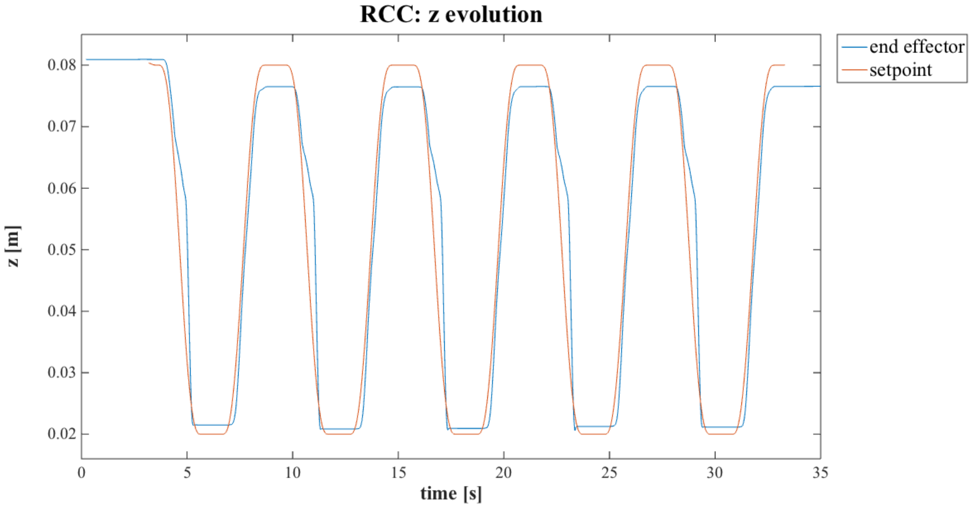
The replication of the RCC was made with the stiffness matrix

$$K^{v(ee)} = K^{ee} = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 400 & 0 & 0 \\ 0 & 0 & 0 & 0 & 400 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1000 \end{bmatrix} \tag{6.4}$$

Remember that the orientation terms have a mesure unit Nm/rad, while the translational ones are N/m. Therefore, 100 is a really high value for orientation terms, while 400 is quite low for the translations.
Results corresponding to this stiffness matrix and with $H^{ee}_{v(ee)} = H^{sp}_{v(sp)} = I$ are shown in the following plots.

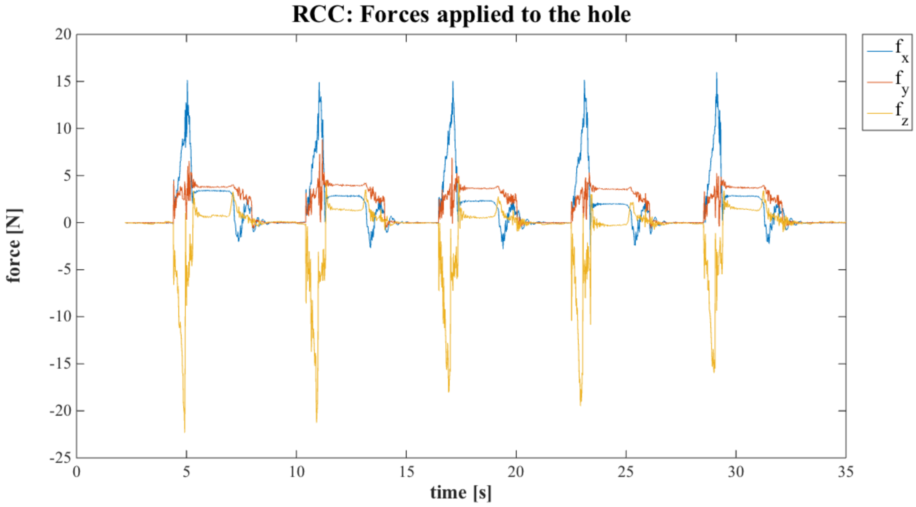


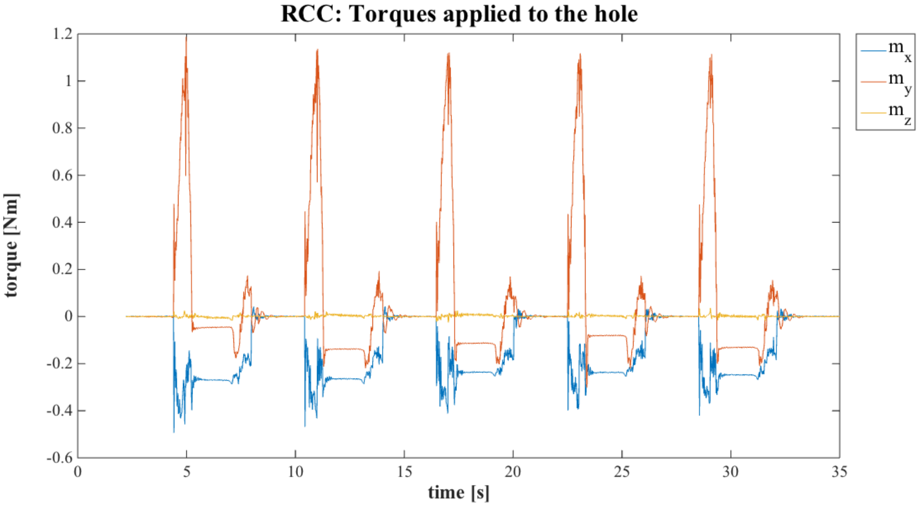




As plots show, the misalignment on x and y is always ”compensated” (look at the blue lines on the relative plots) and simultaneously the peg is allowed to descend along z. The . The task is always completed.
The two next plots show the forces along the axis and the torques around the same axis respectively, in order to understand the amount of stress that the arm apply to the hole. These will be usefull later, to compare other results.

As shown above, the replication of the RCC works pretty good. Anyway the





stress on the hole could be too high (in particular the forces at which it is subjected), expecially if the material is soft and delicate. A solution could be define $H^{ee}_{v(ee)}$ and $H^{sp}_{v(sp)}$ different from the identity, moving $\Psi_{v(ee)}$ and $\Psi_{v(sp)}$ along the z axis of Ψ_{ee} and Ψ_{sp} respectively.

$$H^{ee}_{v(ee)} = H^{v(sp)}_{sp} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.5}$$

where with z is indicated the distance along $\hat{z}_{v(ee)}$ and $\hat{z}_{v(sp)}$ between the frames Ψ_{ee}

and $\Psi_{v(ee)}$ and the frames Ψ_{sp} and $\Psi_{v(sp)}$.
From Equations (5.11), the relation between K^{ee} and $K^{v(ee)}$ is given by means of the Adjoint map $Ad_{H_{ee}^{v(ee)}}$.
Consider always a $K^{v(ee)}$ diagonal as in Eq. (6.1). It follows that applying the homogeneous matrix of Eq. (6.5) to Eq. (5.11):

$$Ad_{H_{ee}^{vee}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -z & 0 & 0 & 1 & 0 \\ z & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.6}$$

$$K^{ee} = \begin{bmatrix} k_{o,x} + z^2 k_{t,y} & 0 & 0 & 0 & z k_{t,y} & 0 \\ 0 & k_{o,y} + z^2 k_{t,x} & 0 & -z k_{t,x} & 0 & 0 \\ 0 & 0 & k_{o,z} & 0 & 0 & 0 \\ 0 & -z k_{t,x} & 0 & k_{t,x} & 0 & 0 \\ z k_{t,y} & 0 & 0 & 0 & k_{t,y} & 0 \\ 0 & 0 & 0 & 0 & 0 & k_{t,y} \end{bmatrix} \tag{6.7}$$

Such a positioning between Ψ_{ee} and $\Psi_{v(ee)}$ induces a stiffness matrix K^{ee} with orientational stiffness around $\hat{x}_{ee}$ and $\hat{y}_{ee}$ that are greater than the of K^{ee}, proportional to the square of the distance z, while the translational terms $k_{t,i}$ remain the same. Furthermore, K_c (see Eq. (5.15)) is no more null.
Let's see how this kind of configuration can influence the *peg in hole* task.
To compare this case with the previous one, that is the normal RCC, it would be nice to have similar values on the diagonal of K^{ee}. The idea is to set a $K^{v(ee)}$ with diagonal terms not null, in order to always be able to control either the position and the orientation. In this way, thinking about a real use of it, it is possible to properly control the robot during free motion, setting $H_{v(ee)}^{ee} = I$, and once in proximity of the

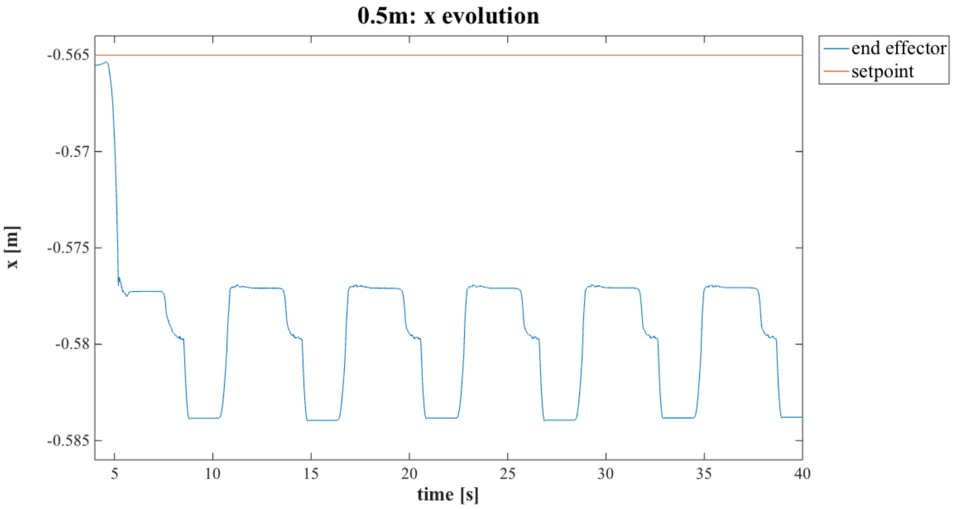
hole, move $\Psi_{v(ee)}$ along $\hat{z}_{ee}$ of the desired value. Simulations were made using:

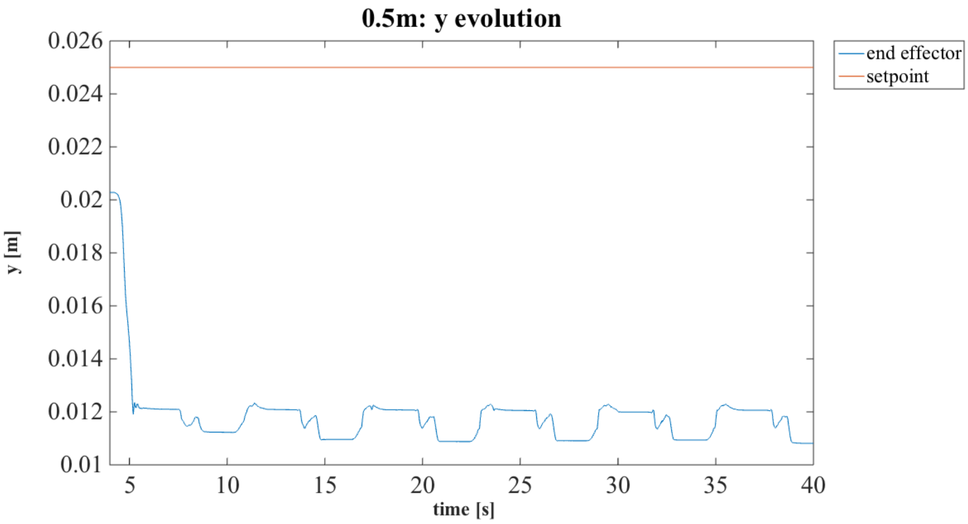
$$K^{v(ee)} = \begin{bmatrix} 5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 400 & 0 & 0 \\ 0 & 0 & 0 & 0 & 400 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1000 \end{bmatrix} \tag{6.8}$$

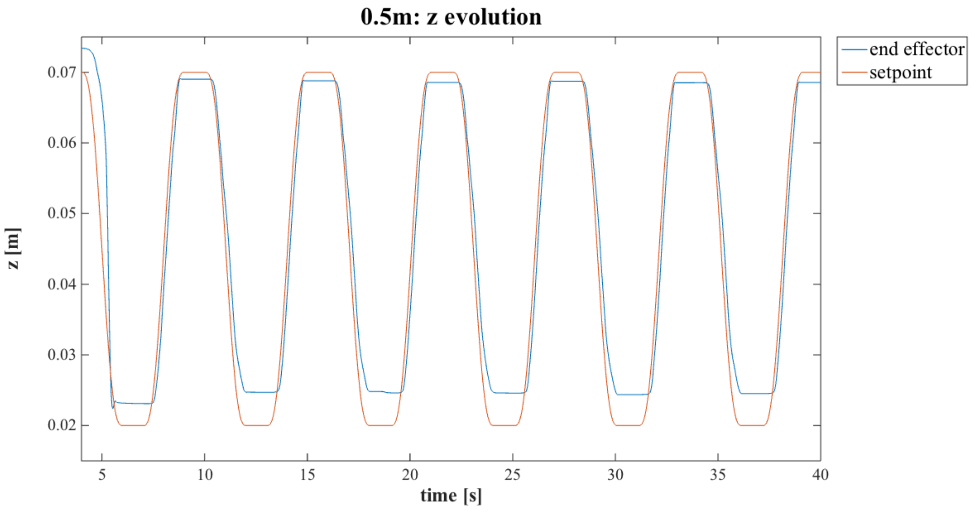
If then z is set to 0.5m (that is the distance position of the origin of Ψ_{ee} along $\hat{z}_{v(ee)}$):

$$K^{ee} = \begin{bmatrix} 105 & 0 & 0 & 0 & 200 & 0 \\ 0 & 105 & 0 & -200 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & -200 & 0 & 400 & 0 & 0 \\ 200 & 0 & 0 & 0 & 400 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1000 \end{bmatrix} \tag{6.9}$$

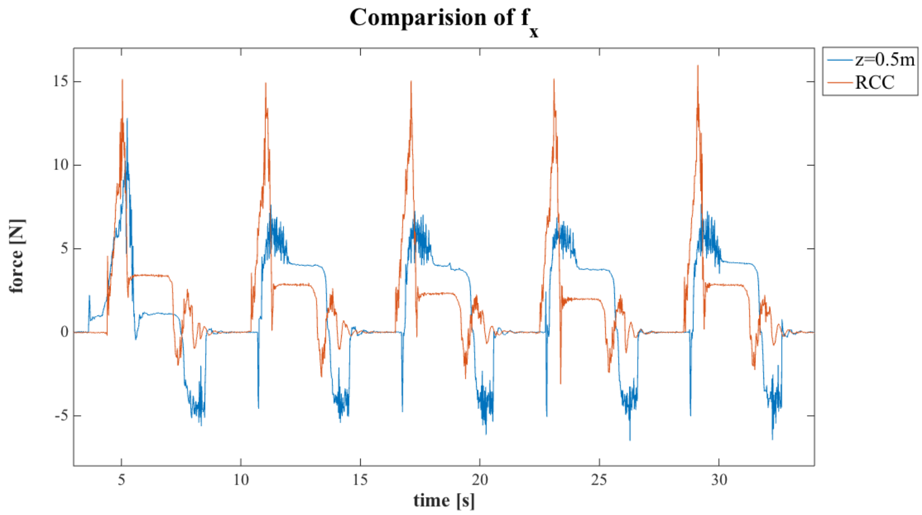
Its diagonal is almost the same of the one of Eq. 6.5 (out of 5 for the orientational terms). Here below the results of the simulation of this setting are shown.

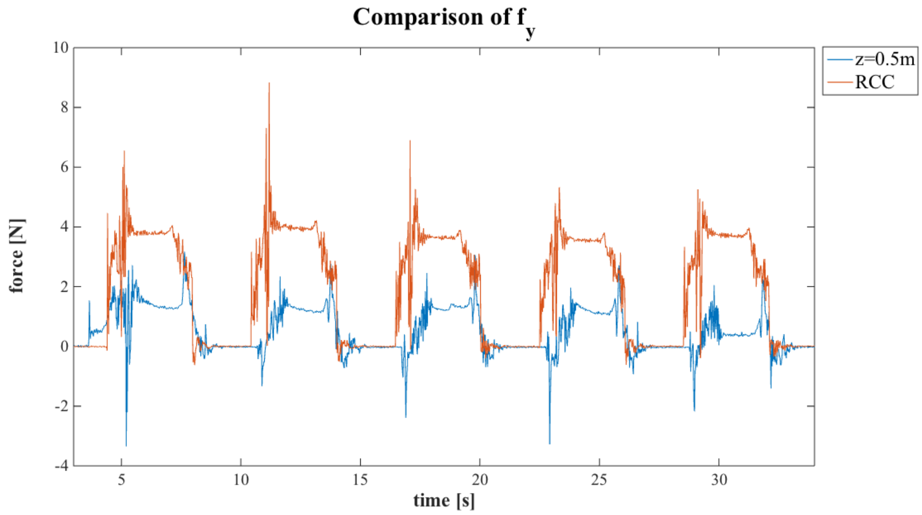


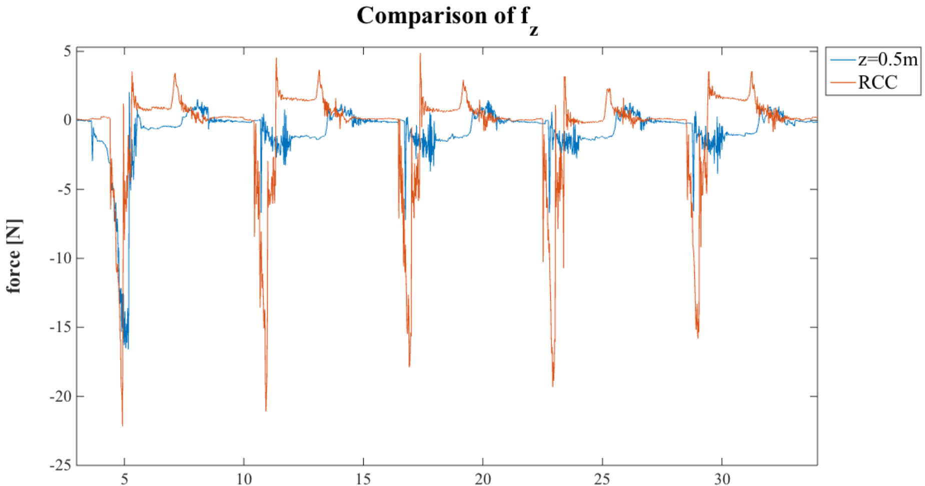


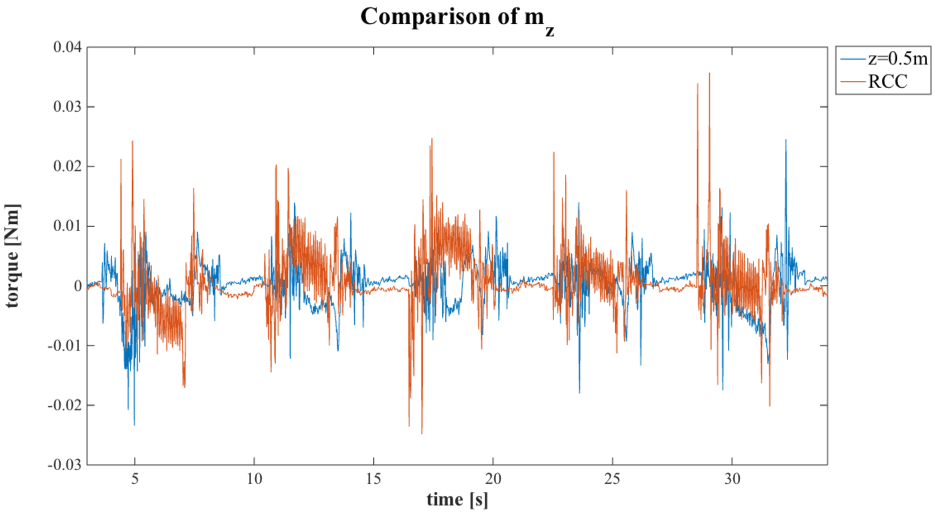


The peg in hole is done also in this case. Anyway, there are some losses along z (see plot). Indeed, the lowest position the peg reaches this time is higher than the previous, and then it does not reaches perfectly the bottom of the hole.
But look at the following plots, that compare each force and torque applied to the hole in the two different cases.


Comparision of f_x
z=0.5m
RCC
force [N]
15
10
5
0
-5
time [s]
5
10
15
20
25
30


Comparison of f_y
z=0.5m
RCC
force [N]
10
8
6
4
2
0
-2
-4
time [s]
5
10
15
20
25
30


Comparison of f_z
z=0.5m
RCC
force [N]
5
0
-5
-10
-15
-20
-25
time [s]
5
10
15
20
25
30



The maximum absolute values of each force/torque in the last case are always lower with respect to the "original" RCC. This happen thanks to the coupling terms created setting the distance on the z axis different from zero.
Only m_z could be considered the same, or at least really similar. This result is completely consistent with the analysis of the stiffness matrices. Indeed, only the relations between m_z and f_z and the displacements in K^{ee} do not change with respect the ones in $K^{v(ee)}$. The reason why also f_z is lower is that it is influenced by the reactions along x and y axis.
To conclude, more z is high more the performances increase: the evolution of z follows better the setpoint and the stress on the hole is lower.

This results show that, within this technique it is possible to act a peg in hole decreasing the stress imposed to the hole. Anyway, a compromise with the performance on position is needed.

Conclusions

The work of this thesis is finished. A fully analysis of the port-Hamiltonian framework was done. It was shown that this framework leads to an energetic interpretation of systems and their interactions. This approach completely reflects what happens in nature. Therefore, approaching the problem of the control of a plant in this framework means to understand which kind of behaviour is desired and that is possible to achieve it always through the energetic approach, by means of control by *energy shaping* and *damping injection*. Port-Hamiltonian framework allows also to easily introduce all the concepts belonging to the Passivity Theory. If these are applied to the control issues by means of the *control by interconnection*, it is possible to control the behavior of the plant such that it is ALWAYS passive, that is the IPC. The main advantage given by this is that it is possible to ensure an intrinsically safety behavior of the robot system. The anylisis of the replication of the Remote Center of Compliance through the IPC shows that such devices are no more necessary if it is possible to control the arm with ”torque controls”. Finally, using the features of the IPC in a clever and proper way, it has been shown that RCC can be improved in the forces applied to the hole. This could be really useful if the task of the peg-in-hole has to take place with a soft hole. Regulating the IPC configuration, it is possible to decrase as much as necessary the stress to which the hole is subjected.

Further works could be made in order to understand how the behavior at the end effector changes as a function of the configuration of the IPC, through the interpretation of the stiffness matrix K^{ee}. What happens if a translation along another axis is operated? And if a rotation of $\Psi_{v(ee)}$ is made instead of a translation? And what if both are operated simultaneously? Other useful behaviors, as the one achieved with a tralsation along z, could be achieved also with respect to the forces applied to the environment.

Appendix A

Mathematical Background

In this Appendix several concept are defined. These are useful to understand the work did in this thesis. A deeper analysis and explanation of these topics can be found in [5] and [11].

A.1 Linear algebra

Definition A.1.1. Mappings Given two sets A and B, we call a map f from A to B and we denote it as:

$$f : A \to B; f(a) \mapsto b$$

an operation which associates to each element $a \in A$ a unique element $f(a) \in B$. The set A is called the domain of the map and B its co-domain.The set of all $b \in B$ such that there exists an $a \in A$ with $f(a) \mapsto b$ is called the *range* of f.

Definition A.1.2. Surjective, injective, bijective A mapping $f : A \to B$ is *surjective* if its range is equal to its co-domain. It is *injective* if for every b in its range there is exactly one $a \in A$ such that $f(a) \mapsto b$. A mapping is *bijective* if it is injective and surjective.

Definition A.1.3. Diffeomorphism A mapping $\phi : \mathbb{R}^n \to \mathbb{R}^n$ is a *diffeomorphism* if and only if it is bijective and ϕ and ϕ^{-1} are differentiable.

Definition A.1.4. Algebraic structure An *algebraic structure* is a pair (A, T) for which A is a set and T is an internal binary operator, which means T is a map of the following form:

$$T : A \times A \to A$$

Definition A.1.5. Identity element and inverse element The *identity element* for an algebraic structure (A, T) is an element $u \in A$ for which:

$$T(a, u) = T(u, a) = a \,\forall\, a \in A$$

Furthermore, given the identity element u, we say that $a \in A$ has inverse iff $\exists\, b \in A$, which is denoted by a^{-1} such that $T(a, a^{-1}) = T(a^{-1}, a) = u$.

Definition A.1.6. Associative and commutative property An algebraic structure satisfies the associative property iff:

$$T(T(a, b), c) = T(a, T(b, c)) \,\forall\, a, b, c \in A$$

while it satisfies the commutative property iff:

$$T(a, b) = T(b, a)) \,\forall\, a, b \in A$$

Definition A.1.7. Group An algebraic structure (A, T) that satisfies the associative property, it has an identity element and for each element in A there exist an inverse is called *group*. A group that satisfies the commutative property is called a *commutative* ora *Abelian group*.

Definition A.1.8. Vector Space A real vector space V is a set of elements (called vectors), one element called the identity (or zero-vector $\overrightarrow{0}$), and two operations $\oplus$ (addition of two vectors) and $\cdot$ (multiplication of a vector by a scalar), such that

- $\forall\, v_1, v_2 \in V$ also $v_1 \oplus v_2 \in V$
- $\forall\, v_1 \in V$ and $\forall\, x \in \mathbb{R}$ also $x \cdot v \in V$
- $\forall\, v_1 \in V$ there exist $v^{-1} \in V$ such that $v_1 \oplus v_2 = \overrightarrow{0}$

and such that the following properties holds for all $v_1, v_2, v_3 \in V$ and $x_1, x_2 \in \mathbb{R}$

- $1 \cdot v_1 = v_1$;
- $v_1 \oplus \overrightarrow{0} = v_1$;
- $x_1 \cdot (x_2 \cdot v_1) = (x_1 x_2) \cdot v_1$;
- $(v_1 \oplus v_2) \oplus v_3 = v_1 \oplus (v_2 \oplus v_3)$;

- $(x_1 + x_2) \cdot v_1 = (x_1 \cdot v_1) \oplus (x_2 \cdot v_2)$;
- $x_1 \cdot (v_1 \oplus v_2) = (x_1 \cdot v_1) \oplus (x_1 \cdot v_2)$.

Definition A.1.9. Dual vector space The *dual* space V^* of the vector space V is the space of the all linear mappings (called *co-vectors*) from V to $\mathbb{R}$, i.e all mappings $f : V \to \mathbb{R}$ such that $\forall v_i \in V$ and $\forall x_i \in \mathbb{R}$

$$f((x_1 \cdot v_1) \oplus \ldots \oplus (x_k \cdot v_k)) = x_1 f(v_1) + \ldots + x_k f(v_k)$$

Definition A.1.10. Dual product The *dual product* is the natural pairing of an element $v \in V$ and an element $f \in V^*$ as $\langle f|v\rangle := f(v) \in \mathbb{R}$. Often the elements belonging to V are called *flows*, while the ones belonging to V^* are called *efforts*.

Definition A.1.11. Tensor Given a vector space V and its dual V^*, a tensor T is a mapping of the form

$$T : \underbrace{V^* \times \cdots \times V^*}_{p \text{ times}} \times \underbrace{V \times \cdots \times V}_{q \text{ times}} \to \mathbb{R}$$

that is linear in all its arguments. The tensor T is said to have order $p + q$, order p contra-variant and order q co-variant, and is a type (p, q) tensor.

Definition A.1.12. Lie algebra A *Lie algebra* is a vector space together with a bilineary opearator $\left[.\right] : V \times V \to V$ (called *Lie bracket* or commuator), that satisfies the following properties for all $v_1, v_2, v_3 \in V$ and $x_1, x_2 \in \mathbb{R}$:

- bilinearity: $\begin{cases} \left[x_1 v_1 + x_2 v_2, v_3\right] = x_1\left[v_1, v_3\right] + x_2\left[v_2, v_3\right] \\ \left[v_1, x_1 v_2 + x_2 v_3\right] = x_1\left[v_1, v_2\right] + x_2\left[v_1, v_3\right] \end{cases}$
- skew-symmetry: $\left[v_1, v_2\right] = -\left[v_2, v_1\right]$
- Jacobi's identity: $\left[v_1, \left[v_2, v_3\right]\right] + \left[v_2, \left[v_3, v_1\right]\right] + \left[v_3, \left[v_1, v_2\right]\right] = 0$

A.2 Differential geometry

Definition A.2.1. Manifold A *manifold* is intuitively defined as a set which is locally diffeomorphic to $\mathbb{R}^n$ around each of its points.

Definition A.2.2. Tangent space Given a manifold $\mathcal{M}$, we define the tangent space $T_p\mathcal{M}$ to $\mathcal{M}$ at p as the linear space of mappings of the following form:

$$X_p : C^\infty \rightarrow \mathbb{R}$$

satisfying the following:

- Linearity: $X_p(\alpha f + \beta g) = \alpha(X_p f) + \beta(X_p g)$
- Leibniz rule: $X_p(fg) = (X_p f)g(p) + f(p)(X_p g)$

Definition A.2.3. Canonical Projection A *canonical projection* for the tangent space $T_p\mathcal{M}$ to the manifold $\mathcal{M}$ is a mapping of the following form:

$$\pi : T_p\mathcal{M} \rightarrow \mathcal{M}; (x, v) \mapsto x$$

Definition A.2.4. Vector field We call a smooth mapping of the following form a vector field:

$$X : \mathcal{M} \rightarrow T\mathcal{M}$$

Definition A.2.5. Co-vector field We call a smooth mapping of the following form a co-vector field:

$$X^* : \mathcal{M} \rightarrow T^*\mathcal{M}$$

Definition A.2.6. Lie group A Lie group $(\mathcal{M}, o)$ is a manifold $\mathcal{M}$ whose points, together with a binary operator "o" defined on them, form a group.

$$X^* : \mathcal{M} \rightarrow T^*\mathcal{M}$$

Definition A.2.7. Lie Derivative The derivative of a function f along the vector field X is a map defined on $\mathcal{M}$ and it is defined by:

$$L_X f : \mathcal{M} \rightarrow \mathbb{R}; \quad L_X f(x) = \frac{\partial^T f}{\partial x} X(x) \quad x \in \mathcal{M} \tag{A.1}$$

Appendix B

An Introduction to Bond-graphs

In this appendix the main rules and elements of the bond graphs will be presented. The **Bond graph** is a language that allows to represent a physical system in a graphical way. The main difference from the well known *block diagrams* representation is that bond graph are built in such a way that the first principle of the thermodynamics, the *conservation of the energy*, is assured.
Indeed, the vertices, or elements, of a bond-graph represent the physical concepts of *energy storage*, *dissipation* or *transformation*, while the edges, here called *power bonds*, represent the *energy flow*, that by definition is the *power*, between two elements of the graph. This energetic approach is also physical convenient: the dynamics of a system are due to an exchange of energy among the different parts of the system. This holds for each physical domain, from the electromagnetic to the thermic. It follows that the bond-graph language easily allows the representation of a network of systems belonging to different domains.
Another difference between bond graphs and other graphical representation, such as block diagrams, is the absence of *causality*. In block diagrams the exchange of information between two blocks A and B is represented with an arrow. So an arrow representing a variable x going from A to B means that A produce x and give it to B that use it. This change of information is *causal*: x is an *effect* of A and a *cause* for B. Often this causality is physically artificial and not justified. Thinking about an electrical resistor, there is no reason why the current must be an effect of a voltage or vice versa. Bond graph allow us to not consider the causality during the modelling phase and to introduce it only when simulation is needed.

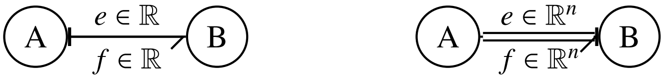


Figure B.1. A general bond-graph: single-bond (left) and multi-bond (right)

B.1 The bonds

Edges in the bond graphs are called *power bonds*, or, shortly, *bonds*, and represent the power exchanged between two elements. Bonds are represented as half arrows, and then the direction of the arrows represent the direction of the power passing through them.

In each physical domain there are two variables that multiplied give a result dimensionally equal to power, as velocities v and forces F in the mechanical translational domain or currents i and voltages v in the electrical domain. These couples of variables are called *power conjugate variables*, and for each of them a variable is called *flow* and the other is called *effort*. In the generic case, a flow is indicated as f and an effort as e. A list of flows and efforts for each domain is shown in Table B.1.

Physical Domain	**Flow**	**Effort**
Electromagnetic	current i	voltage v
Mechanical translational	lin. velocity v	force F
Mechanical rotational	ang. velocity ω	torque τ
Hydraulic	flow rate Q	pressure p
Thermic	entropy flow $\dot{E}$	temperature T

Table B.1. Flows and efforts in different domains

It follows that for each bond we can explicit its effort and flow. The notation impose that the effort value must be indicated above an horizontal and on the left of a vertical one and the dual flow under an horizontal bond and on the right side of a vertical one. Also, these variables can be single or multi-dimensional. Then a single-bond or a multi-bond is used to explicit it, as shown in figure B.1.

As said before, if needed the causality can be expressed. This is done using the *causal stroke* that specify the direction of the relative effort. As a consequence, the dual flow has the opposite direction. The reason is that if an element could impose

both effort and flow, it could set the power flow $P = e^T f$ independently of the rest of the system and, therefore, it could extract infinite energy from the other system and this in physically inconsistent.
An example of the introduction of the causality can be seen always in Figure B.1. In the single-bond graph the causal stroke is on the left of the bond. This means that the effort e goes from B to A, and so the flow goes from A to B. Conversely, in the multi-bond graph the causal stroke is on the right of the bond. This means that the effort e goes from A to B, and so the flow goes from B to A.

B.2 Energy Storage elements

A *storage element* represent a physical storage of energy in a model. Real examples of such elements are springs or inductors. A storage element in his *integral form* is characterized by:

- An input u
- An output y
- A physical state x
- An energy function $E(x)$ of the state x

The term ”integral form” means that the state of the element results from the integration of the input ($x(t) = x(t_0) + \int_{t_0}^{t} f(s)ds$). The state space equations of these kind of system are:

$$\dot{x}(t) = u(t) \tag{B.1}$$

$$y(t) = \gamma(x(t)) = \frac{\partial E}{\partial x}(t) \tag{B.2}$$

In the physical modelling,inputs are either efforts or flows. Then, if u is an efforts f must be a flow and vice versa. It follows that the power resulting from the dual product of u and f is equal to the power supplied to the storage element and the bond connected to this element has as effort and flow the input and output of the equations B.1 and B.2. Furthermore, due to the nature of the storage element, the half arrow of the power bond conneceted to it, indicating the direction of the positive power, must be ALWAYS directed towards the element.

The inputs of the storage element can be either an effort and a flow. Then two types of storage elements in their integral form can be defined, depending on which kind of input is considered. These two elements are called **C** and **I**.

B.2.1 C element

A **C** element is a storage element that in its integral form as a flow as the input u and as a consequence has the dual effort as an output y. Then, the causal stroke is not on the side of the element. The representation of a **C** element in its integral form is shown on the left side of Figure B.2.

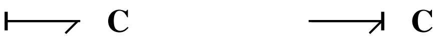


Figure B.2. C element in its integral (left) and differential (right) forms

Since the state x is the integral of a flow it is called *generalized displacement* and the energy, function of a generalized displacement, is called *generalized potential energy*. Inverting inputs and outputs, so to have an effort as input and a flow as output, the result is the *differential form*, since there is a differentiation in the equations of the model, as it will be shown in few moments.
In this form the "flow" of the variables is inverted and then the inverse of $\gamma(\cdot)$ and a derivative action is needed. In the same way as γ is a gradient of the energy function, γ^{-1} can be calculated as the gradient of a new function which is called *co-energy*, since it is not more function of the state (like an energy function is for a **C** element) but a function of an effort:

$$E^{*}(e) \Longrightarrow \gamma^{-1}(e) = \frac{\partial E^{*}(e)}{\partial e}$$

The causal representation of the differential form of a **C** element is shown on the right side of Figure B.2 and its state space equations are:

$$x(t) = \gamma^{-1}(e) = \frac{\partial E^{*}(e)}{\partial e} \tag{B.3}$$

$$y(t) = \frac{dx}{dt}(t) \tag{B.4}$$

The multidimensional case of **C** is $\mathbb{C}$.

B.2.2 I element

A **I** element is a storage element that in its integral form as an effort as the input u and as a consequence has the dual flow as an output y.Then, the causal stroke is on the side of the element. The representation of a **I** element in its integral form is shown on the left side of Figure B.3.

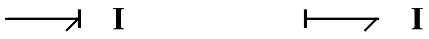


Figure B.3. I element in its integral (left) and differential (right) forms

Since the state x is the integral of an effort it is called *generalized momenta* and the energy, function of a generalized displacement, is called *generalized kinetic energy*. Inverting inputs and outputs, so to have an effort as input and a flow as output, the result is the *differential form*, since there is a differentiation in the equations of the model.
In this form the "flow" of the variables is inverted and then the inverse of $\gamma(\cdot)$ and a derivative action is needed. In the same way as γ is a gradient of the energy function, γ^{-1} can be calculated as the gradient of a new function which is called *co-energy*, since it is not more function of the state (like an energy function is for a **I** element) but a function of a flow:

$$E^*(f) \Longrightarrow \gamma^{-1}(f) = \frac{\partial E^*(f)}{\partial f}$$

The causal representation of the differential form of a **I** element is shown on the right side of Figure B.3 and its state space equations are:

$$x(t) = \gamma^{-1}(f) = \frac{\partial E^*(f)}{\partial f} \tag{B.5}$$

$$y(t) = \frac{dx}{dt}(t) \tag{B.6}$$

The multidimensional version of **I** is denoted as $\mathbb{I}$.

B.3 Energy Dissipation element

Excluding thermic domain from the analysis, it is possible to consider elements which dissipate energy, like electrical resistors or mechanical dampers. An ideal dissipator has no-state and its characteristic equation is a static relation between an effort and a flow:

$$e = Z(f) \qquad \text{(impedance form)} \tag{B.7}$$

$$f = Y(e) \qquad \text{(admittance form)} \tag{B.8}$$

for which, the following must hole:

$$Z(f)f \leq 0 \quad \text{or} \quad eY(e) \leq 0 \tag{B.9}$$

This last condition ensure that the energy can flow just toward the element but cannot be generated by the element because otherwise we would have for example that if $Z(f)f > 0$ would result in:

$$P_{out} = -P_{in} = -Z(f)f > 0 \tag{B.10}$$

The bond graph representation of a dissipation element is shown in Figure B.4

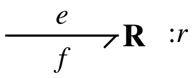


Figure B.4. Bond graph representation of a dissipation element

B.4 Energy Transformation elements

The element analyzed since now are characterized only by one port power. There are also elements with two power ports, corresponding to the two power bonds connected to the element, like electrical transformers and mechanical reducers. This kind of element are ideal and *power continuous* which means that in each instant of time the power flowing into the element from one of the two ports (input bond) is

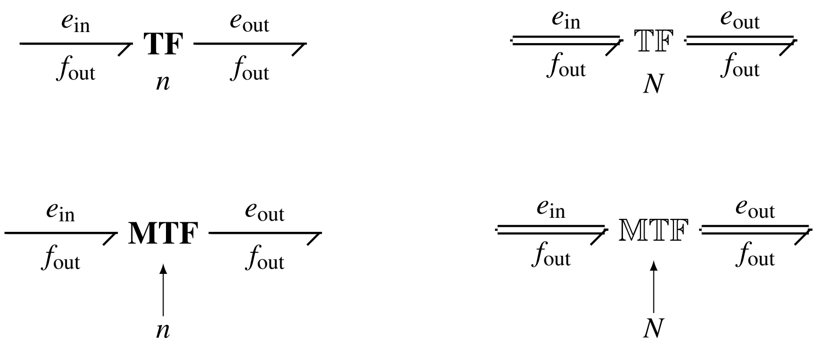


Figure B.5. Bond-graph representation of transformers

identical to the one flowing out from the other port (output bond). As a consequence, the element cannot store energy. Then, the following equation must holds:

$$P_{\text{in}} = e_{\text{in}}^{\text{T}} f_{\text{in}} = e_{\text{out}}^{\text{T}} f_{\text{out}} = P_{\text{out}} \tag{B.11}$$

Furthermore, these elements describe a linear relation between one of the external variable on one port to one of the external variables on the other port. Depending which variables this relation takes into account the element can be a *transformers* or a *gyrator*.

B.4.1 Transformers

A *transformer* describe a linear relation between flows and, due to the power continuity, a linear relation also between efforts of the two ports. Its bondgraph symbol is **TF** and its characteristic equation is:

$$f_{\text{out}} = n f_{\text{in}} \tag{B.12}$$

where n is the linear constant characterizing the transformer. The power continuity impose that

$$P_{\text{out}} = e_{\text{out}} f_{\text{out}} = e_{\text{out}} n f_{\text{in}} = e_{\text{in}} f_{\text{in}} = P_{\text{out}} \tag{B.13}$$

and so

$$e_{\text{in}} = n e_{\text{out}} \qquad \text{or} \qquad e_{\text{out}} = \frac{1}{n} e_{\text{in}} \tag{B.14}$$

This equations are used, for example, to describe the model of a gear-box. As well-known, its reduction ratio can be changed. This means that n is no more constant. In bond-graph representation this is achieved adding a signal port, not a power port, as input for the transformer. This transformers are called *modulated transformers* and their bond-graph symbol is **MTF**. The general representation of single-bond transformers **TF** and modulated transformers **MTF** is shown on the left side of Figure B.5.

B.4.1.1 Transformers in multi-dimensional domain

If the bonds a multi-dimensional (for example n), then the constant of the transformer $N \in \mathbb{R}^n \times \mathbb{R}^n$ and the equations that characterize the transformers are:

$$f_{\text{out}} = N f_{\text{in}} \qquad e_{\text{in}} = N^{\text{T}} e_{\text{out}} \tag{B.15}$$

The concepts applied to the single-dimensional case for a modulated transformer can be applied also to the multi-dimensional case. A multi-dimensional transformer is indicated as $\mathbb{TF}$ and the modulated case as $\mathbb{MTF}$. Representation of these elements are on the right side of Figure B.5.

B.4.2 Gyrators

A *gyrator*, differently to a transformer, relates the output effort with the input flow . Its bond-graph symbol is **GY** and its characteristic equation is:

$$e_{\text{out}} = n f_{\text{in}} \tag{B.16}$$

where n is the linear constant characterizing the gyrator. The power continuity impose that

$$P_{\text{out}} = e_{\text{out}} f_{\text{out}} = n f_{\text{in}} f_{\text{out}} = e_{\text{in}} f_{\text{in}} = P_{\text{out}} \tag{B.17}$$

and so

$$e_{\text{in}} = n f_{\text{out}} \qquad \text{or} \qquad f_{\text{out}} = \frac{1}{n} e_{\text{in}} \tag{B.18}$$

A perfect example of a gyrator is a DC motor, in which the electrical power flows in and the mechanical power flows out, coupling the resulting torque τ with the current i by the motor constant K.

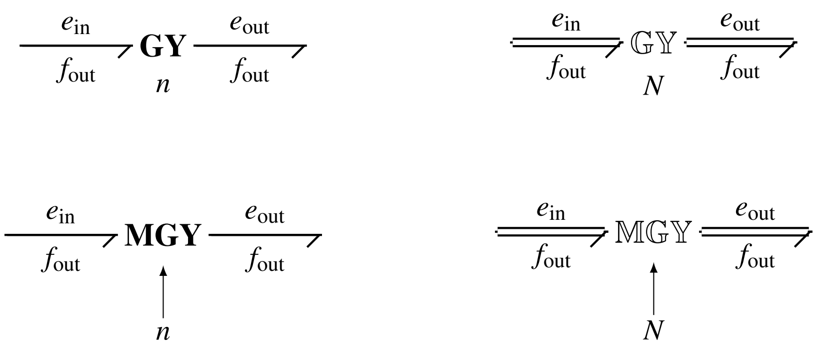


Figure B.6. Bond-graph representation of gyrators

As for the transformers, the characteristic ratio of a gyrator can be a variable. Again, this is represented adding a signal port as input for the gyrator and it is called *modulated transformers*, indicated as **MGY**. The general representation of single-bond transformers **GY** and modulated transformers **MGY** is shown on the left side of Figure B.6.

B.4.2.1 Gyrators in multi-dimensional domain

If the bonds a multi-dimensional (for example n), then the constant of the transformer $N \in \mathbb{R}^n \times \mathbb{R}^n$ and the equations that characterize the gyrator are:

$$e_{\text{out}} = N f_{\text{in}} \qquad\qquad e_{\text{in}} = N^{\mathrm{T}} f_{\text{out}} \tag{B.19}$$

Again, multi-bonds gyrator cab be *modulated.* A multi-dimensional gyratir is indicated as $\mathbb{GY}$ and the modulated case as $\mathbb{MGY}$. Representation of these elements are on the right side of Figure B.6.

B.5 Energy Source elements

Energy source elements are sources of either a flow or an effort and are called respectively *flow source* and *effort source* and indicated as $\mathbf{S}_f$ and $\mathbf{S}_e$, or as $\mathbb{S}_f$ and $\mathbb{S}_e$ in the multi-dimensional case.
An effort source can supply an effort independently of the dual flow, as a flow source can supply a flow independelty of the dual effort.
Being sources, the power bond direction goes always out, because the positive flow

of energy is the power $P_{\text{source}} = e^T f$. Bond-graph representation of the sources are shown in Figure B.7

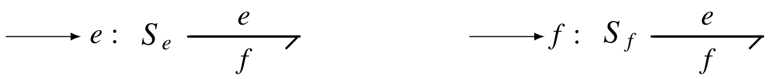


Figure B.7. Bond-graph representation of sources

B.6 Junction elements

Junction elements are the elements that allow the interconnection between all the types of elements explained above. How elements are interconnected specify how the energy flows in the modeled system.
The specification of the interconnections can be done using a generalization of Kirchhoff's laws. Furthermore, junctions can have any number of bonds attached to them and, as for the other elements, are power continuous which means that the total power flowing in must be equal to the total one flowing out. According to the generalization of Kirchhoff's laws the types of junctions are two, respectively called **0**-junction and **1**-junction.

B.6.1 0-junction

The **0**-junction is characterized by the fact that all bonds connected to it are constrained to have the same effort value at all times. For this reason, this junction is also called *effort junction*. This propriety implies two things: the first is that, causally speaking, only one of the bonds connected to the junction will set the effort value and all other bonds "use" it, while the second is that, due to the power continuity, the sum of the flows of the all entering bonds must be equal to the sum of the flows of all the exiting bonds. According to the Figure B.8 the characterizing equations of a **0**-junction are:

$$e_{i1} = \ldots = e_{im} = e_{o1} = \ldots = e_{on} \tag{B.20}$$

$$\sum_{k=1}^{m} f_{ik} = \sum_{k=1}^{n} f_{ok} \tag{B.21}$$

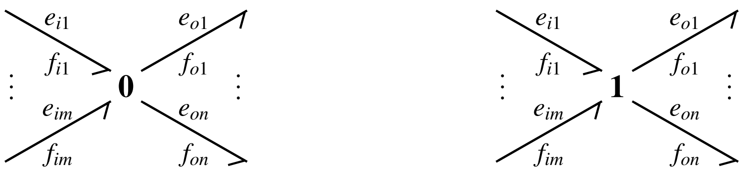


Figure B.8. Bond-graph representation of **0**-junction and **1**-junction

B.6.2 1-junction

The **1**-junction is characterized by the fact that all bonds connected to it are constrained to have the same flow value at all times. For this reason, this junction is also called *flow junction*. Than, as for the **0**-junction, only one of the bonds connected to the junction will set the flow value and all other bonds ”use” it, while the second is that, due to the power continuity, the sum of the efforts of the all entering bonds must be equal to the sum of the efforts of all the exiting bonds. According to the Figure B.8 the characterizing equations of a **1**-junction are:

$$f_{i1} = \ldots = f_{im} = f_{o1} = \ldots = f_{on} \tag{B.22}$$

$$\sum_{k=1}^{m} e_{ik} = \sum_{k=1}^{n} e_{ok} \tag{B.23}$$

Bibliography

[1] *20-Sim*. http://www.20sim.com.

[2] Massimo Cefalo. *Notes on the Kuka LWR4p dynamic model*. http://www.coppeliarobotics.com/contributions/LBR4p_dynamic_model.pdf.

[3] Namik Ciblak and Harvey Lipkin. “Design and Analysis of Remote Center of Compliance Structures”. In: *Journal of Robotic System* (2003).

[4] Douwe Dresscher. “Robust autonomy for the youBot”. Master thesis. University of Twente, Mar. 2010.

[5] V. Duindam et al. *Modeling and Control of Complex Physical Systems - The Port-Hamiltonian Approach*. Springer, 2009. ISBN: 978-3-642-03195-3.

[6] *FRI - Fast Research Interface*. http://cs.stanford.edu/people/tkr/fri/html/.

[7] Claudio Melchiorri. *Foundation of Industrial Robotics*. University Lecture. 2011.

[8] *ROS - Robotic Operative System*. http://www.ros.org.

[9] *RViz*. http://wiki.ros.org/rviz.

[10] C. Secchi, S. Stramigioli, and C. Fantuzzi. *Control of Interactive Robotic Interface*. Springer, 2007. ISBN: 10 3-540-49712-9.

[11] Stefano Stramigioli. *Modeling and IPC Control of Interactive Mechanical Systems - A Coordinate-Free Approach*. Lecture notes in control and information process. Springer, 2001. ISBN: 1-85233-395-2.

[12] Stefano Stramigioli. *Modern Robotics*. University Lecture. 2014.

[13] *urdf*. http://wiki.ros.org/urdf.

[14] Martijn Visser, Stefano Stramigioli, and Cock Heemskerk. “Cayley-Hamilton for robotics”. In: *IEEE* (2006).