8

Cartesian Control for Robot Manipulators

Pablo Sánchez-Sánchez and Fernando Reyes-Cortés
Benemérita Universidad Autónoma de Puebla (BUAP)
Facultad de Ciencias de la Electrónica
México

1. Introduction

A robot is a reprogrammable multi-functional manipulator designed to move materials, parts, tools, or specialized devices through variable programmed motions, all this for a best performance in a variety of tasks. A useful robot is the one which is able to control its movements and the forces it applies to its environment. Typically, robot manipulators are studied in consideration of their displacements on joint space, in other words, robot's displacements inside of its workspace usually are considered as joint displacements, for this reason the robot is analyzed in a joint space reference. These considerations generate an important and complex theory of control in which many physical characteristics appear, this kind of control is known as *joint control*.

The joint control theory expresses the relations of position, velocity and acceleration of the robot in its native language, in other words, describes its movements using the torque and angles necessary to complete the task; in majority of cases this language is difficult to understand by the end user who interprets space movements in cartesian space easily. The singularities in the boundary workspace are those which occur when the manipulator is completely streched-out or folded back on itself such as the end-effector is near or at the boundary workspace. It's necessary to understand that singularity is a mathematical problem that undefined the system, that is, indicates the absence of velocity control which specifies that the end-effector never get the desired position at some specific point in the workspace, this doesn't mean the robot cannot reach the desired position structurally, whenever this position is defined inside the workspace. This problem was solved by S. Arimoto and M. Takegaki in 1981 when they proposed a new control scheme based on the Jacobian Transposed matrix; eliminating the possibility of singularities and giving origin to the *cartesian control*.

The joint control is used for determining the main characteristics of the cartesian control based on the Jacobian Transposed matrix. It is necessary to keep in mind that to consider the robot's workspace like a joint space, has some problems with interpretation because the user needs having a joint dimensional knowledge, thus, when the user wants to move the robot's end-effector through a desired position he needs to understand the joint displacements the robot needs to do, to get the desired position. This interpretation problem is solved by using the cartesian space, that is, to interpret the robot's movements by using cartesian coordinates on reference of cartesian space; the advantage is for the final user who has the cartesian dimensional knowledge for understanding the robot's movements. Due this reason, learning the mathematical tools for analysis by the robot's movements on cartesian space is necessary, this allows us to propose control structures, to use the dynamic model and to understand the

physical phenomenons on robot manipulators on cartesian space. When we control the global motion or position of general manipulators, we are confronted with the nonlinear dynamics in a lot of degrees of freedom. In literature focused with the dynamic control of manipulators, the complexity of nonlinear dynamics is emphasized and some methods, compensating all nonlinear terms in dynamics in real time, are developed in order to reduce the complexity in system control. However, these methods require a large amount of complicated calculation so it is difficult to implement these methods with low level controllers such as microcomputers. In addition, the reliability of these methods may be lost when a small error in computation or a small change in system parameters occurs, occurs because they are not considered in the control. Most industrial robots, each joint of manipulator is independently controlled by a simple linear feedback. However, convergence for target position has not been enough investigated for general nonlinear mechanical systems.
This chapter is focused on the position control for robot manipulators by using control structures defined on the cartesian space because the robot move freely in its workspace, which is understood by the final user like cartesian space. Besides, the mathematical tools will be detailed for propose, analyze and evaluating control structures in cartesian space.

2. Preliminaries: forward kinematics and Jacobian matrix

A rigid multi-body system consists in a set of rigid objects, called links, joined together by joints. Simple kinds of joints include revolute (rotational) and prismatic (translational) joints. It is also possible to work with more general types of joints, and thereby simulate non-rigid objects. Well-known applications of rigid multi-bodies include robotic arms. A robot manipulator is modeled with a set of links connected by joints. There are a variety of possible joint types. Perhaps the most common type is a rotational joint with its configuration described by a single scalar angle value. The key point is: "the configuration of a joint is a continuous function of one or more real scalars; for rotational joints", the scalar is the angle of the joint. Complete configuration in robot manipulators is specified by vectors, for example the position is described as:

$$q = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix} \tag{1}$$

where $q \in \mathbb{R}^{n\times 1}$. We assume there are n joints and each q_n value is called a *joint position*.
The robot manipulator will be controlled by specifying target positions by the end-effectors. The desired positions are also given by a vector

$$q_d = \begin{bmatrix} q_{d_1} \\ q_{d_2} \\ \vdots \\ q_{d_n} \end{bmatrix} \tag{2}$$

where q_{d_i} is the desired position for the ith end-effector. We let $\tilde{q}_i = q_{d_i} - q_i$, the desired change in position of the ith end effector, also this vector is well-known as an error position. The end-effector positions (x, y, z) are functions of the joint angles q; this fact can be expressed as:

$$x_i = f_i(q) \qquad \text{for } i = 1, 2, \ldots, k \tag{3}$$

this equation is well-known as *forward kinematics*.

2.1 Case of study: Cartesian robot (forward kinematics

In order to understand application of cartesian control in robot manipulators a case of study will be used, which all the concepts were evaluated. In this section we will obtain the forward kinematics of a three degrees of freedom cartesian robot, Figure 1; and we will use this information in the following sections.

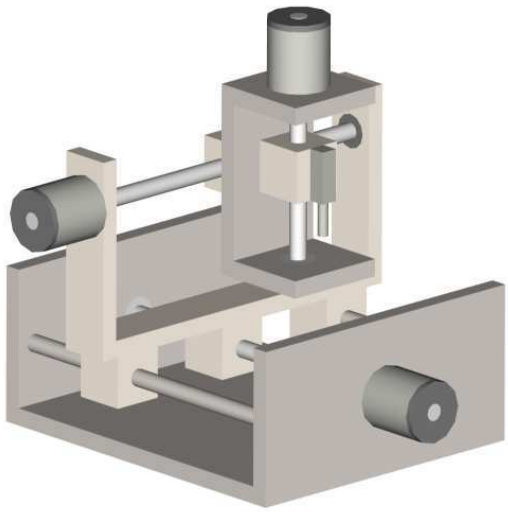


Fig. 1. Three degrees of freedom cartesian robot.

In order to obtain the forward kinematics of three degrees of freedom cartesian robot we need to draw a system diagram, Figura 2,

where q_1, q_2, q_3 are join displacements; and m_1, m_2, m_3 represent the masses of each link. As it is observed, translation is the unique movement that realizes this kind of robots, then the forward kinematics are defined as:

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} q_1 \\ 0 \\ 0 \end{bmatrix} \quad ; \quad \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ 0 \end{bmatrix} \quad ; \quad \begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}. \tag{4}$$

We can observe, that in the first vector is contemplated only by the first displacement of value q_1, in the second one considers the movement of translation in q_1 and q_2 respecting the axis x and y, and finally the complete displacement in third axis described in the last vector, being this representation the robot forward kinematics.

2.2 Jacobian matrix

The Jacobian matrix $J(q)$ is a multidimensional form of the derivative. This matrix is used to relate the joint velocity $\dot{q}$ with the cartesian velocity $\dot{x}$, based on this reason we are able to think about Jacobian matrix as mapping velocities in q to those in x:

$$\dot{x} = J(q)\,\dot{q}. \tag{5}$$

where $\dot{x}$ is the velocity on cartesian space; $\dot{q}$ is the velocity in joint space; and $J(q)$ is the Jacobian matrix of the system.

In many cases, we use modeling and simulation as a tool for analysis about the behavior of a given system. Even though at this stage, we have not formed the equations of motion for a robotic manipulator, by inspecting the kinematic models, we are able to revel many characteristics from the system. One of the most important quantities (for the purpose of analysis) in (5), is the Jacobian matrix $J(q)$. It reveals many properties of a system and can be used for the formulation of motion equations, analysis of special system configurations, static analysis, motion planning, etc. The robot manipulator's Jacobian matrix $J(q)$ is defined as follow:

$$J(q) = \frac{\partial f(q)}{\partial q} = \begin{bmatrix} \frac{\partial f_1(q)}{\partial q_1} & \frac{\partial f_1(q)}{\partial q_2} & \cdots & \frac{\partial f_1(q)}{\partial q_n} \\ \frac{\partial f_2(q)}{\partial q_1} & \frac{\partial f_2(q)}{\partial q_2} & \cdots & \frac{\partial f_2(q)}{\partial q_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m(q)}{\partial q_1} & \frac{\partial f_m(q)}{\partial q_2} & \cdots & \frac{\partial f_m(q)}{\partial q_n} \end{bmatrix} \tag{6}$$

where $f(q)$ is the relationship of forward kinematics, equation (3); n is the dimension of q; and m is the dimension of x. We are interested about finding what joint velocities $\dot{q}$ result in given (desired) v. Hence, we need to solve a system equations.

2.2.1 Case of study: Jacobian matrix of the cartesian robot

In order to obtain the Jacobian matrix of the three degrees of freedom cartesian robot it is necessary to use the forward kinematics which is defined as:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{7}$$

Now, doing the partial derivation of x in reference to q_1, q_2, q_3 we have:

$$\frac{\partial x}{\partial q_1} = \frac{\partial q_1}{\partial q_1} = \dot{q}_1$$
$$\frac{\partial x}{\partial q_2} = \frac{\partial q_1}{\partial q_2} = 0 \tag{8}$$
$$\frac{\partial x}{\partial q_3} = \frac{\partial q_1}{\partial q_3} = 0$$

The partial derivation of y in reference to q_1, q_2, q_3 are:

$$\frac{\partial y}{\partial q_1} = \frac{\partial q_2}{\partial q_1} = 0$$
$$\frac{\partial y}{\partial q_2} = \frac{\partial q_2}{\partial q_2} = \dot{q}_2 \tag{9}$$
$$\frac{\partial y}{\partial q_3} = \frac{\partial q_2}{\partial q_3} = 0$$

The partial derivation of z in reference to q_1, q_2, q_3, we have:

$$\frac{\partial z}{\partial q_1} = \frac{\partial q_3}{\partial q_1} = 0$$
$$\frac{\partial z}{\partial q_2} = \frac{\partial q_3}{\partial q_2} = 0 \tag{10}$$
$$\frac{\partial z}{\partial q_3} = \frac{\partial q_3}{\partial q_3} = \dot{q}_3$$

The system $\dot{x} = J(q)\dot{q}$ is described by following equation:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \frac{\partial x}{\partial q_2} & \frac{\partial x}{\partial q_3} \\ \frac{\partial y}{\partial q_1} & \frac{\partial y}{\partial q_2} & \frac{\partial y}{\partial q_3} \\ \frac{\partial z}{\partial q_1} & \frac{\partial z}{\partial q_2} & \frac{\partial z}{\partial q_3} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} \tag{11}$$

where the Jacobian matrix elements are defined using the equations (8), (9) and (10):

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{J(q)} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} \tag{12}$$

2.3 Jacobian transpose matrix

The transpose of a matrix $J(q)$ is another matrix $J(q)^T$ created by anyone of the following equivalent actions: write the $J(q)^T$ rows as the $J(q)^T$ columns; write the $J(q)^T$ columns as the $J(q)^T$ rows; and reflect $J(q)$ by its main diagonal (which starts from the top left) to obtain $J(q)^T$. Formally, the transpose of an $m \times n$ matrix $J(q)$ with elements $J(q)_{ij}$ is $n \times m$ matrix as follow

$$J_{ji}(q)^T = J_{ij}(q) \quad \text{for} \quad 1 \leq i \leq n, 1 \leq j \leq m. \tag{13}$$

The transposing of a scalar is the same scalar.

2.3.1 Case of study: Jacobian transpose matrix of the cartesian robot

In order to obtain the Jacobian transpose matrix $J(q)^T$ we apply (13) leaving of the equation (12). In particular case of cartesian robot the Jacobian matrix $J(q)$ is equal to the identity matrix I, thus its transposed matrix $J(q)^T$ is the same, thus we have:

$$J\,(q)^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{14}$$

2.4 Singularities

Singularities correspond certain configurations in robot manipulators which have to be avoided because they lead to an abrupt loss of manipulator rigidity. In the vicinity of these configurations, manipulator can become uncontrollable and the joint forces could increase considerably and may there would be risk to even damage the manipulator mechanisms. The singularities in a workspace can be identified mathematically when the determinant in the Jacobian matrix is zero:

$$\det J(q) = 0. \tag{15}$$

Mathematically this means that matrix $J(q)$ is degenerated and there is, in the inverse geometrical model, an infinity of solutions in the vicinity of these points.

2.5 Singular configurations

Due to the tuning of derivative and proportional matrices from the control algorithms of which objective is to maintain in every moment the error position nearest to zero, it exists the possibility that in certain values of the determinant in Jacobian matrix the system is singular undefined. It's denominated *singular configurations of a robot* those distributions in which that determinant of the Jacobian matrix is zero, equation (15). Because of this circumstance, in the singular configurations the inverse Jacobian matrix doesn't exist. For a undefine Jacobian matrix, an infinitesimal increment in the cartesian coordinates would suppose an infinite increment at joint coordinates, which is translated as movements from the articulations to inaccessible velocities on some part of its links for reaching the desired position for a constant velocity in the practice. Therefore, in the vicinity of the singular configurations lost some degrees in the robot's freedom, being impossible their end-effector moves in a certain cartesian address.

Different singular configurations on robot can be classified as:

- *Singularities in the limits in the robot's workspace*. These singularities are presented when the robot's boundary is in some point of the limit of interior or external workspace. In this situation it is obvious the robot won't be able to move in the addresses that were taken away from this workspace.
- *Singularities inside the robot's workspace*. They take place generally inside the work area and for the alignment of two or more axes in the robot's articulations.

2.5.1 Case of study: determinant of the Jacobian matrix of the cartesian robot

In order to determine if there are singularities in the system, it is necessary to obtain the determinant on the system $\det J(q)$, considering a general structure of the Jacobian matrix, thus we have:

$$\begin{aligned}
&\det J\left(q\right) = j_{11}\begin{bmatrix} j_{22} & j_{23} \\ j_{32} & j_{33} \end{bmatrix} - j_{12}\begin{bmatrix} j_{21} & j_{23} \\ j_{31} & j_{33} \end{bmatrix} + j_{13}\begin{bmatrix} j_{21} & j_{22} \\ j_{31} & j_{32} \end{bmatrix} \\
&\det J\left(q\right) = j_{11}\left(j_{22}j_{33} - j_{32}j_{23}\right) - j_{12}\left(j_{21}j_{33} - j_{31}j_{23}\right) + j_{13}\left(j_{21}j_{32} - j_{31}j_{22}\right) \\
&\det J\left(q\right) = 1
\end{aligned} \tag{16}$$

As it is observed, the determinant in the Jacobian matrix is not undefined in any point which indicates the workspace for the cartesian robot is complete.

2.5.2 Workspace

The workspace is the area where the robot can move freely with no damage. This area is determined by the robot's physical and mechanical capacities. The workspace is defined without considering the robot's end-effector, in the Figure 3 the workspace of a robot of three degrees of freedom is described.

2.6 Inverse Jacobian matrix

In mathematics, and especially in linear algebra, a matrix squared A with an order $n \times n$ it is said is reversible, nonsingular, non-degenerate or regular if exists another squared matrix with order $n \times n$ called *inverse matrix* A^{-1} and represented matrix like

$$AA^{-1} = A^{-1}A = I \tag{17}$$

I is the identity matrix with order $n \times n$ and the used product is the usual product of matrices. The mathematical definition in the inverse matrix is defined as follow:

$$J\left(q\right)^{-1} = \frac{C^T}{\det J\left(q\right)} \tag{18}$$

where C is the co-factors matrix.

2.6.1 Case of study: co-factors matrix in the cartesian robot

In order to obtain the co-factor matrix it is necessary to apply the following procedure: Considering the matrix A defined like:

$$A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \tag{19}$$

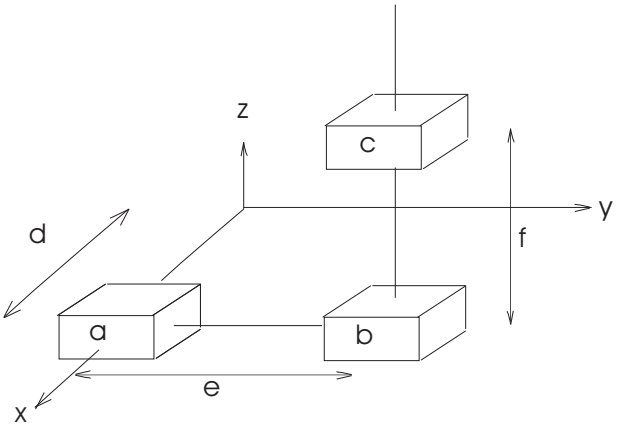


Fig. 2. Diagram of three degrees of freedom cartesian robot.

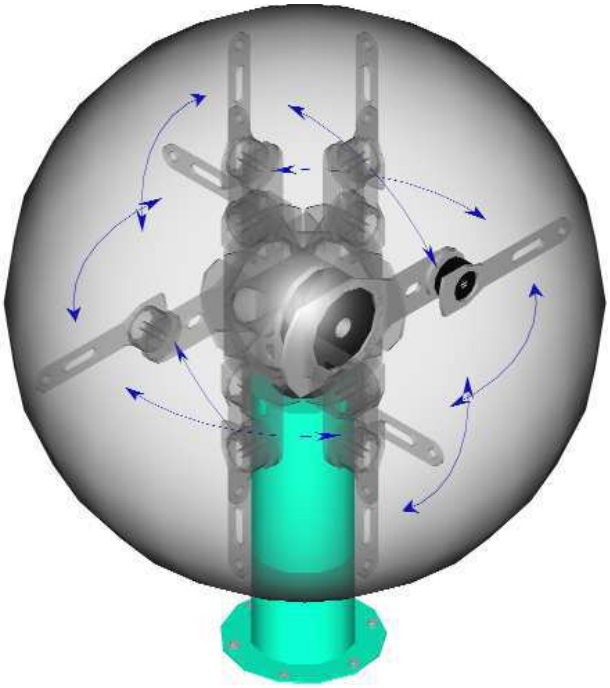


Fig. 3. Robot manipulator's workspace.

we obtain the following co-factors matrix:

$$C = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \tag{20}$$

where each component is defined as:

$$\begin{aligned} c_{11} &= +(ei - hf) \\ c_{12} &= -(di - gf) \\ c_{13} &= +(dh - ge) \\ c_{21} &= -(bi - hc) \\ c_{22} &= +(ai - gc) \\ c_{23} &= -(ah - gb) \\ c_{31} &= +(bf - ec) \\ c_{32} &= -(af - dc) \\ c_{33} &= +(ae - db) \end{aligned} \tag{21}$$

Considering the Jacobian matrix (12) we can obtain the following co-factors matrix:

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{22}$$

where the components of the matrix are defined as:

$$\begin{aligned} c_{11} &= +(ei - hf) &&= +(1-0) = 1 \\ c_{12} &= -(di - gf) &&= -(0-0) = 0 \\ c_{13} &= +(dh - ge) &&= +(0-0) = 0 \\ c_{21} &= -(bi - hc) &&= -(0-0) = 0 \\ c_{22} &= +(ai - gc) &&= +(1-0) = 1 \\ c_{23} &= -(ah - gb) &&= -(0-0) = 0 \\ c_{31} &= +(bf - ec) &&= +(0-0) = 0 \\ c_{32} &= -(af - dc) &&= -(0-0) = 0 \\ c_{33} &= +(ae - db) &&= +(1-0) = 1 \end{aligned} \tag{23}$$

2.6.2 Case of study: inverse Jacobian matrix of the cartesian robot

In order to obtain the inverse Jacobian matrix $J(q)^{-1}$ according the definition on (18), it is necessary the transposing co-factor matrix C^T,

$$C^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{24}$$

and the determinant of the Jacobian matrix (16), we obtain:

$$J(q)^{-1} = \frac{C^T}{\det J(q)} = \frac{1}{1}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
$$J(q)^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{25}$$

As it is observed, for the specific case of the three degrees of freedom cartesian robot the inverse matrix does exist.

3. Dynamic model

The dynamic model is the mathematical representation of a system which describes its behavior in the internal and external stimulus presented in the system. For cartesian control design purposes, and for designing better controllers, it is necessary to reveal the dynamic behavior of the robot via a mathematical model obtained from some basic physical laws. We use Lagrangian dynamics to obtain the describing mathematical equations. We begin our development with the general Lagrange equation about motion. Considering Lagrange's equation for a conservative system as given by:

$$\frac{d}{dt}\left[\frac{\partial \mathcal{L}(q,\dot{q})}{\partial \dot{q}}\right] - \frac{\partial \mathcal{L}(q,\dot{q})}{\partial q} = \tau - f(\tau,\dot{q}) \tag{26}$$

where $q, \dot{q} \in \mathbb{R}^{n\times 1}$ are position and velocity in a joint space, respectively; $\tau \in \mathbb{R}^{n\times 1}$ is a vector of an applied torque; $f(\tau,\dot{q}) \in \mathbb{R}^{n\times 1}$ is the friction vector; and the Lagrangian $\mathcal{L}(q,\dot{q})$ is the difference between kinetic $\mathcal{K}(q,\dot{q})$ and potential $\mathcal{U}(q)$ energies:

$$\mathcal{L}(q,\dot{q}) = \mathcal{K}(q,\dot{q}) - \mathcal{U}(q). \tag{27}$$

The application of the Lagrange's equation results in the mathematical equation which describes the system behavior at any stimulus, *dynamic model equation*. Then it can be shown the robot dynamics are given by:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) + f(\tau,\dot{q}) = \tau \tag{28}$$

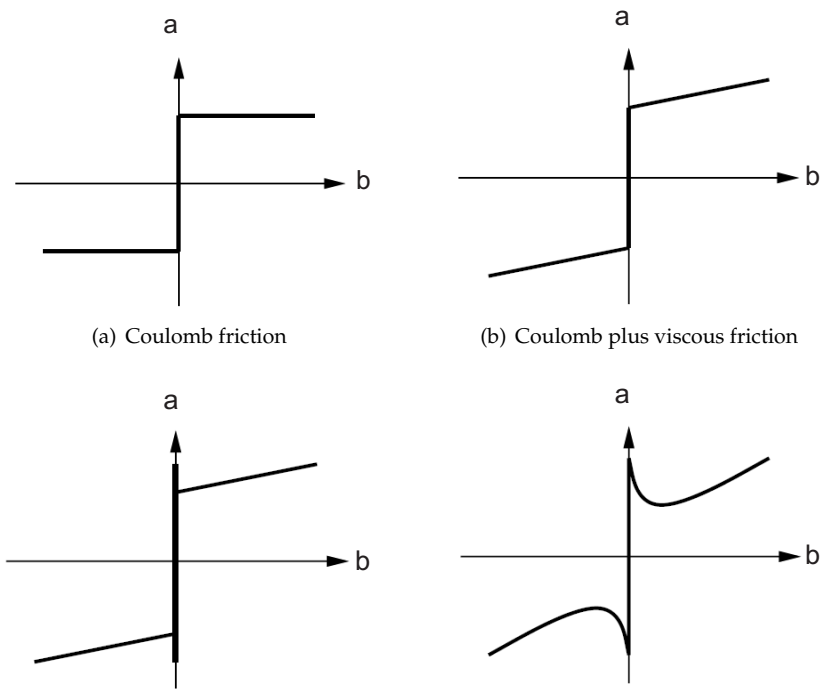
where $q, \dot{q}, \ddot{q}$ are the position, velocity and acceleration in joint space, respectively; $M(q) \in \mathbb{R}^{n\times 1}$ is symmetric, positive-definite inertial matrix; $C(q,\dot{q}) \in \mathbb{R}^{n\times n}$ is a matrix containing the Coriolis and centripetal torques effects; $g(q) \in \mathbb{R}^{n\times 1}$ is a vector of gravity torque obtained as gradient result on the potential energy,

$$g(q) = \frac{\partial \mathcal{U}(q)}{\partial q}, \tag{29}$$

and $f(\tau,\dot{q}) \in \mathbb{R}^{n\times 1}$ are the vector of friction torques. The friction torque is decentralize in the sense that $f(\tau,\dot{q})$ depends only on τ and $\dot{q}$

$$f(\tau,\dot{q}) = \begin{bmatrix} f_1(\tau_1,\dot{q}_1) \\ f_2(\tau_2,\dot{q}_2) \\ \vdots \\ f_n(\tau_n,\dot{q}_n) \end{bmatrix}. \tag{30}$$

Friction is the tangential reaction force between two surfaces in contact. Physically these reaction forces are the result of many different mechanisms, which depend on geometry and topology contact, properties of bulk and surface materials on the bodies, displacement and relative velocity on the bodies and presence of lubrication.
It is well known that exist two friction models: the *static and dynamic*. The static models of friction consist on different components, each take care about certain friction force issues. The main idea is: friction opposes motion and its magnitude is independent on velocity and contact area. The friction torques are assumed to be a dissipated energy at all nonzero velocities, therefore, their entries are bounded within the first and third quadrants. The friction force is given by a static function possibly except for a zero velocity. Figure 4(a) shows Coulomb friction; Figure 4(b) Coulomb plus viscous friction; Stiction plus Coulomb and viscous friction is shown in Figure 4(c); and Figure 4(d) shows how the friction force may decrease continuously from the static friction level.



(a) Coulomb friction (b) Coulomb plus viscous friction

(c) Stiction, Coulomb and viscous friction (d) Friction may decrease continuously

Fig. 4. Examples of static friction models.

This feature allows considering the common Coulomb and viscous friction models. At zero velocities, only static friction is satisfying presented

$$f_i(\tau_i, 0) = \tau_i - g_i(q) \tag{31}$$

for $-f_i \leq \tau_i - g_i(q) \leq f_i$ with f_i being the limit on the static friction torques for joint i.
Lately there has been a significant interest in dynamic friction models. This has been driven by intellectual curiosity, demands for precision servos and advances in hardware that make it possible for implementing friction compensators. The *Dahl model* was developed with the purpose of simulating control systems with friction. Dahl's starting point had several experiments on friction in servo systems with ball bearings. One of his findings was that bearing friction behave was very similar on solid friction. These experiments indicate that there are metal contacts between the surfaces. Dahl developed a simple comparatively model and was used extensively to simulate systems with ball bearing friction.
The starting point for Dahl's model is the stress-strain curve in classical solid mechanics, Figure 5. When the subject is under stress the friction force increases gradually until a rupture occurs. Dahl modeled the stress-strain curve by a differential equation. x will be the displacement, F the friction force, and F_c the Coulomb friction force. Then Dahl's model has this form:

$$\frac{dF}{dx} = \sigma \left(1 - \frac{F}{F_c}\mathrm{sgn}(v)\right)^{\alpha} \tag{32}$$

where σ is the stiffness coefficient; and α is a parameter which determines the shape of the stress-strain curve. The value $\alpha = 1$ is most commonly used. Higher values will give a stress strain curve with a sharper bend. The friction force $|F|$ will never be larger than F_c if its initial value is such that $|F(0)| < F_c$.

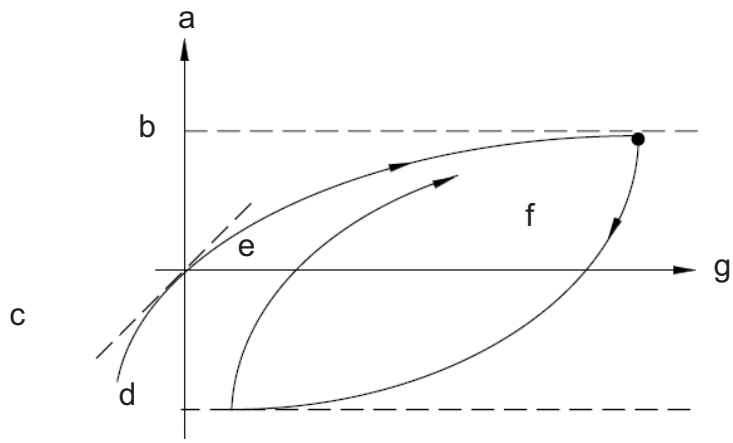


Fig. 5. Friction force as a function of displacement for Dahl's model.

With an absence of friction and other disturbances, the dynamic model (28) about n-links rigid robot manipulator can be written as:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau. \tag{33}$$

It is assumed that the robot links are joined together with revolute joints. Although the equation (28) is complex, for this reason we use the equation (33) for the analysis to facilitate control system design. It is necessary to indicate that the Euler-Lagrange's methodology is not the

only procedure to obtain the robot's dynamic model since this issue has been object of many study and researching. Researchers have developed alternative formulations based on the Newtonian and Lagrangian mechanics with one objective: obtaining a more efficient model.

3.1 Properties

It is essential to analyze the properties on the model to be able to apply them in the obtaining in the model on cartesian space.

3.1.1 Inertial matrix properties

The inertia matrix $M(q)$ has an important characteristic like its intimate relation with kinetic energy $\mathcal{K}(q,\dot{q})$,

$$\mathcal{K}(q,\dot{q}) = \frac{1}{2}\dot{q}^T M(q)\dot{q}. \tag{34}$$

The inertial matrix $M(q)$ is a positive definite matrix $M(q) > 0$; so is a symmetric matrix, $M(q) > 0 \Rightarrow \exists\, M(q)^{-1} > 0$. Another vital property of $M(q)$ is that it could be bounded above and below. So,

$$\mu_1(q)\, I \leq M(q) \leq \mu_2(q)\, I \tag{35}$$

where I is the identity matrix, $\mu_1(q) \neq 0$ and $\mu_2(q)$ are scalar constants for a revolute arm and scalar function of q for an arm generally containing prismatic joints. It is easy to realize then that $M^{-1}(q)$ is also bounded since

$$0 \leq \frac{1}{\mu_2(q)} I \leq M^{-1}(q) \leq \frac{1}{\mu_1(q)} I. \tag{36}$$

In the case of robots provided solely of rotational joints, a constant $\beta > 0$ exists like:

$$\lambda_{\max}\{M(q)\} \leq \beta \quad \forall q \in \mathbb{R}^{n\times 1} \tag{37}$$

where β is calculated as:

$$\beta \geq n \left[\max_{i,j,q} \left| M_{ij}(q) \right| \right] \tag{38}$$

where $M_{ij}(q)$ are the ijth element of the matrix $M(q)$.

3.1.2 Coriolis and centripetal terms properties

The matrix of Coriolis and centripetal force $C(q,\dot{q})$ is $n \times n$ matrix, of which elements are functions of q and $\dot{q}$. Matrix $C(q,\dot{q})$ can't be unique, but the vector $C(q,\dot{q})\dot{q}$ can. If the matrix $C(q,\dot{q})$ is evaluated considering the joint velocity $\dot{q}$ in zero, the matrix is zero for all vector q

$$C(q,0) = 0 \quad \forall q. \tag{39}$$

For all vector $q, x, y, z \in \mathbb{R}^{n\times 1}$ and the scale α we have:

$$\begin{aligned} C(q,x)\,y &= C(q,y)\,x \\ C(q,z+\alpha x)\,y &= C(q,z)\,y + \alpha C(q,x)\,y \end{aligned} \tag{40}$$

Vector $C(q,x)y$ can be expressed on the form:

$$C(q,x)\,y = \begin{bmatrix} x^T C_1(q)\,y \\ x^T C_2(q)\,y \\ \vdots \\ x^T C_n(q)\,y \end{bmatrix} \tag{41}$$

where $C_k(q)$ are symmetrical matrices of dimensions $n \times n$ for all $k = 1, 2, \ldots, n$; In fact the ijth element $C_{k_{ij}}(q)$ of matrix $C_k(q)$ corresponds to the symbol of Chistoffel:

$$C_{ijk}(q) = \frac{1}{2}\left[\frac{\partial M_{kj}(q)}{\partial q_i} + \frac{\partial M_{ki}(q)}{\partial q_j} - \frac{\partial M_{ij}(q)}{\partial q_k}\right]. \tag{42}$$

In the case of robots provided solely of rotational joints, a constant $k_{C_1} > 0$ exists like:

$$\|C(q,x)\,y\| \leq k_{C_1}\|x\|\,\|y\| \quad \text{for all } q, x, y \in \mathbb{R}^{n\times 1} \tag{43}$$

In the case of robots provided solely of rotational joints, a constants $k_{C_1} > 0$ and $k_{C_2} > 0$ exist like:

$$\|C(x,z)\,w - C(yu)\,w\| \leq k_{C_1}\|z-u\|\,\|w\| + k_{C_2}\|x-y\|\,\|w\|\,\|z\| \quad \forall u, x, y, w \in \mathbb{R}^{n\times 1} \tag{44}$$

Matrix $C(q,\dot{q})$ is related with the inertial matrix $M(q)$ by the expression:

$$x^T\left[\frac{1}{2}\dot{M}(q) - C(q,\dot{q})\right]x = 0 \quad \forall q, \dot{q}, x \in \mathbb{R}^{n\times 1} \tag{45}$$

In analogous form, the matrix $\dot{M}(q) - 2C(q,\dot{q})$ is skew-symmetric, and it also is certain that

$$\dot{M}(q) = C(q,\dot{q}) + C(q,\dot{q})^T. \tag{46}$$

3.1.3 Gravity terms properties

The gravity vector is present in robots which have not been designed mechanically with compensation of gravity, so, without counterbalances; or for robots assigned to move outside the horizontal plane. The gravity terms only depends on joint positions q; and the gravity terms can be related with the joint velocity $\dot{q}$ this means:

$$\int_0^T g(q)^T\,\dot{q}\,dt = \mathcal{U}(q(T)) - \mathcal{U}(q(0)) \quad \text{for all } T \in \mathbb{R}_+. \tag{47}$$

In the case of robots provided solely of rotational joints, a constant $k_{\mathcal{U}}$ exists like:

$$\int_0^T g(q)^T\,\dot{q}\,dt + \mathcal{U}(q(0)) \geq k_{\mathcal{U}} \quad \text{for all } T \in \mathbb{R}_+ \text{ and where } k_{\mathcal{U}} = \min_q\{\mathcal{U}(q)\} \tag{48}$$

In the case of robots provided solely of rotational joints the gravity vector $g(q)$ is Lipschitz, this means that a constant $k_g > 0$ exists like:

$$\|g(x) - g(y)\| \leq k_g\|x-y\| \quad \text{for all } x, y \in \mathbb{R}^{n\times 1} \tag{49}$$

A simple form to calculate k_g is:

$$k_g \geq n \left[\max_{i,j,q} \left| \frac{\partial g_i(q)}{\partial q_j} \right| \right] \tag{50}$$

In addition k_g satisfies:

$$k_g \geq \left\| \frac{\partial g(q)}{\partial q} \right\| \geq \lambda \max \left\{ \frac{\partial g(q)}{\partial q} \right\} \tag{51}$$

The gravity term $g(q)$ is bounded only if q is bounded:

$$\|g(q)\| \leq g_b \tag{52}$$

where g_b is a scalar constant for revolute arms and a scalar function of q for arms containing revolute joints.

3.2 Case of study: Dynamic model of cartesian robot

In this section we will obtain the dynamic model on three degrees of freedom cartesian robot, and we will use this information in the following sections. The case of study is represented in Figure 1. In order to obtain the dynamic model we need to consider its forward kinematics (4). However, as the movement of the cartesian robot only is about transferring, the rotation energy is zero, therefore, the equation in the kinetic energy is reduced to:

$$\mathcal{K}(q, \dot{q}) = \frac{mv^2}{2} = \frac{q^T M(q) q}{2}. \tag{53}$$

Considering velocity, it is defined as:

$$v = \frac{d}{dt} \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \tag{54}$$

and in (53) is represented in the squared way, it is necessary its vectorial representation,

$$v^2 = \|v\|^2 = v^T v = [v_1 \quad v_2 \quad v_3] \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}. \tag{55}$$

Deriving (4) and solving (55) we have:

$$\begin{aligned} v_1^2 &= \dot{q}_1^2 \\ v_2^2 &= \dot{q}_1^2 + \dot{q}_2^2 \\ v_3^2 &= \dot{q}_1^2 + \dot{q}_2^2 + \dot{q}_3^2. \end{aligned} \tag{56}$$

Replacing the values on v_1^2, v_2^2 and v_3^2 in (53) we obtain the kinetic energy of each link,

$$\begin{aligned}\mathcal{K}_1(q,\dot{q}) &= \frac{m_1\dot{q}_1^2}{2}\\ \mathcal{K}_2(q,\dot{q}) &= \frac{m_2(\dot{q}_1^2+\dot{q}_2^2)}{2}\\ \mathcal{K}_3(q,\dot{q}) &= \frac{m_3(\dot{q}_1^2+\dot{q}_2^2+\dot{q}_3^2)}{2}.\end{aligned} \tag{57}$$

Adding the kinetic energy of each link,

$$\mathcal{K}(q,\dot{q}) = \frac{m_1\dot{q}_1^2}{2} + \frac{m_2(\dot{q}_1^2+\dot{q}_2^2)}{2} + \frac{m_3(\dot{q}_1^2+\dot{q}_2^2+\dot{q}_3^2)}{2}, \tag{58}$$

we can expand the equation by the following form:

$$\mathcal{K}(q,\dot{q}) = \frac{m_1\dot{q}_1^2}{2} + \frac{m_2\dot{q}_1^2}{2} + \frac{m_2\dot{q}_2^2}{2} + \frac{m_3\dot{q}_1^2}{2} + \frac{m_3\dot{q}_2^2}{2} + \frac{m_3\dot{q}_3^2}{2}, \tag{59}$$

obtaining the total kinetic energy on the robot when grouping terms:

$$\mathcal{K}(q,\dot{q}) = \frac{(m_1+m_2+m_3)}{2}\dot{q}_1^2 + \frac{(m_2+m_3)}{2}\dot{q}_2^2 + \frac{m_3}{2}\dot{q}_3^2. \tag{60}$$

The potential energy $\mathcal{U}(q)$ is obtained considering in this case $h = q_3$ and $m = (m_1+m_2+m_3)$:

$$\mathcal{U}(q) = (m_1+m_2+m_3)gq_3. \tag{61}$$

After calculating the potential and kinetic energy on the robot we calculated the Lagrangian using (27):

$$\mathcal{L}(q,\dot{q}) = \frac{(m)\dot{q}_1^2}{2} + \frac{(m_2+m_3)\dot{q}_2^2}{2} + \frac{m_3\dot{q}_3^2}{2} - (m_1+m_2+m_3)gq_3. \tag{62}$$

When we used the obtained representation of the Lagrangian (27), we solve part by part the Euler-Lagrange equation for a conservativo system (26), we begin to solve the partial derived one from the Lagrangian with respect to the joint velocity $\dot{q}$:

$$\frac{\partial\mathcal{L}(q,\dot{q})}{\partial\dot{q}} = \begin{bmatrix} m_1 & 0 & 0\\ 0 & (m_2+m_3) & 0\\ 0 & 0 & (m_1+m_2+m_3)\end{bmatrix}\begin{bmatrix}\dot{q}_1\\ \dot{q}_2\\ \dot{q}_3\end{bmatrix}; \tag{63}$$

We continued with the derived in (63) with respect time:

$$\frac{d}{dt}\left[\frac{\partial\mathcal{L}(q,\dot{q})}{\partial\dot{q}}\right] = \begin{bmatrix} m_1 & 0 & 0\\ 0 & (m_2+m_3) & 0\\ 0 & 0 & (m_1+m_2+m_3)\end{bmatrix}\begin{bmatrix}\ddot{q}_1\\ \ddot{q}_2\\ \ddot{q}_3\end{bmatrix}, \tag{64}$$

and finally the independent term is solved:

$$\frac{\partial\mathcal{L}(q,\dot{q})}{\partial q} = \begin{bmatrix} 0\\ 0\\ (m_1+m_2+m_3)g\end{bmatrix}. \tag{65}$$

Thus, the dynamic model of the cartesian robot is:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 + m_3 & 0 \\ 0 & 0 & m_1 + m_2 + m_3 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ m_1 + m_2 + m_3 \end{bmatrix} g \tag{66}$$

being τ_1, τ_2 and τ_3 the applied torques. As we can observe, the dynamic model represented in (66) it is not under a friction influence.
Considering the following physical parameters:

Description	Notation	Value	Units
Link mass 1	m_1	16.180	kg
Link mass 2	m_2	14.562	kg
Link mass 3	m_3	12.944	kg
Gravity acceleration	g	9.8100	$\frac{m}{s^2}$

Table 1. Physical parameters on the three degrees of freedom cartesian robot

we can describe the dynamic model of the robot of three degrees of freedom by following:

$$\begin{aligned} M(q) &= \begin{bmatrix} 16.180 & 0 & 0 \\ 0 & 30.742 & 0 \\ 0 & 0 & 43.686 \end{bmatrix} \\ g(q) &= \begin{bmatrix} 0 \\ 0 \\ 43.686 \end{bmatrix} \end{aligned} \tag{67}$$

As it is observed in a cartesian robot the presence of the Coriolis and centripetal forces matrix $C(q, \dot{q})$ does not exist.

4. Hamilton's equations

Elegant and powerful methods have also been devised for solving dynamic problems with constraints. One of the best known is called Lagrange's equations, equation (26), where the Lagrangian $\mathcal{L}(q, \dot{q})$ is defined in (27). There is even a more powerful method called *Hamilton's equations*. It begins by defining a generalized momentum ρ, which is related to the Lagrangian $\mathcal{L}(q, \dot{q})$ and the generalized velocity $\dot{q}$ by:

$$\rho = \frac{\partial \mathcal{L}(q, \dot{q})}{\partial \dot{q}} \tag{68}$$

A new function, the Hamiltonian $\mathcal{H}(q, \rho)$, is defined by the addition of kinetic and potential energy:

$$\mathcal{H}(q, \rho) = \mathcal{K}(q, \dot{q}) + \mathcal{U}(q). \tag{69}$$

From this point it is not difficult to derive

$$\dot{q} = \frac{\partial \mathcal{H}(q, \rho)}{\partial \rho} \tag{70}$$

and

$$\dot{\rho} = \tau - \frac{\partial \mathcal{H}(q,\rho)}{\partial q} \tag{71}$$

These are called *Hamilton's equations*. There are two of them for each generalized coordinates. They may be used in place of Lagrange's equations, with the advantage that only the first derivatives not the second ones are involved.

Proof. In order to verify the obtaining of Hamilton's equations, the procedure begins by solving the first element on the equation:

$$\underbrace{\frac{d}{dt}\left[\frac{\partial \mathcal{L}(q,\dot{q})}{\partial \dot{q}}\right]}_{\text{first element}} - \frac{\partial \mathcal{L}(q,\dot{q})}{\partial q} = \tau. \tag{72}$$

In order to solve this part in the equation, we consider the Lagrangian $\mathcal{L}(q,\dot{q})$ as the difference between the kinetic $\mathcal{K}(q,\dot{q})$ and potential $\mathcal{U}(q)$ energy, equation (27); and we substitute it in the equation (72):

$$\frac{\partial \mathcal{L}(q,\dot{q})}{\partial \dot{q}} = \frac{\partial \mathcal{K}(q,\dot{q})}{\partial \dot{q}} - \frac{\partial \mathcal{U}(q)}{\partial \dot{q}}. \tag{73}$$

It is observed when we solve the partial derivation, the term which contains the potential energy $\mathcal{U}(q)$ is eliminated:

$$\frac{\partial \mathcal{L}(q,\dot{q})}{\partial \dot{q}} = \frac{\partial \mathcal{K}(q,\dot{q})}{\partial \dot{q}} - \cancel{\frac{\partial \mathcal{U}(q)}{\partial \dot{q}}} = \frac{\partial \mathcal{K}(q,\dot{q})}{\partial \dot{q}} \tag{74}$$

and considering that kinetic energy $\mathcal{K}(q,\dot{q})$ is defined as:

$$\mathcal{K}(q,\dot{q}) = \frac{\dot{q}^T M(q)\dot{q}}{2} = \frac{\rho^T M^{-1}(q)\rho}{2} \tag{75}$$

we can represent and solve the equation as follows:

$$\frac{\partial \mathcal{L}(q,\dot{q})}{\partial \dot{q}} = \frac{\partial \mathcal{K}(q,\dot{q})}{\partial \dot{q}} = \frac{\partial}{\partial \dot{q}}\left(\frac{\dot{q}^T M(q)\dot{q}}{2}\right) = M(q)\dot{q} = \rho \tag{76}$$

where ρ is the momentum. Finally, deriving (76) we have:

$$\frac{d}{dt}\left[\frac{\partial \mathcal{L}(q,\dot{q})}{\partial \dot{q}}\right] = M(q)\ddot{q} + \dot{M}(q)\dot{q} = \dot{\rho}. \tag{77}$$

Substituting the equation (77) in the equation (72) we have:

$$\dot{\rho} - \frac{\partial \mathcal{L}(q,\dot{q})}{\partial q} = \tau. \tag{78}$$

Now, in order to solve the second part of the equation (72) we need to consider the possible relationships between $\dot{\rho}$ and the energies as follows:

$$\dot{\rho} = \tau + \frac{\partial \mathcal{L}(q,\dot{q})}{\partial q} = \tau - \left(\frac{\partial \mathcal{U}(q)}{\partial q}\right) = \tau - \left(\frac{\partial \mathcal{H}(q,\rho)}{\partial q}\right) \tag{79}$$

this allows us to represent the equation (72) in the way:

$$\boxed{\dot{\rho} + \left(\frac{\partial \mathcal{H}(q,\rho)}{\partial q}\right) = \tau} \tag{80}$$

where $\mathcal{H}(q,\rho)$ is the Hamiltonian which represents the total energy of the system and it is defined as the sum of the kinetic $\mathcal{K}(q,\dot{q})$ and potential $\mathcal{U}(q)$ energy, equation (69). It is assumed that the potential energy $\mathcal{U}(q)$ of the system is twice differentiable respect q and any entry of the Hessian of $\mathcal{U}(q)$, it is bounded for all q. This assumption is done for general manipulators. Now, if we evaluate the Hamiltonian $\mathcal{H}(q,\rho)$ using the partial derivation in function the momentum ρ, we can observe the potential energy is eliminated:

$$\frac{\partial \mathcal{H}(q,\rho)}{\partial \rho} = \frac{\partial \mathcal{K}(q,\dot{q})}{\partial \rho} + \frac{\partial \mathcal{U}(q)}{\partial \rho} = \frac{\partial \mathcal{K}(q,\dot{q})}{\partial \rho} + \cancel{\frac{\partial \mathcal{U}(q)}{\partial \rho}} = \frac{\partial \mathcal{K}(q,\dot{q})}{\partial \rho} \tag{81}$$

and considering the form on the kinetic energy $\mathcal{K}(q,\dot{q})$, equation (75), we'll get:

$$\frac{\partial \mathcal{H}(q,\rho)}{\partial \rho} = \frac{\partial \mathcal{K}(q,\dot{q})}{\partial \rho} = \frac{\partial}{\partial \rho}\left(\frac{\rho^T M^{-1}(q)\,\rho}{2}\right) = M^{-1}(q)\,\rho = \dot{q} \tag{82}$$

Until now, we have obtained the equations (79) and (82), these equations are called as *Hamiltonian's equations of motion*:

$$\dot{\rho} = \tau - \left(\frac{\partial \mathcal{H}(q,\rho)}{\partial q}\right) \tag{83}$$

$$\dot{q} = \left(\frac{\partial \mathcal{H}(q,\rho)}{\partial \rho}\right) \tag{84}$$

□

4.1 Work and energy Principle

When forces act on a mechanism, *work* (in technical sense) is accomplish if the mechanism moves through a displacement. Work $\mathcal{W}$ is defined as force acting through a distance and it is a scalar with units of energy.

In order to understand and apply the concept about work $\mathcal{W}$, we must start from the physics point of view. In physics, many forces are the function of an appear position. Two examples are gravitational and the electrical forces. The one-dimensional movement equation which describes a body under the action of a force that depends on position q is:

$$F(q) = m\frac{d\dot{q}}{dt} \tag{85}$$

Integral on a force F which depends on position q, the realized force which represents the work $\mathcal{W}$ on a body while this one moves from starting point q_0 to the final q:

$$\mathcal{W} = \int_{q_0}^{q} F(q)\,dq. \tag{86}$$

The integral on the work $\mathcal{W}$ can be always found, in an explicit or numerical way, in both cases we can define a function $\mathcal{U}(q)$:

$$\mathcal{U}(q) = -\int_{q_0}^{q} F(q)\, dq \tag{87}$$

like:

$$F(q) = -\frac{d\mathcal{U}(q)}{dq}. \tag{88}$$

This allows us to express the work $\mathcal{W}$ meaning the difference between $\mathcal{U}(q)$ value in the external points:

$$\mathcal{W} = \int_{q_0}^{q} F(q)\, dq = -\mathcal{U}(q)\Big|_{q_0}^{q} = -\mathcal{U}(q) + \mathcal{U}(q_0) \tag{89}$$

Function $\mathcal{U}(q)$ is the potential energy which has the body when it is placed on point q. Up to this moment we have found that: work $\mathcal{W}$ carried out by force $F(q)$ is equal to the difference between initial and final value on potential energy $\mathcal{U}(q)$ of the body.
Starting off in (86), we can observe that the value of the force $F(q)$ can be replace by us on equation (85), as follows:

$$\mathcal{W} = \int_{q_0}^{q} F(q)\, dq = \int_{q_0}^{q} m\frac{d\dot{q}}{dt} dq \tag{90}$$

Considering velocity definition $\dot{q}$:

$$\dot{q} = \frac{dq}{dt} \tag{91}$$

We can realize a change on variable based on dq

$$dq = \dot{q}dt, \tag{92}$$

thus we have:

$$\mathcal{W} = \int_{q_0}^{q} F(q)\, dq = \int_{q_0}^{q} m\frac{d\dot{q}}{\cancel{dt}}\dot{q}\cancel{dt} = \int_{t_0}^{t} m\dot{q}d\dot{q} = m\int_{t_0}^{t} \dot{q}d\dot{q} \tag{93}$$

Solving the integral, we obtain:

$$\mathcal{W} = m\int_{t_0}^{t} \dot{q}d\dot{q} = \frac{1}{2}m\dot{q}^2\Big|_{t_0}^{t} = \frac{1}{2}m\dot{q}^2(t) - \frac{1}{2}m\dot{q}^2(t_0) \tag{94}$$

As it is observed in (94), when it is solved the integral, appears the kinetic energy of the body. It is well known the unit of kinetic energy $\mathcal{K}(q, \dot{q})$ in MKS system equal to work $\mathcal{W}$, that is the Joule. Until this moment we have found for any force $F(q)$

$$\mathcal{W} = \int_{q_0}^{q} F(q)\, dq = -\mathcal{U}(q) + \mathcal{U}(q_0) = \mathcal{K}(q, \dot{q}) - \mathcal{K}(q_0, \dot{q}_0) \tag{95}$$

and using (89) and (94) it is possible to express as:

$$\mathcal{W} = -\mathcal{U}(q) + \mathcal{U}(q_0) = \mathcal{K}(q, \dot{q}) - \mathcal{K}(q_0, \dot{q}_0), \tag{96}$$

and putting together all the equation elements we have:

$$\mathcal{W} = \mathcal{K}(q, \dot{q}) + \mathcal{U}(q) = \mathcal{U}(q_0) + \mathcal{K}(q_0, \dot{q}_0) \tag{97}$$

(97) expresses *the total conservation of energy principle* $\mathcal{E}(q, \dot{q})$.

$$\mathcal{W} = \underbrace{\mathcal{K}(q, \dot{q}) + \mathcal{U}(q)}_{\mathcal{E}(q,\dot{q})} = \mathcal{U}(q_0) + \mathcal{K}(q_0, \dot{q}_0) \tag{98}$$

This principle is applicable for any one-dimensional problem where the force just represents function of position q. Equation (98) also receives the name of *the work and energy principle*, and establishes the work $\mathcal{W}$ carried out by a force $F(q)$ on a body is equal to the change or variation of its kinetic energy $\mathcal{K}(q, \dot{q})$.
One must observe that the function on defined potential energy $\mathcal{U}(q)$ in (87) is indefinite by a constant value, constant integration. Nevertheless, this has no matter, since in any application it will only seem the difference of potential energies, equation (89). It is important to remember this, because it will allow us to choose arbitrarily the point where the body has potential energy zero $\mathcal{U}(q) = 0$. In addition, it will allow us, at any time, being able to do it, adding in all points the same constant amount to the potential energy $\mathcal{U}(q)$ of a body without affecting the results.

4.2 Principle of energy

The total energy of the system $\mathcal{E}(q, \dot{q})$ expressed in (98) can be considered like *the Hamiltonian* $\mathcal{H}(q, \rho)$,

$$\mathcal{W} = \mathcal{E}(q, \dot{q}) = \underbrace{\mathcal{K}(q, \dot{q}) + \mathcal{U}(q)}_{\mathcal{H}(q, \rho)} \tag{99}$$

and its derived can be considered like the *power* $\mathcal{P}$. Power $\mathcal{P}$ can be interpreted like the work velocity or the work carried out by a time unit; and it is defined as follow:

$$\mathcal{P} = \frac{d\mathcal{W}}{dt} = \frac{d\mathcal{H}(\dot{q}, \rho)}{dt} \tag{100}$$

From a mechanical point of view, the power (mechanical power) is the transmitted force by means on the associated mechanical element or by means on the contact forces. The simplest case is that a variable force acts in a free particle. According to the classic dynamics, this power is used by some variation from its kinetic energy $\mathcal{K}(q, \dot{q})$ or carried out by a time unit. Whereas in mechanical systems more complex like rotating elements on a constant axis, and where the moment of inertia I remains constant, the mechanical power can be related to the engine torque or torque applied τ, and the joint velocity $\dot{q}$, being the variation power of the angular kinetic energy by time unit; in case of expressed vectorial systems, thus we have:

$$\mathcal{P} = \frac{d\mathcal{W}}{dt} = \frac{d\mathcal{H}(\dot{q}, \rho)}{dt} = \tau^T \dot{q}. \tag{101}$$

where $\tau \in \mathbb{R}^{n \times 1}$ represents the vector of forces and torques at the end-effector; and $\dot{q} \in \mathbb{R}^{n \times 1}$ is a joint velocity.

Proof. We begin differentiating (100); we can observed that the following balance energy immediately appears:

$$\frac{d\mathcal{H}(q,\rho)}{dt} = \left(\frac{\partial\mathcal{H}(q,\rho)}{\partial q}\right)^T \dot{q} + \left(\frac{\partial\mathcal{H}(q,\rho)}{\partial \rho}\right)^T \dot{\rho} \tag{102}$$

Substituting the joint velocity $\dot{q}$ from equation (84) in (102) we have:

$$\frac{d\mathcal{H}(q,\rho)}{dt} = \left(\frac{\partial\mathcal{H}(q,\rho)}{\partial q}\right)^T \left(\frac{\partial\mathcal{H}(q,\rho)}{\partial \rho}\right) + \left(\frac{\partial\mathcal{H}(q,\rho)}{\partial \rho}\right)^T \dot{\rho}. \tag{103}$$

Applying the matrix property $x^T y = y^T x$ we get:

$$\frac{d\mathcal{H}(q,\rho)}{dt} = \left(\frac{\partial\mathcal{H}(q,\rho)}{\partial \rho}\right)^T \left(\frac{\partial\mathcal{H}(q,\rho)}{\partial q}\right) + \left(\frac{\partial\mathcal{H}(q,\rho)}{\partial \rho}\right)^T \dot{\rho}. \tag{104}$$

The factorization of the equation (104):

$$\frac{d\mathcal{H}(q,\rho)}{dt} = \left(\frac{\partial\mathcal{H}(q,\rho)}{\partial \rho}\right)^T \left[\left(\frac{\partial\mathcal{H}(q,\rho)}{\partial q}\right) + \dot{\rho}\right] \tag{105}$$

this allow us to substitute the equation (80) in the equation (105)

$$\frac{d\mathcal{H}(q,\rho)}{dt} = \left(\frac{\partial\mathcal{H}(q,\rho)}{\partial \rho}\right)^T \tau. \tag{106}$$

Applying the matrix property $x^T y = y^T x$ we get:

$$\frac{d\mathcal{H}(q,\rho)}{dt} = \tau^T \left(\frac{\partial\mathcal{H}(q,\rho)}{\partial \rho}\right) \tag{107}$$

and substituting the joint velocity $\dot{q}$ from equation (84) in (107) we obtain:

$$\frac{d\mathcal{H}(q,\rho)}{dt} = \tau^T \dot{q} \tag{108}$$

□

5. Cartesian space

The joint space is analyzed because it offers mathematical bases for the cartesian space. Cartesian space gives advantages of interpretation to the end-user, and for him is easier to locate the cartesian coordinates (x, y, z) which joint displacements $(q_1, q_2, \ldots, q_n)$; that is, for the final user it is intuitive to understand the space location of a body expressed in cartesian coordinates; so it is important to describe the characteristics and properties of the cartesian space. The analysis of the cartesian space leaving of the joint space begins by considering the inverse kinematics, which are one of the basic functions for control systems robot manipulators. Inverse kinematics is the process which determines the joint parameters of a based object on the cartesian position which is described as a function f on the joint variable q:

$$x = f(q). \tag{109}$$

In order to solve the inverse problem in (109) it is necessary to determine q using a partial derivation as follow:

$$\dot{x} = J(q)\dot{q}, \tag{110}$$

where $J(q)$ is the Jacobian matrix, $\dot{q}$ is the joint velocity; and $\dot{x}$ is the cartesian velocity. The equation (110) allows us to obtain the joint velocity representation as follow:

$$\dot{q} = J(q)^{-1}\dot{x} \tag{111}$$

After some operations, we can relate the joint space with cartesian space using some equations, Table 2.

Joint space	**Cartesian space**
$\dot{q} = J(q)^{-1}\dot{x}$	$\dot{x} = J(q)\dot{q}$
$\ddot{q} = J(q)^{-1}\ddot{x} - J(q)^{-1}\dot{J}(q)J(q)^{-1}\dot{x}$	$\ddot{x} = J(q)\ddot{q} + \dot{J}(q)\dot{q}$

Table 2. Equations which relate both workspaces.

Partial derivation on the inverse kinematic model establishes a relationship between the joint and cartesian velocity. The inverse Jacobian matrix obtained will be used for study on singular positions of the robot manipulator, for evaluation of its maneuverability and also for optimization of its architecture.
The forward kinematics model, equation (109), provides the relationships to determine cartesian and joint position on the end-effector given by the joint position. As it is observed in the equations where they relate the workspaces, Table 2, the Jacobian matrix $J(q)$ appears.

5.1 Jacobian transpose controller

In order to define the cartesian space for control proposes is required the dynamic model in joint space, equation (33); the equations which relate the work spaces, Table 1; and a new scheme of control known *Jacobian transpose controller*. In 1981 Suguru Arimoto and Morikazu Takegaki members of the mechanical engineering department in the Osaka University in Japan, they published in the *Journal of Dynamic Systems, Measurement and Control* Vol. 103 a new control scheme based on the Jacobian Transpose matrix; the conservation energy idea; the principle of virtual works and generalized force; and the static equilibrium. Jacobian transpose method removes the problematic from the Jacobian inversion mentioned above. The annoying inversion is replaced by a simple transposition. This control scheme was used for stability proof in the PD controller in global way, this was the first stability proof in the PD controller. This proposal changed the point of view from the control theory because it avoided singularities, doing as possible the robot manipulator all desired positions inside its workspace. It is well known, the applied torque and cartesian force satisfies:

$$\tau = J(q)^T\mathcal{F} \tag{112}$$

where $\tau \in \mathbb{R}^{n\times 1}$ is the vector of applied torques, $J(q) \in \mathbb{R}^{n\times n}$ is the Jacobian matrix and $\mathcal{F} \in \mathbb{R}^{n\times 1}$ is the vector from the applied force at the end-effector in cartesian space. The equation (112) is called *Jacobian transpose controller*. The external force $\mathcal{F}$ is applied to the end-effector on the articulated structure and results in internal forces and torques in joints.

Proof. The *principle of energy* allows us to make certain statements about the static case by allowing the amount of this displacement to go to an infinitesimal. From the physical point of view, it is well known that the work has units of energy, this must be the same measured in any set of generalized coordinates, this allows us to describe the power $\mathcal{P}$ as follow:

$$\frac{d\mathcal{W}}{dt} = \frac{d\mathcal{H}\left(\dot{q},\rho\right)}{dt} = \tau^T\dot{q} \tag{113}$$

Specifically, we can equate the work done in cartesian terms with the work done in joint space terms.
In the multidimensional case, work $\mathcal{W}$ is the dot product of a vector force or torque and the vector displacement. Thus we have:

$$\frac{d\mathcal{W}}{dt} = \mathcal{F}^T\dot{x}, \tag{114}$$

a necessary condition to satisfy the static equilibrium:

$$\mathcal{F}^T\dot{x} = \tau^T\dot{q} \tag{115}$$

where $\mathcal{F} \in \mathbb{R}^{n\times 1}$ represents the vector of forces and torques at the end-effector in cartesian coordinates; $\dot{x} \in \mathbb{R}^{n\times 1}$ is a cartesian velocity; $\tau \in \mathbb{R}^{n\times 1}$ is a vector of torque; and $\dot{q} \in \mathbb{R}^{n\times 1}$ is the joint velocity. Finally, let's $\dot{q}$ represent the corresponding joint velocity. These velocity are related through the Jacobian matrix $J(q)$ according to equation (110):

$$\mathcal{F}^T J(q)\dot{q} = \tau^T\dot{q} \tag{116}$$

The *virtual work* of the system is defined as:

$$\mathcal{F}^T J(q)\dot{q} - \tau^T\dot{q} = 0, \tag{117}$$

this is equal to zero if the manipulator is in equilibrium. Factorizing the equation (117) we have:

$$\left(\mathcal{F}^T J\left(q\right) - \tau^T\right)\dot{q} = 0. \tag{118}$$

If we analyzed the equation (118) we can determine that the system is equal to zero, this assumption let us make the following equality:

$$\mathcal{F}^T J\left(q\right) - \tau^T = 0. \tag{119}$$

Applying the property $x^T y = y^T x$

$$J\left(q\right)^T\mathcal{F} - \left(\tau^T\right)^T = 0 \tag{120}$$

and the property $(x^T)^T = x$ in the equation (120) we have:

$$J\left(q\right)^T\mathcal{F} - \tau = 0 \tag{121}$$

and obtaining the applied torque τ we have:

$$\tau = J\left(q\right)^T\mathcal{F} \tag{122}$$

where $\tau \in \mathbb{R}^{n\times 1}$ is the applied torque; $\mathcal{F} \in \mathbb{R}^{n\times 1}$ represent the vector of forces and torques at the end-effector in cartesian coordinates; and $J(q) \in \mathbb{R}^{n\times n}$ is the Jacobian matrix on the system. In other words the end-effector forces are related to joint torques by the Jacobian transpose matrix according to (122).

□

5.2 Dynamic model based-on the Jacobian transpose controller

In 1981 Suguru Arimoto and Morikazu Takegaki members of the mechanical engineering department in the Osaka University in Japan, they published in the Journal of Dynamic Systems, Measurement and Control Vol. 103 a new scheme control based on the Jacobian Transpose matrix (122); the energy conservation idea; the principle of virtual works and generalized force; and the static equilibrium. Jacobian transpose method removes the problematic of the Jacobian inversion and the singularity problem. Suguru Arimoto and Morikazu Takegaki substituted the Jacobian transpose controller, equation (122), in the dynamic model, equation (33),

$$M(q)\,\ddot{q} + C(q,\dot{q})\,\dot{q} + g(q) = J(q)^T\,\mathcal{F} \tag{123}$$

and using the equations described in Table 1, we obtain:

$$M(x)\ddot{x} + C(x,\dot{x})\dot{x} + g(x) = \tau_x, \tag{124}$$

where:

$$\begin{aligned} M(x) &= J(q)^{-T}M(q)J(q)^{-1} && (125)\\ C(x,\dot{x}) &= J(q)^{-T}C(q,\dot{q})J(q)^{-1} - J(q)^{-T}M(q)J(q)^{-1}\dot{J}(q)J(q)^{-1} && (126)\\ g(x) &= J(q)^{-T}g(q) && (127)\\ \tau_x &= \mathcal{F} && (128) \end{aligned}$$

we obtained a dynamic model representation on Jacobian transpose terms.

5.2.1 Properties

Although the equation of motion (124) is complex, it has several fundamental properties which can be exploited to facilitate a control system design. We use the following important properties. In order to establish the properties on the described dynamic model in cartesian space it is necessary to do the following assumptions:

Assumption 1. Jacobian matrix does exist, $J(q)\,\exists$; is continuously differentiable respecting each entry of q, $J(q) \in \mathcal{C}^k$; and it is considered that is of a full rank:

$$\text{rank}\,\{J(q)\} = n.$$

This assumption is required for technical reason in stability analysis inside the workspace in cartesian space.

Assumption 2. According to the assumption 1, Jacobian matrix has a full rank, this consideration indicates that its inverse representation does exist,

$$\text{If rank}\,\{J(q)\} = n \text{ then } J(q)^{-1}\,\exists.$$

This assumption indicates the existence of the Jacobian matrix and its inverse within the workspace Ω.

Assumption 3. According to the assumption 1, Jacobian matrix is continuously differentiable respecting each entry of q, this consideration indicates that its derivative representation does exist,

$$\text{If rank } \{J(q)\} = n \text{ then } \dot{J}(q) \, \exists.$$

This assumption indicates the existence on the derivative representation of the Jacobian matrix within the workspace Ω.

Assumption 4. If the Jacobiana matrix does exist, its transpose does exist,

$$\text{If } J(q) \, \exists \text{ then } J(q)^T \, \exists.$$

This assumption indicates the existence of the transpose representation of the Jacobian matrix within the workspace Ω.

Assumption 5. According to the assumption 1, 2 and 4, the Jacobiana transpose matrix does exist, and its inverse does exist,

$$\text{If } J(q)^T \, \exists \text{ then } J(q)^{-T} \, \exists.$$

This assumption indicates the existence of the inverse transpose representation of the Jacobian matrix within the workspace Ω.

Assumption 6. According to assumptions 1, 2 and 5; and considering the definition of an inertial matrix $M(x)$, equation (125), we can say inertial matrix $M(x)$ does exist,

$$\text{If } J(q)^{-1} \, \exists, J(q)^{-T} \, \exists \text{ and } M(q) \, \exists \text{ then } M(x) \, \exists.$$

This assumption indicates the existence on the inertial matrix $M(x)$ within the workspace Ω. Obviously, matrix $M(q)$ must exists.

Assumption 7. According to assumption 6 the inertial matrix $M(x)$ does exist, then according to assumption 1 its inverse does exist,

$$M(x) \, \exists \quad \text{and} \quad M(x)^{-1} \, \exists.$$

Assumption 8. Matrix $M(x)$ does exist, and it is symmetric,

$$M(x) \, \exists \quad \text{and} \quad M(x) = M(x)^T \, \exists.$$

Proof. In order to verify this assumption it is necessary to consider the definition of matrix $M(x)$, equation (125), and transposing the matrix,

$$M(x)^T = \left(J(q)^{-T} M(q) J(q)^{-1} \right)^T \tag{129}$$

Applying the formula $(xyz)^T = z^T y^T x^T$ we have:

$$
\begin{aligned}
M(x)^T &= \left(J(q)^{-1}\right)^T (M(q))^T \left(J(q)^{-T}\right)^T \\
M(x)^T &= J(q)^{-T} M(q)^T J(q)^{-1}
\end{aligned}
\tag{130}
$$

Matrix $M(q)$ is symmetrical[1], this allows us to represent (130) the following form:

$$
M(x)^T = J(q)^{-T} M(q) J(q)^{-1} \tag{131}
$$

We can conclude that the following equality is fulfilled:

$$
M(x) = M(x)^T \tag{132}
$$

□

Assumption 9. Considering that the matrix $J(q)$ does exist, assumption 1, and its inverse does exist $J(q)^{-1}$, assumption 2, when multiplying $J(q)J(q)^{-1}$ or $J(q)^{-1}J(q)$ we obtain the identity matrix I.

$$
\text{If } J(q)\ \exists \text{ and } J(q)^{-1}\ \exists \text{ then } J(q)\, J(q)^{-1} = I_{JJ^{-1}}
$$

$$
\text{If } J(q)^{-1}\ \exists \text{ and } J(q)\ \exists \text{ then } J(q)^{-1}\, J(q) = I_{J^{-1}J}
$$

We observed that obtained matrices I are equal,

$$
I_{JJ^{-1}} = I_{J^{-1}J} = I
$$

Assumption 10. Considering that the matrix $J(q)^T$ does exist, assumption 4, and its inverse does exist $J(q)^{-T}$, assumption 5, when multiplying $J(q)^T J(q)^{-T}$ or $J(q)^{-T} J(q)^T$ we obtain the identity matrix I.

$$
\text{If } J(q)^T\ \exists \text{ and } J(q)^{-T}\ \exists \text{ then } J(q)^T\, J(q)^{-T} = I_{J^T J^{-T}}
$$

$$
\text{If } J(q)^T\ \exists \text{ and } J(q)^{-T}\ \exists \text{ then } J(q)^{-T}\, J(q)^T = I_{J^{-T} J^T}
$$

We observed that obtained matrices I are equal,

$$
I_{J^T J^{-T}} = I_{J^{-T} J^T} = I
$$

5.2.1.1 Inertial matrix $M(x)$ properties

In accordance with assumption 6 the inertial matrix $M(x)$ exists, according to assumption 7 the inverse inertial matrix exists, and in reference about assumption 8 the inertial matrix $M(x)$ is symmetric. Another vital property of $M(x)$ is that it is bounded above and below. So,

$$
\mu_1(x) I \leq M(x) \leq \mu_2(x) I \tag{133}
$$

where I is the identity matrix, $\mu_1(x) \neq 0$ and $\mu_1(x)$ are constant scalars for a revolute arm and generally the function scalar of x for an arm containing prismatic joints.

[1] For more information, consult section 3.1.1.

5.2.1.2 Coriolis and centripetal terms $C(x,\dot{x})$ properties

The matrix $\dot{x}^T\left[\dot{M}(x)-2C(x,\dot{x})\right]\dot{x}\equiv 0$ is skew-symmetric, so,

$$\dot{M}(x)=C(x,\dot{x})+C(x,\dot{x})^T. \tag{134}$$

We need to keep in mind that the equality described in (134) can be written in the following form:

$$\dot{M}\left(x\right)-\left[C\left(x,\dot{x}\right)+C\left(x,\dot{x}\right)^T\right]=0 \tag{135}$$

Proof. Considering the definition on the inertia matrix $M(x)$, equation (125), and the Coriolis and centripetal terms $C(x,\dot{x})$, equation (126), both in cartesian space; we will verify the equation (134) is fulfilled. Therefore we initiated transposing the Coriolis matrix, thus we have:

$$C\left(x,\dot{x}\right)^T=J\left(q\right)^{-T}C\left(q,\dot{q}\right)^T J\left(q\right)^{-1}-J\left(q\right)^{-T}\dot{J}\left(q\right)^T J\left(q\right)^{-T}M\left(q\right)J\left(q\right)^{-1} \tag{136}$$

what it allows us to solve operation $C(x,\dot{x})+C(x,\dot{x})^T$:

$$\begin{aligned}C\left(x,\dot{x}\right)+C\left(x,\dot{x}\right)^T=\;& J\left(q\right)^{-T}C\left(q,\dot{q}\right)J\left(q\right)^{-1}-J\left(q\right)^{-T}M\left(q\right)J\left(q\right)^{-1}\dot{J}\left(q\right)J\left(q\right)^{-1}\\ &+J\left(q\right)^{-T}C\left(q,\dot{q}\right)^T J\left(q\right)^{-1}-J\left(q\right)^{-T}\dot{J}\left(q\right)^T J\left(q\right)^{-T}M\left(q\right)J\left(q\right)^{-1}\end{aligned} \tag{137}$$

As is observed, we can put together the following terms:

$$\begin{aligned}C\left(x,\dot{x}\right)+C\left(x,\dot{x}\right)^T=\;& J\left(q\right)^{-T}C\left(q,\dot{q}\right)J\left(q\right)^{-1}-J\left(q\right)^{-T}M\left(q\right)J\left(q\right)^{-1}\dot{J}\left(q\right)J\left(q\right)^{-1}\\ &+J\left(q\right)^{-T}C\left(q,\dot{q}\right)^T J\left(q\right)^{-1}-J\left(q\right)^{-T}\dot{J}\left(q\right)^T J\left(q\right)^{-T}M\left(q\right)J\left(q\right)^{-1}\end{aligned} \tag{138}$$

Thus we have:

$$\begin{aligned}C\left(x,\dot{x}\right)+C\left(x,\dot{x}\right)^T=\;& J\left(q\right)^{-T}\left[C\left(q,\dot{q}\right)+C\left(q,\dot{q}\right)^T\right]J\left(q\right)^{-1}-J\left(q\right)^{-T}M\left(q\right)J\left(q\right)^{-1}\dot{J}\left(q\right)J\left(q\right)^{-1}\\ &-J\left(q\right)^{-T}\dot{J}\left(q\right)^T J\left(q\right)^{-T}M\left(q\right)J\left(q\right)^{-1}\end{aligned} \tag{139}$$

Applying (46) we have:

$$\begin{aligned}C\left(x,\dot{x}\right)+C\left(x,\dot{x}\right)^T=\;& J\left(q\right)^{-T}\dot{M}\left(q\right)J\left(q\right)^{-1}-J\left(q\right)^{-T}M\left(q\right)J\left(q\right)^{-1}\dot{J}\left(q\right)J\left(q\right)^{-1}\\ &-J\left(q\right)^{-T}\dot{J}\left(q\right)^T J\left(q\right)^{-T}M\left(q\right)J\left(q\right)^{-1}\end{aligned} \tag{140}$$

Now, replacing (125) in (140),

$$\begin{aligned} C(x,\dot{x}) + C(x,\dot{x})^T = \quad & J(q)^{-T}\dot{M}(q)J(q)^{-1} - \underbrace{J(q)^{-T}M(q)J(q)^{-1}}_{M(x)}\dot{J}(q)J(q)^{-1} \\ & -J(q)^{-T}\dot{J}(q)^T\underbrace{J(q)^{-T}M(q)J(q)^{-1}}_{M(x)} \end{aligned} \tag{141}$$

thus we have:

$$\boxed{C(x,\dot{x}) + C(x,\dot{x})^T = J(q)^{-T}\dot{M}(q)J(q)^{-1} - M(x)\dot{J}(q)J(q)^{-1} - J(q)^{-T}\dot{J}(q)^T M(x)} \tag{142}$$

Equation (142) represents the first part on the proof.
The second step consists on deriving matrix $M(x)$ defined in (125), thus we have:

$$\dot{M}(x) = \dot{J}(q)^{-T}M(q)J(q)^{-1} + J(q)^{-T}\dot{M}(q)J(q)^{-1} + J(q)^{-T}M(q)\dot{J}(q)^{-1} \tag{143}$$

Using the equation (125), we can find $M(q)$ as follows:

$$\begin{aligned} M(x) &= J(q)^{-T}M(q)J(q)^{-1} \\ J(q)^T M(x) &= M(q)J(q)^{-1} \\ J(q)^T M(x)J(q) &= M(q) \end{aligned} \tag{144}$$

This allows us to replace $M(q)$ expressed in (144) in (143), as follows:

$$\begin{aligned} \dot{M}(x) = \quad & \dot{J}(q)^{-T}J(q)^T M(x)J(q)J(q)^{-1} + J(q)^{-T}\dot{M}(q)J(q)^{-1} \\ & +J(q)^{-T}J(q)^T M(x)J(q)\dot{J}(q)^{-1} \end{aligned} \tag{145}$$

Some terms can be eliminated applying the identity matrix property:

$$\begin{aligned} \dot{M}(x) = \quad & \dot{J}(q)^{-T}J(q)^T M(x)\underbrace{J(q)J(q)^{-1}}_{I} + J(q)^{-T}\dot{M}(q)J(q)^{-1} \\ & + \underbrace{J(q)^{-T}J(q)^T}_{I} M(x)J(q)\dot{J}(q)^{-1} \end{aligned} \tag{146}$$

thus we have:

$$\boxed{\dot{M}(x) = \dot{J}(q)^{-T}J(q)^T M(x) + J(q)^{-T}\dot{M}(q)J(q)^{-1} + M(x)J(q)\dot{J}(q)^{-1}} \tag{147}$$

Equation (147) represents the second part on the proof.

For the following step on the proof, we must consider next equation:

$$\dot{I}_{J^{-T}J^{T}} M(x) + M(x)\, \dot{I}_{JJ^{-1}} = 0 \tag{148}$$

where $\dot{I}_{J^{-T}J^{T}}$ and $\dot{I}_{JJ^{-1}}$ are derivative forms on following equations:

$$\begin{aligned} I_{J^{-T}J^{T}} &= J(q)^{T} J(q)^{-T} \\ I_{JJ^{-1}} &= J(q) J(q)^{-1} \end{aligned} \tag{149}$$

thus we have:

$$\dot{I}_{J^{-T}J^{T}} = \frac{d\left(J(q)^{-T} J(q)^{T}\right)}{dt} = \dot{J}(q)^{-T} J(q)^{T} + J(q)^{-T} \dot{J}(q)^{T} = 0 \tag{150}$$

$$\dot{I}_{JJ^{-1}} = \frac{d\left(J(q) J(q)^{-1}\right)}{dt} = \dot{J}(q) J(q)^{-1} + J(q) \dot{J}(q)^{-1} = 0 \tag{151}$$

In (150) and (151) we are applying assumption 9 and 10. It is well known that derivation of identity matrix is equal to zero, $\dot{I} = 0$. Now, replacing (150) and (151) in 148 we get:

$$\underbrace{\left[\dot{J}(q)^{-T} J(q)^{T} + J(q)^{-T} \dot{J}(q)^{T}\right]}_{\frac{d\left(J(q)^{-T} J(q)^{T}\right)}{dt}} M(x) + M(x) \underbrace{\left[J(q) \dot{J}(q)^{-1} + \dot{J}(q) J(q)^{-1}\right]}_{\frac{d\left(J(q) J(q)^{-1}\right)}{dt}} = 0 \tag{152}$$

Solving internal operations we have:

$$\dot{J}(q)^{-T} J(q)^{T} M(x) + J(q)^{-T} \dot{J}(q)^{T} M(x) + M(x) J(q) \dot{J}(q)^{-1} + M(x) \dot{J}(q) J(q)^{-1} = 0 \tag{153}$$

Adding a zero on form:

$$J(q)^{-T} \dot{M}(q) J(q)^{-1} - J(q)^{-T} \dot{M}(q) J(q)^{-1} = 0 \tag{154}$$

thus we have:

$$\begin{aligned} &\dot{J}(q)^{-T} J(q)^{T} M(x) + J(q)^{-T} \dot{J}(q)^{T} M(x) + J(q)^{-T} \dot{M}(q) J(q)^{-1} \\ &+ M(x) J(q) \dot{J}(q)^{-1} + M(x) \dot{J}(q) J(q)^{-1} - J(q)^{-T} \dot{M}(q) J(q)^{-1} = 0 \end{aligned} \tag{155}$$

As it is observed, the equality is conserved. Ordering equation (155) we get:

$$\begin{aligned} &\dot{J}(q)^{-T} J(q)^{T} M(x) + J(q)^{-T} \dot{M}(q) J(q)^{-1} + M(x) J(q) \dot{J}(q)^{-1} \\ &- J(q)^{-T} \dot{M}(q) J(q)^{-1} + J(q)^{-T} \dot{J}(q)^{T} M(x) + M(x) \dot{J}(q) J(q)^{-1} = 0 \end{aligned} \tag{156}$$

Replacing (142) and (147) in (156),

$$\underbrace{\dot{J}(q)^{-T} J(q)^{T} M(x) + J(q)^{-T} \dot{M}(q) J(q)^{-1} + M(x) J(q) \dot{J}(q)^{-1}}_{\dot{M}(x)}$$
$$- \underbrace{\left[J(q)^{-T} \dot{M}(q) J(q)^{-1} - J(q)^{-T} \dot{J}(q)^{T} M(x) - M(x) \dot{J}(q) J(q)^{-1} \right]}_{C(x,\dot{x})+C(x,\dot{x})^{T}} = 0 \tag{157}$$

thus we have:

$$\dot{M}(x) - \left[C(x,\dot{x}) + C(x,\dot{x})^{T} \right] = 0. \tag{158}$$

Ordering (158) we have:

$$\dot{M}(x) = C(x,\dot{x}) + C(x,\dot{x})^{T}. \tag{159}$$

□

5.2.1.3 Gravity terms properties

The generalized gravitational forces vector

$$g(x) = \frac{\partial \mathcal{U}(x)}{\partial x} \tag{160}$$

satisfies:

$$\left\| \frac{\partial g(x)}{\partial x} \right\| \leq k_g \tag{161}$$

for some $k_g \in \mathrm{R}_+$, where $\mathcal{U}(x)$ is the potential energy expressed in the cartesian space and is supposed to be bounded from below.

5.2.2 Case of study: Dynamic model based-on the $J(q)^T$ on cartesian robot

Along the chapter, we have evaluated a three degrees of freedom cartesian robot and we have obtained several equations which are required to obtain the dynamic model based on the Jacobian transpose controller, equation (124), these matrices are:

- The Jacobian matrix $J(q)$ defined in (12), this matrix fulfills assumption 1:

$$J(q) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- The transpose representation on the Jacobian matrix defined in (14), this matrix fulfills assumption 1 and 4:

$$J(q)^{T} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- The inverse representation on the Jacobian matrix is defined in (25), this matrix fulfills assumption 1 and 2:

$$J(q)^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- The inverse representation of the Jacobian transpose matrix, this matrix fulfills assumption 1 and 5:

$$J(q)^{-T} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- The derivative representation on the Jacobian matrix, this matrix fulfills assumptions 1 and 3:

$$\dot{J}(q) = \frac{d}{dt}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

To obtain the defined dynamic model in (124) last set of matrices is needed, thus we have, for the inertial matrix $M(x)$ defined in (125):

$$M(x) = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{J(q)^{-T}} \underbrace{\begin{bmatrix} 16.180 & 0 & 0 \\ 0 & 30.472 & 0 \\ 0 & 0 & 43.686 \end{bmatrix}}_{M(q)} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{J(q)^{-1}} \tag{162}$$

Solving (162) we obtain:

$$M(x) = \begin{bmatrix} 16.180 & 0 & 0 \\ 0 & 30.472 & 0 \\ 0 & 0 & 43.686 \end{bmatrix}. \tag{163}$$

For the Coriolis and centripetal matrix $C(x, \dot{x})$ defined in (126) thus we have:

$$C(x, \dot{x}) = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{J(q)^{-T}} \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{C(q,\dot{q})} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{J(q)^{-1}}$$

$$- \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{J(q)^{-T}} \underbrace{\begin{bmatrix} 16.180 & 0 & 0 \\ 0 & 30.472 & 0 \\ 0 & 0 & 43.686 \end{bmatrix}}_{M(q)} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{J(q)^{-1}} \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{\dot{J}(q)} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{J(q)^{-1}} \tag{164}$$

Solving (164) we obtain:

$$C(x,\dot{x}) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{165}$$

As it is observed by obtaining from matrix $C(x,\dot{x})$, in a cartesian robot rotation behavior does not exist, thus the matrix of Coriolis does not exist either.
For the gravity term $g(x)$ defined in (127) thus we have:

$$g(x) = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{J(q)^{-T}} \underbrace{\begin{bmatrix} 0 \\ 0 \\ 43.686 \end{bmatrix}}_{g(q)} \tag{166}$$

Solving (166) we obtain:

$$g(x) = \begin{bmatrix} 0 \\ 0 \\ 43.686 \end{bmatrix} \tag{167}$$

Now we have the dynamic model based on Jacobian transpose controller is defined as:

$$\underbrace{\begin{bmatrix} 16.180 & 0 & 0 \\ 0 & 30.472 & 0 \\ 0 & 0 & 43.686 \end{bmatrix}}_{M(x)} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 43.686 \end{bmatrix}}_{g(x)} = \underbrace{\begin{bmatrix} \mathcal{F}_1 \\ \mathcal{F}_2 \\ \mathcal{F}_3 \end{bmatrix}}_{\tau_x} \tag{168}$$

5.3 Cartesian controllers

In this section we present our main results concerning about stability analysis on the cartesian controllers. Now we are in position to formulate a cartesian control problem. Typically we propose controllers using the *energy shaping* on joint coordinates, now we use this methodology on cartesian space.

5.3.1 Energy shaping on cartesian space

The *energy shaping* is a technique which allows to design the control algorithms using kinetic and artificial potential energy which is shaping via a gradient for stabilization at the equilibrium point and damping injection to make this equilibrium attractive. The designed control algorithm is composed by the gradient on the artificial potential energy plus a velocity feedback. We use the following cartesian control scheme:

$$\tau_x = \nabla\mathcal{U}(k_p,\tilde{x}) - f_v(k_v,\dot{x}) + g(x) \tag{169}$$

where $\mathcal{U}(k_p,\tilde{x})$ is the *artificial potential energy* described by:

$$\mathcal{U}(k_p,\tilde{x}) = \frac{f(\tilde{x})^T k_p f(\tilde{x})}{2}, \tag{170}$$

and the term $f_v(k_v,\dot{x})$ is the derivative action. We use the following Lyapunov scheme:

$$V(\dot{x},\tilde{x}) = \frac{\dot{x}^T M(x)\dot{x}}{2} + \mathcal{U}(k_x,\tilde{x}). \tag{171}$$

where $M(x)$ is a local definite function. The energy shaping methodology consists about finding a $\mathcal{U}(k_x,\tilde{x})$ function to fulfill the next Lyapunov's conditions:

$$\begin{array}{ll} V(0,0) = 0 & \forall\, \dot{x},\tilde{x} = 0 \\ V(\dot{x},\tilde{x}) > 0 & \forall\, \dot{x},\tilde{x} \neq 0 \end{array} \tag{172}$$

and doing the derivation of the Lyapunov equation we get,

$$\dot{V}(\dot{x},\tilde{x}) = \dot{x}^T M(x)\ddot{x} + \frac{\dot{x}^T \dot{M}(x)\dot{x}}{2} - \frac{\partial \mathcal{U}(k_p,\tilde{x})^T}{\partial \tilde{x}}\dot{x}, \tag{173}$$

fulfill the condition:

$$\dot{V}(\dot{x},\tilde{x}) \leq 0, \tag{174}$$

verify asymptotical stability with LaSalle theorem:

$$\dot{V}(\dot{x},\tilde{x}) < 0. \tag{175}$$

In general terms, when we consider the dynamic model on cartesian space, equation (124), together with control law (169), then the closed-loop system is locally stable and the positioning aim:

$$\lim_{t\to\infty} x(t) = x_d \quad \wedge \quad \lim_{t\to\infty} \dot{x}(t) = 0 \tag{176}$$

is achieved.

Proof. The closed-loop system equation obtained by combining the robot dynamic model on cartesian space, equation (124), and the control scheme (169), can be written as:

$$\frac{d}{dt}\begin{bmatrix} \tilde{x} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} -\dot{x} \\ M(x)^{-1}\left[\tau_x - C(\dot{x},x)\,\dot{x}\right] \end{bmatrix} \tag{177}$$

which is an autonomous differential equation, and the origin of state space is its unique equilibrium point, we need to keep in mind that the inverse representation of the inertial matrix $M(x)$ exists only if only the Jacobian matrix fulfills assumption 1. Considering the autonomous system:

$$\dot{x} = f(x), \tag{178}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is locally Lipschitz map in $\mathbb{R}^n$. Let x_e be an equilibrium point for $f(x_e) = 0$. Let $V : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable, positive definite function such as $\dot{V}(x) \leq 0\ \forall\ x \in \mathbb{R}^n$. Let $\Omega = \{\dot{x} \in \mathbb{R}^n | \dot{V}(x) = 0\}$ and suppose that no solution could stay identically in Ω, other than the trivial solution, then the origin is locally stable. In our case $f(x)$ is given by the closed-loop system equation (178), where $x = [\tilde{x},\dot{x}]^T \in \mathbb{R}^{2n}$. The origin of the space state is its unique equilibrium point for (178). Let $V : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable, positive definite function such as $\dot{V}(\tilde{x},\dot{x}) \leq 0\ \forall\ \dot{x},\tilde{x} \in \mathbb{R}^n$. Let the region:

$$\Omega = \left\{ \begin{bmatrix} \tilde{x} \\ \dot{x} \end{bmatrix} \in \mathbb{R}^{2n} : \dot{V}(\tilde{x}, \dot{x}) = 0 \right\}$$
$$\Omega = \left\{ \tilde{x} \in \mathbb{R}^n, \dot{x} = 0 \in \mathbb{R}^n : \dot{V}(\tilde{x}, \dot{x}) = 0 \right\}, \tag{179}$$

since $\dot{V}(\tilde{x}, \dot{x}) \leq 0 \in \Omega, V(\tilde{x}(t), \dot{x}(t))$ is a decreasing function of t. $V(\tilde{x}, \dot{x})$ is continuous on the compact set Ω, so it is bounded from below Ω.
For example, it satisfies $0 \leq V(\tilde{x}(t), \dot{x}(t)) \leq V(\tilde{x}(0), \dot{x}(0))$. Therefore, $V(\tilde{x}(t), \dot{x}(t))$ has limit α as $t \to \infty$. Hence $\dot{V}(\tilde{x}(t), \dot{x}(t)) = 0$ and the unique invariant is $\tilde{x} = 0$ and $\dot{x} = 0$. Since the trivial solution is the closed-loop system unique solution (178) restricted to Ω, then it is concluded that the origin of the state space is asymptotically stable in a local way.

□

The following block diagram describes the relationship between the robot manipulator on cartesian space dynamic model and the controller structure, specifying a position controller.

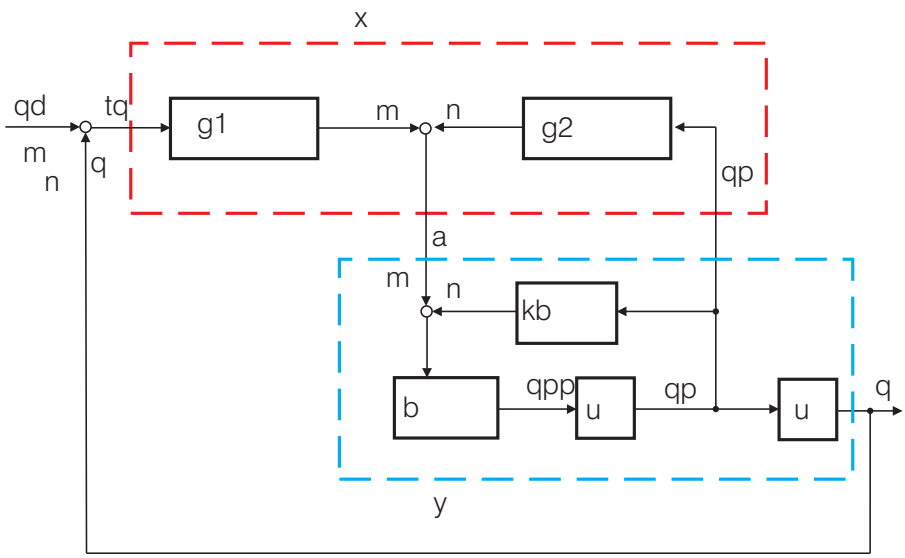


Fig. 6. Blocks of dynamic model and control scheme on cartesian space.

5.3.1.1 PD cartesian controller

In this section, we recall the stability proof of the simple PD cartesian controller which is given as:

$$\tau_x = K_p \tilde{x} - K_v \dot{x} + g(x) \tag{180}$$

where $\tilde{x} = x_d - x$ denotes the position error on cartesian coordinates, x_d is the desired position, and K_p and K_v are the propositional and derivative gains, respectively.

The control problem can be stated by selecting the design matrices K_p and K_v then the position error $\tilde{x}$ vanishes asymptotically in a local way, i.e.

$$\lim_{t \to \infty} \tilde{x}(t) = 0 \in \mathbb{R}^n. \tag{181}$$

The closed-loop system equation obtained by combining the cartesian robot model, equation (124), and control scheme, equation (180), can be written as:

$$\frac{d}{dt}\begin{bmatrix} \tilde{x} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} -\dot{x} \\ M(x)^{-1}\left[K_p\tilde{x} - K_v\dot{x} - C(x,\dot{x})\dot{x}\right] \end{bmatrix} \tag{182}$$

wthis is an autonomous differential equation and the origin of the state space is its unique equilibrium point. To accomplish the stability proof of equation (182), we proposed the following Lyapunov function candidate based on the energy shaping methodology oriented on cartesian space:

$$V(\dot{x},\tilde{x}) = \frac{\dot{x}^T M(x)\dot{x}}{2} + \frac{\tilde{x}^T K_p \tilde{x}}{2}. \tag{183}$$

The first term of $V(\dot{x},\tilde{x})$ is a positive definite function respecting to $\dot{x}$ because $M(x)$ in the case of study is a positive definite matrix. The second one of Lyapunov function candidate (183) is a positive definite function respecting to position error $\tilde{x}$, because K_p is a positive definite matrix. Therefore $V(\dot{x},\tilde{x})$ is a global positive definite and a radially unbounded function. The time derivative of Lyapunov function candidate (183) along the trajectories on the closed-loop (182),

$$\dot{V}(\dot{x},\tilde{x}) = \dot{x}^T M(x)\ddot{x} + \frac{\dot{x}^T \dot{M}(x)\dot{x}}{2} + \tilde{x}^T K_p \dot{\tilde{x}} \tag{184}$$

and after some algebra and using the property of the Coriolis and centripetal term described in section 5.2.1.2. it can be written as:

$$\dot{V}(\dot{x},\tilde{x}) = -\dot{x}^T K_v \dot{x} \leq 0 \tag{185}$$

which is a locally negative semi-definite function and therefore we conclude with stability on the equilibrium point.
In order to prove asymptotic stability in a local way, we exploit the autonomous nature of closed-loop (182) by applying the *La Salle's invariance principle*:

$$\dot{V}(\dot{x},\tilde{x}) < 0. \tag{186}$$

In the region

$$\Omega = \left\{ \begin{bmatrix} \tilde{x} \\ \dot{x} \end{bmatrix} \in \mathbb{R}^n : V(\tilde{x},\dot{x}) = 0 \right\} \tag{187}$$

the unique invariant is $\begin{bmatrix} \tilde{x}^T & \dot{x}^T \end{bmatrix}^T = 0 \in \mathbb{R}^{2n}$.

5.3.1.2 A polynomial family of PD-type cartesian controller

This control structure is a control scheme in joint space generalization proposed in [Reyes & Rosado] and [Sánchez-Sánchez & Reyes-Cortés]. The family about proposed controllers with PD-type structure and its global asymptotic stability analysis. We intend to extend the results on the simple PD controller to a large class of polynomial PD-type controllers for robot manipulators on cartesian space. Considering the following control scheme with gravity compensation given by:

$$\tau_x = \sum_{j=1}^{n} \left(K_{p_{2j-1}} \tilde{x}^{2j-1} - K_{v_{2j-1}} \dot{x}^{2j-1} \right) + g(x) \tag{188}$$

where $\tilde{x}$ denotes the position error on cartesian coordinates, x_d is the desired position, K_p and K_v are the propositional and derivative gains, respectively, and $2j-1$ give the equation the polynomial characteristic. The closed-loop system equation obtained by combining the dynamic model on the robot manipulator on cartesian, equation (124), and the control scheme, equation (188), can be written as:

$$\frac{d}{dt} \begin{bmatrix} \tilde{x} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} -\dot{x} \\ M(x)^{-1} \left[\tau_x - C(x, \dot{x}) \dot{x} \right] \end{bmatrix} \tag{189}$$

where $\tau_x = \sum_{j=1}^{n} \left(K_{p_{2j-1}} \tilde{x}^{2j-1} - K_{v_{2j-1}} \dot{x}^{2j-1} \right)$, which is an autonomous differential equation and the origin of the state space is its unique equilibrium point. To analyze the existence of the equilibrium point we have evaluated $\tilde{x}$ and $\dot{x}$ in the following way: For $I\dot{x} = 0 \Rightarrow \dot{x} = 0$, and $M(x)^{-1} K_{p_{2j-1}} \tilde{x}^{2j-1} = 0 \Rightarrow \tilde{x}^{2j-1} = 0 \Rightarrow \tilde{x} = 0$.

To make the proof of stability on the equation (189), we proposed the following Lyapunov function candidate based in the energy shaping methodology oriented on cartesian space:

$$V(\dot{x}, \tilde{x}) = \frac{\dot{x}^T M(x) \dot{x}}{2} + \frac{\sum_{j=1}^{n} K_{p_{2j-1}} \tilde{x}^{2j}}{2j}, \tag{190}$$

the first term of $V(\dot{x}, \tilde{x})$ is a positive define function respecting to $\dot{x}$ because $M(x)$ in the case of study is a positive definite matrix. The second one Lyapunov function candidate (190) is a positive definite function respecting to position error $\tilde{x}$, because K_p is a positive define matrix. Therefore $V(\dot{x}, \tilde{x})$ is a locally positive definite. The simple cartesian PD Controller is a particular case on the polynomial family of PD-type cartesian controller when $j = 1$. The time derivative of Lyapunov function candidate (190) along the trajectories of the closed-loop (189),

$$\dot{V}(\dot{x}, \tilde{x}) = \dot{x}^T M(x) \ddot{x} + \frac{\dot{x}^T \dot{M}(x) \dot{x}}{2} - \dot{\tilde{x}}^T \sum_{j=1}^{n} K_{p_{2j-1}} \tilde{x}^{2j-1} \tag{191}$$

after some algebra and using the property of the Coriolis and centripetal term described in section 5.2.1.2. it can be written as:

$$\dot{V}(\dot{x}, \tilde{x}) = -\dot{x}^T K_{v_{2j-1}} \dot{x}^{2j-1} \leq 0 \tag{192}$$

which is a locally negative semi-definite function and therefore we conclude stability on the equilibrium point. In order to prove asymptotic stability in a local way we exploit the autonomous nature of closed-loop (189) by applying the *La Salle's invariance principle*:

$$\dot{V}(\dot{x}, \tilde{x}) < 0. \tag{193}$$

In the region

$$\Omega = \left\{ \begin{bmatrix} \tilde{x} \\ \dot{x} \end{bmatrix} \in \mathbb{R}^n : V(\tilde{x}, \dot{x}) = 0 \right\} \tag{194}$$

the unique invariant is $\begin{bmatrix} \tilde{x}^T & \dot{x}^T \end{bmatrix}^T = 0 \in \mathbb{R}^{2n}$. Since (192) is a locally negative semi-definite function in full state and the Lyapunov function (190) is a radially unbounded locally positive definite function, then it satisfies:

$$0 \leq V\left(\tilde{x}\left(t\right), \dot{x}\left(t\right)\right) \leq V\left(\tilde{x}\left(0\right), \dot{x}\left(0\right)\right) \tag{195}$$

the bounds for the position error are given by:

$$\begin{aligned} &\sum_{j=1}^{n} \lambda_{\min}\left\{K_{p_{2j-1}}\right\} \left\|\tilde{x}^{2j-1}\left(t\right)\right\|^2 \\ &\leq \left\|\dot{x}\left(0\right)\right\|^2 \beta + \frac{1}{m} \sum_{j=1}^{n} \lambda_{\max}\left\{K_{p_{2j-1}}\right\} \left\|\tilde{x}^{2j-1}\left(0\right)\right\|^2 \\ &\forall \quad m \in Z^+, t \geq 0 \end{aligned} \tag{196}$$

where $\lambda_{\min}\left\{K_{p_{2j-1}}\right\}$ and $\lambda_{\max}\left\{K_{p_{2j-1}}\right\}$ represent the smallest and largest eigenvalues on the diagonal matrix $K_{p_{2j-1}}$, respectively, for derivative gain bounds are:

$$\begin{aligned} &\sum_{j=1}^{n} \lambda_{\min}\left\{K_{v_{2j-1}}\right\} \left\|\dot{x}^{2j-1}\left(t\right)\right\|^2 \\ &\leq \left\|\tilde{x}\left(0\right)\right\|^2 \beta + \frac{1}{m} \sum_{j=1}^{n} \lambda_{\max}\left\{K_{v_{2j-1}}\right\} \left\|\dot{x}^{2j-1}\left(0\right)\right\|^2 \\ &\forall \quad m \in Z^+, t \geq 0 \end{aligned} \tag{197}$$

where $\lambda_{\min}\left\{K_{v_{2j-1}}\right\}$ and $\lambda_{\max}\left\{K_{v_{2j-1}}\right\}$ represent the smallest and largest eigenvalues of the diagonal matrix $K_{v_{2j-1}}$, respectively, β is a positive constant, strictly speaking, boundlessness of the inertial matrix requires, generally, that all joints must be revolute:

$$\begin{aligned} &\beta \left\|\dot{x}\right\| \geq \left\|M\left(x\right)\dot{x}\right\| \quad \forall \quad x, \dot{x} \in \mathbb{R}^n \\ &\beta \geq n \left(\max_{i,j,x} \left|M_{ij}\left(x\right)\right| \right) \end{aligned} \tag{198}$$

where M_{ij} are elements of $M(x)$.

5.3.1.3 Pascal's Cartesian Controller

Now, we present a control structure based on Pascal's triangle,

$$\tau_x = K_p \psi_{\tilde{x}} - K_v \psi_{\dot{x}} + g(x) + f(\tau_x, \dot{x}) \tag{199}$$

where $\tilde{x}$ denotes the position error on cartesian space, K_p, K_v are the proportional and derivative gains, and $\psi_{\tilde{x}} = \tanh(\tilde{x}) \sqrt[2j]{1 + \tanh^{2j}(\tilde{x})}$, $\psi_{\dot{x}} = \tanh(\dot{x}) \sqrt[2j]{1 + \tanh^{2j}(\dot{x})}$.
The closed-loop system equation obtained by combining the dynamic model of the robot manipulator on cartesian space, equation (124), and the control structure, equation (199), can be written as:

$$\frac{d}{dt}\begin{bmatrix} \tilde{x} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} -\dot{x} \\ M(x)^{-1}\left[K_p \psi_{\tilde{x}} - K_v \psi_{\dot{x}} - C(x, \dot{x})\dot{x}\right] \end{bmatrix} \tag{200}$$

which is an autonomous differential equation and the origin of the state space is its unique equilibrium point. Based on *Pascal's triangle* and the next trigonometrical hyperbolic function,

$$\cosh^2(x) + \operatorname{senh}^2(x) = 2\cosh^2(x) - 1 \tag{201}$$

we solve the terms inside the radical, giving the following triangle:

$$\begin{array}{ccccccccccccccc}
 & & & & & 2 & & -1 & & & & & & & \\
 & & & & 2 & & 1 & & & & -2 & & & & \\
 & & & 2 & & -1 & & & 3 & & & & -3 & & \\
 & & 2 & & 1 & & & -4 & & & 6 & & & -4 & \\
 & 2 & & -1 & & & 5 & & & -10 & & & 10 & & -5
\end{array} \tag{202}$$

Inside the radical we have:

$$\begin{gathered}
2\cosh^2(x) - 1 \\
2\cosh^4(x) + 1 - 2\cosh^2(x) \\
2\cosh^6(x) - 1 + 3\cosh^2(x) - 3\cosh^4(x) \\
2\cosh^8(x) + 1 - 4\cosh^2(x) + 6\cosh^4(x) - 4\cosh^6(x) \\
2\cosh^{10}(x) - 1 + 5\cosh^2(x) - 10\cosh^4(x) + 10\cosh^6(x) - 5\cosh^8(x)
\end{gathered} \tag{203}$$

Plotting the terms within the radical we have:
When multiplying the radical for the function tanh, we modify the behavior of the function in the following way:
To carry out the stability analysis in (200), we proposed the following Lyapunov function candidate based in the energy shaping methodology oriented on cartesian space:

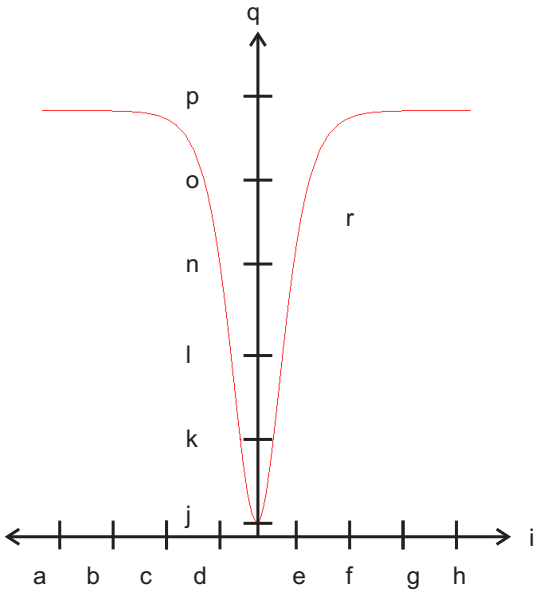


Fig. 7. terms within the radical.

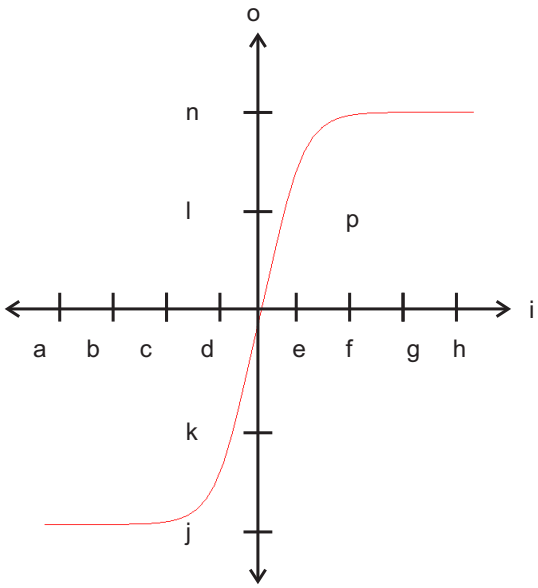


Fig. 8. Complete behavior.

$$V(\dot{x},\tilde{x}) = \frac{\dot{x}^T M(x)\dot{x}}{2} + \begin{bmatrix} \sqrt{\ln(\cosh(\tilde{x}_1))} \\ \sqrt{\ln(\cosh(\tilde{x}_2))} \\ \vdots \\ \sqrt{\ln(\cosh(\tilde{x}_n))} \end{bmatrix}^T K_p \begin{bmatrix} \sqrt{\ln(\cosh(\tilde{x}_1))} \\ \sqrt{\ln(\cosh(\tilde{x}_2))} \\ \vdots \\ \sqrt{\ln(\cosh(\tilde{x}_n))} \end{bmatrix}, \tag{204}$$

the first term on $V(\dot{x},\tilde{x})$ is a positive define function respecting to $\dot{x}$ because $M(x)$ in the case of study is a positive definite matrix. The second one of Lyapunov function candidate (204) is a positive definite function respecting to position error $\tilde{x}$, because K_p is a positive define matrix. Therefore $V(\dot{x},\tilde{x})$ is a locally positive definite. The time derivative of Lyapunov function candidate (204) along the trajectories of the closed-loop (200),

$$\dot{V}(\dot{x},\tilde{x}) = \dot{x}^T M(x)\ddot{x} + \frac{\dot{x}^T \dot{M}(x)\dot{x}}{2} + \begin{bmatrix} \sqrt{\ln(\cosh(\tilde{x}_1))} \\ \sqrt{\ln(\cosh(\tilde{x}_2))} \\ \vdots \\ \sqrt{\ln(\cosh(\tilde{x}_n))} \end{bmatrix}^T K_p \begin{bmatrix} \dfrac{\tanh \tilde{x}}{\sqrt{\ln(\cosh(\tilde{x}))}} \end{bmatrix} \dot{\tilde{x}} \tag{205}$$

after some algebra and using the property of Coriolis and centripetal term described in section 5.2.1.2. it can be written as:

$$\dot{V}(\dot{x},\tilde{x}) = -\dot{x}^T K_v \begin{bmatrix} \tanh(\dot{x}_1)\sqrt[2j]{1+\tanh^{2j}(\dot{x}_1)} \\ \tanh(\dot{x}_2)\sqrt[2j]{1+\tanh^{2j}(\dot{x}_2)} \\ \vdots \\ \tanh(\dot{x}_n)\sqrt[2j]{1+\tanh^{2j}(\dot{x}_n)} \end{bmatrix} \leq 0. \tag{206}$$

which is a locally negative semi-definite function and therefore we conclude stability on the equilibrium point. In order to prove asymptotic stability in local way we exploit the autonomous nature of closed-loop (200) by applying the *LaSalle invariance principle*:

$$\dot{V}(\dot{x},\tilde{x}) < 0. \tag{207}$$

In the region

$$\Omega = \left\{ \begin{bmatrix} \tilde{x} \\ \dot{x} \end{bmatrix} \in \mathbb{R}^n : V(\tilde{x},\dot{x}) = 0 \right\} \tag{208}$$

the unique invariant is $\begin{bmatrix} \tilde{x}^T & \dot{x}^T \end{bmatrix}^T = 0 \in \mathbb{R}^{2n}$.

5.4 Experimental Set-Up

We have designed and built an experimental system for researching on cartesian robot control algorithms and currently it is a turn key research system for developing and validation on cartesian control algorithms for robot manipulators. The experimental system is a servomotor robot manipulator with three degrees of freedom moving itself into a three dimensional space as it is shown in the Figure 5.

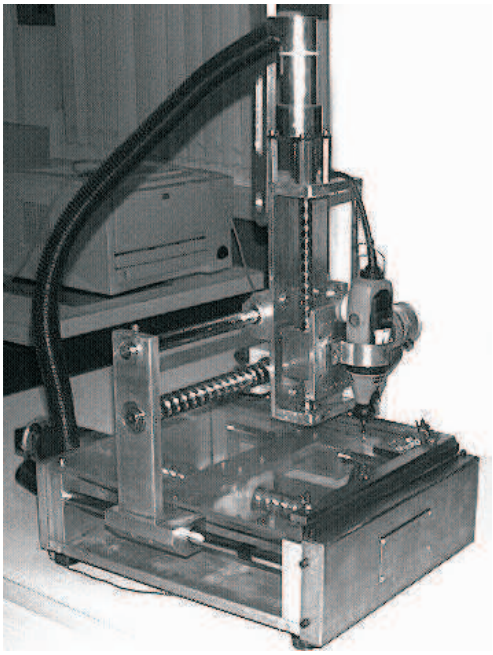


Fig. 9. Experimental Prototype. "DRILL-BOT"

The structure is made of stainless iron, direct-drive shaft with servomotors from Reliance Electronics©. Advantages in this kind of drive shaft includes a high torque. The servomotor has an Incremental Encoder from Hewlett Packard©.
Motors used in the experimental system are **E450 model [450 oz-in.]**. Servos are operated in torque mode, so the motors act a reference if torque emits a signal information about position is obtained from incremental encoders located on the motors, which have a resolution of 1024000 p/rev.

5.4.1 Experimental Results

To support our theoretical developments, this section presents an experimental comparison between three position controllers on cartesian space by using an experimental system of three degrees of freedom. To investigate the performance among controllers, they have been classified as τ_{PD} for the simple PD controller; τ_{Poly} for the polynomial family of PD-type cartesian controller; and τ_{Pascal} represent the Pascal's cartesian controller, all the control structures on cartesian space. To analyze the controllers' behavior it is necessary to compare their performances. For this reason we have used the $\mathcal{L}^2$ norm; this norm is a scalar value. A $\mathcal{L}^2$ smaller represents minor position error and thus it is the better performance. A position control experiment has been designed to compare the performances of controllers on a cartesian robot. The experiment consists on moving the manipulator's end-effector from its initial position to a fixed desired target. To the present application the desired positions were chosen as:

$$\begin{bmatrix} x_{d_1} \\ y_{d_1} \\ z_{d_1} \end{bmatrix} = \begin{bmatrix} 0.785 \\ 0.615 \\ 0.349 \end{bmatrix} \tag{209}$$

where x_{d_1}, y_{d_1} and z_{d_1} are in meters and represent the x, y and z axes in the prototype. The initial positions and velocities were set to zero (for example a home position). The friction phenomena were not modeled for compensation purpose. That is, all the controllers did not show any type of friction compensation. We should keep in mind that the phenomenon of friction doesn't have a mathematical structure to be modeled. The evaluated controllers have been written in C language. The sampling rate was executed at 2.5 ms. For proposed controller family were used the gains showed in Table 3.

Parameter	Value
K_{p_1}	359.196
K_{v_1}	35.5960
K_{p_2}	4.85400
K_{v_2}	4.36860
K_{p_3}	22.6520
K_{v_3}	3.23600

Table 3. Gains used in the experiments

5.4.2 Performance index

Robot manipulator is a very complex mechanical system, due to the nonlinear and multivariable nature on the dynamic behavior. For this reason, in the robotics community there are not well-established criteria for a proper evaluation in controllers for robots. However, it is accepted in practice comparing performance of controllers by using the scalar-valued $\mathcal{L}^2$ norm as an objective numerical measure for an entire error curve. The performance index is used to measure $\mathcal{L}^2$ norm of the position error $\tilde{x}$. A small value in $\mathcal{L}^2$ represents a smaller error and therefore it indicates a better performance. A vectorial function $\mathbb{R}^n \to \mathbb{R}^n \in \mathcal{L}^2$, if when we evaluate:

$$\mathcal{L}^2 = \sqrt{\int_0^\infty \|f(x)\|^2 dx} < \infty \tag{210}$$

where $\|f(t)\|$ is the Euclidean norm of the function on the interval; it is a scalar number. This property in vectorial functions is a measure to determine the convergence while the time increases. As the simulation time is finite we must apply the concept of effective value to calculate the deviation in the function between the simulation intervals, thus we defined $\mathcal{L}^2$ norm on the form:

$$\mathcal{L}^2 = \sqrt{\frac{1}{T}\int_0^T \|\tilde{x}\|^2 dx}. \tag{211}$$

It is necessary to count on the discreet norm representation with the purpose of facilitate its implementation:

$$\begin{aligned} \int \|\tilde{x}\|^2 \, dx \to I_k &= I_{k-1} + h \, \|\tilde{x}\|^2 \\ \mathcal{L}^2 &= \sqrt{\frac{1}{T} I_k} \end{aligned} \tag{212}$$

where h is the period of sampling; and T is the evaluation interval. This is not the unique form to obtain the discreet integral representation, being applied the rule of the trapeze we can define the integral in an alternative form:

$$\int_0^T f(t)dt \to I_k = I_{k-1} \frac{T}{2} [f_k + f_{k-1}]. \tag{213}$$

In order to obtain the performance index of proposed controllers the following program in Matlab© receives data obtained in SIMNON© applying $\mathcal{L}^2$ norm.

```
% Platform : DRILL-BOT
% Program to evaluate controllers

% Load the files
load <archivo 1>.dat -ascii;
load <archivo 2>.dat -ascii;

% Time of the system
T=10;

% Reading of:
t  =<archivo 1>(:,1);          %time
xt1=<archivo 1>(:,2);          %xtilde1
xt1=<archivo 2>(:,2);          %xtilde2

%..............integral.............
h=0.0025;
i=size(t);
ik(1)=0;

for j=2:i
    ik(j)=ik(j-1)+h*(xt1(j)*xt1(j)+xt2(j)*xt2(j));
end

%................L2 norm ............
L=sqrt(ik(j)/T)
```

Results obtained by applying $\mathcal{L}^2$ norm are in Table 4.
The performance indices graph is observed in the Figure 10.
Overall results are summarized in Figure 10 which includes the performance indexes for analyzed controllers. To average stochastic influences, data presentation in this figure represents the meaning of root-mean-square position error vector norm of ten runs. For clarity, the data

Control structure	Performance index (rad)
PD cartesian control	0.2160
A polynomial family of PD-type	0.1804
Pascal's cartesian control	0.1618

Table 4. Performance index of the evaluated controllers

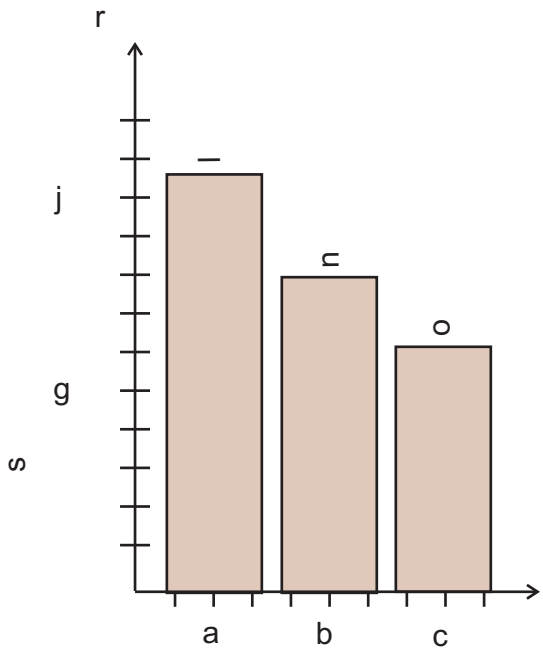


Fig. 10. Performance index of evaluated controllers

presented in Figure 10 are compared respecting to $\mathcal{L}^2$ norm of PD controller. The results from one run to another were observed to be less than 1% of their mean, which underscore the repeatability in the experiments. In general, the performance of the PD controller is improved by its counterpart.

5.4.2.1 Remarks

Through an analysis about obtained experimental data suggests the following results:

- Note that *A polynomial family of PD-type cartesian controller* and the *Pascal's cartesian controller* improves the performance obtained by the *PD cartesian controller*. The proposed controller families effectively exploits its exponential capability in order to enhance the position error, having a short transient phase and a small steady-state error. Fast convergence can be obtained (faster response). Consequently, the control performance is increased in comparison with the aforementioned controller.

- As it can be seen, the position error is bounded to increase the power those where the error signal is to be raised. However, for stability purposes, tuning procedure for the control schemes are sufficient to select a proportional and derivative gains as diagonal matrix, in order to ensure asymptotic stability in a local way.
- Nevertheless, in spite of the presence of friction, signals on position error are acceptably small for proposed families.

The problem about position control for robot manipulators could correspond to the configuration of a simple pick and place robot or a drilling robot. For example, when the robot reaches the desired point, it can return to the initial position. If this process is repetitive (robot plus controller), then it would be a simple pick and place robot used for manufacturing systems. Other applications could be: palletizing materials, press to press transferring, windshield glass handling, automotive components handling, cookie and bottle packing; and drilling. In those applications, the time spent on transferring a workpiece from one station to next or doing one or several perforations still high. In our prototype case, it becomes evident the use of position control due to the coordinates where a bore is desired. It is important to observe that after each perforation done by the robot it returns to their Initial position.

6. Conclusions

As a result about the assumptions and demonstrations realized in this chapter, is possible to conclude that the cartesian control is local. This characteristic restricts the system with its work area and it offers us a better understanding of the space in the location of the end-effector.
In this chapter we have described an experimental prototype for testing cartesian robot controllers with formal stability proof, which allows the programming a general class of cartesian robot controllers. The goal of the test system is to support the research as well as developing new cartesian control algorithms for robot manipulators. Our theoretical results are the propose on cartesian controllers. We have shown asymptotic stability in a local way by using Lyapunov's theory. Experiments on cartesian robot manipulator have been carried out to show the stability and performance for the cartesian controllers. For stability purposes, tuning procedure for the new scheme is enough to select a proportional and derivative gains as diagonal matrix in order to ensure asymptotic stability in a local way. However, the actual choice of gains can also produce torque saturation on the actuators, thus deteriorating the control system performance. To overcome these drawbacks, in this chapter it has been proposed a simple tuning rule. The scheme's performances were compared with the PD controller algorithm on cartesian coordinates by using a real time experiment on three degrees of freedom prototype. From experimental results the new scheme produced a brief transient and minimum steady-state position error.
In general, controllers showed better performance among the evaluated controllers and this statement can be proven by observing the performance index on the controllers. We can conclude that Pascal's cartesian controller is faster than PD cartesian controller and the polynomial family of PD-type cartesian controller, reason why the Pascal's cartesian controller offers some advantages in robot's control and in the time of operation.

7. Acknowledgement

The authors thanks the support received by Electronics Science Faculty on Autonomous University of Puebla, Mexico; and also by the revision on manuscript to Lic. Oscar R. Quirarte-Castellanos.

8. References

Craig J. J. (1989) *Introduction to Robotics, Mechanics and Control*, Addison-Wesley Publishing Company, ISBN 0-201-09528-9, USA

D'Souza A., Vijayakumar S. & Schaal S. (2001) *Learning inverse kinematics*, in Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 1, 298-303.

Gonin R. & H. Money A. (1989) *Nonlinear L_P-norm Estimation*, CRC, ISBN 0-8247-8125-2, USA.

Hauser W. (1965) *Introduction to the principles of mechanics*, Addison-Wesley Publishing Company, Inc. Massachusetts, USA.

Kelly R. & Santibáñez V.(2003) *Control de Movimiento de Robots Manipuladores*, Pearson Education SA, ISBN 84-205-3831-0, Madrid, España.

Olsson H.; Aström K. J.; Canudas de Wit C.; Gäfvert M & Lischinsky P. (1998). Friction Models and Friction Compensation. *European Journal of Control*, Vol. 4, No. 3., (Dec. 1998) 176-195.

Reyes F. & Rosado A.; (2005) Polynomial family of PD-type controllers for robot manipulators, *Journal on Control Engineering Practice*, Vol. 13, No. 4, (April 2005), 441-450, ISSN 0967-0661

Sánchez-Sánchez P. & Reyes-Cortés F. (2005). *Pascal's cartesian controllers*, International Conference on Mechanics and Automation, Niagara Falls, Ontario, Canada, 94-100, ISBN 0-7803-9044-X

Sánchez-Sánchez P. & Reyes-Cortés F. (2005) Position control through Pascal's cartesian controller, *Transactions on Systems*, Vol. 4, No. 12, 2417-2424, ISSN 1109-2777

Sánchez-Sánchez P. & Reyes-Cortés F. (2006) *A new position controller: Pascal's cartesian controllers*, International Conference on Control and Applications, Montreal, Quebec, Canada, 126-132, ISBN 0-88986-596-5

Sánchez-Sánchez P. & Reyes-Cortés F. (2008) A Polynomial Family of PD-Type Cartesian Controllers, *International Journal of Robotics and Automation*, Vol. 23, No. 2, 79-87, ISSN 0826-8185

Santibáñes V.; Kelly R. & Reyes-Cortés F. (1998) A New Set-point controller with Bounded Torques for Robot Manipulators. *IEEE Transactions on Industrial Electronics*, Vol. 45, No. 1, 126-133

Spong, M. W & Vidyasagar M. (1989) *Robot Dynamics and Control*, John Wiley & Sons, ISBN 0-471-61243-X, USA.

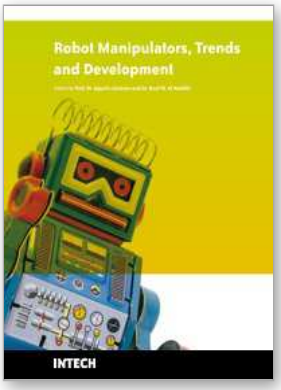
Spong, M. W., Lewis F. L. & Adballah C. T. (1993) *Robot Control, Dynamics, Motion Planning and Analysis*, IEEE Press, ISBN 0-7803-0404-7, USA.

Spong, M. W., Hutchinson S. & Vidyasagar M. (2006) *Robot modeling and Control*, John Wiley & Sons, Inc, ISBN 0-471-64990-8, USA.

Synge L. J. (2008) *Principles of mechanics*, Milward Press, ISBN 1-443-72701-6, USA.

Takegaki M. & Arimoto S. (1981) A New Feedback Method for Dynamic Control of Manipulators, *Journal of Dynamics System, Measurement and Control*, Vol. 103, No. 2, 119-125.

Taylor R. J. (2005) *Classical Mechanics*, University Science Books, ISBN 1-8913-8922-X, USA.



Robot Manipulators Trends and Development

Edited by Agustin Jimenez and Basil M Al Hadithi

ISBN 978-953-307-073-5

Hard cover, 666 pages

Publisher InTech

Published online 01, March, 2010

Published in print edition March, 2010

This book presents the most recent research advances in robot manipulators. It offers a complete survey to the kinematic and dynamic modelling, simulation, computer vision, software engineering, optimization and design of control algorithms applied for robotic systems. It is devoted for a large scale of applications, such as manufacturing, manipulation, medicine and automation. Several control methods are included such as optimal, adaptive, robust, force, fuzzy and neural network control strategies. The trajectory planning is discussed in details for point-to-point and path motions control. The results in obtained in this book are expected to be of great interest for researchers, engineers, scientists and students, in engineering studies and industrial sectors related to robot modelling, design, control, and application. The book also details theoretical, mathematical and practical requirements for mathematicians and control engineers. It surveys recent techniques in modelling, computer simulation and implementation of advanced and intelligent controllers.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Pablo Sanchez-Sanchez and Fernando Reyes-Cortes (2010). Cartesian Control for Robot Manipulators, Robot Manipulators Trends and Development, Agustin Jimenez and Basil M Al Hadithi (Ed.), ISBN: 978-953-307-073-5, InTech, Available from: http://www.intechopen.com/books/robot-manipulators-trends-and-development/cartesian-control-for-robot-manipulators

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821