10

Design of Adaptive Controllers based on Christoffel Symbols of First Kind

Juan Ignacio Mulero-Martínez
Technical Universty of Cartagena
Spain

1. Introduction

The present chapter is aimed at systematically exposing the reader to certain modern trends in designing advanced robot controllers. More specifically, it focuses on a new and improved method for building suitable adaptive controllers guaranteeing asymptotic stability. It covers the complete design cycle, while providing detailed insight into most critical design issues of the different building blocks. In this sense, it takes a more global design perspective in jointly examining the design space at control level as well as at the architectural level.

The primary purpose is to provide insight and intuition into adaptive controllers based on Christoffel symbols of first kind for a serial-link robot arm, (Mulero-Martínez, 2007a). These controllers are referred to as static since the positional dependence of the nonlinear functions. In this context, the preferred method of nonlinear compensation is the method of building emulators. Often, however, the full power of the method is overlooked, and very few works deal with these techniques at the level of detail that the subject deserves. As a result, the chapter fills that gap and includes the type of information required to help control engineers to apply the method to robot manipulators. Developed in this chapter are several deep connections between dynamics analysis and implementation emphasizing the powerful adaptive methods that emerge when separate techniques from each area are properly assembled in a larger context.

After beginning with a comprehensive presentation of the fundamentals of these techniques, the chapter addresses the problem of factorization of the Coriolis/centripetal matrix, (Mulero-Martinez, 2009). This aspect is crucial when designing non-linear compensators by emulation. At this point, it is provided a concise and didactically structured description of the design of emulators as matters stand, (Mulero-Martinez, 2006). Specifically, emulators are split up into sub-emulators to improve and simplify the design of controllers while making faster the updating of parameters. From a practical point of view, the implementation is developed by resorting to parametric structures. This means to obtain a set of system's own function as regression functions.

Most of the adaptive schemes start from the notable property of linearity in the parameters, which lead naturally to equivalent structures when designing emulators for the nonlinear terms. When the linearity in the parameters (LIP) is considered as a first assumption in the development of adaptive schemes, it is clear that there exists a strong connection to the LIP emulators formulated in terms of a regression matrix and a vector of parameters. The main difference between standard adaptive schemes and the proposed approach stems from the idea of developing efficient controllers. The present work is aimed by attempts to mitigate the "curse of dimensionality" by exploiting the representation properties associated with the matrix of Coriolis/centripetal effects. By recalling the connection between LIP representation of robot manipulators and LIP adaptive emulators, it can be asserted that standard scheme matches perfectly with a dynamic emulator. Thus, the regression matrix, depend not only on the position joint variables but also on the velocity and acceleration variables.

As regards to the control, a novel theorem guarantees the stability for the whole system and is based on the Lyapunov energy. The proof is generalized to cope with a realistic case where both a functional reconstruction error and an external disturbance are present. It should be observed that the functional reconstruction error is caused by not using a number of regression functions appropiately distributed in the space. As a result, these considerations lead to a quite different approach, since it is required to analyze the initial conditions of the errors to guarantee the validity of the approximation. The specification of a range of validity causes that the stability holds only inside a compact set. As a consequence, the proof guarantees semi-global stability as opposed to the standard schemes where the stability is attained in the whole state space, in a global sense. Apart from these considerations, a number of remarks have been made to address some special aspects such as the boundedness of the parameters, the ultimately uniformly boundedness of all the signals and the stability in the ideal case.

The main benefit of the proposed controller is that it allows to derive tuning laws only for inertia, gravitational and frictional parameters. The Coriolis parameters are not necessary to be used because of the approximation based on Christoffel symbols. This is very useful to implement adaptive controllers since the number of nodes diminishes and the computational performance improves. Previously, an extensive analysis of the mechanical properties for a robot has been discussed. The regression functions for the adaptive controller depend on the non-linear functions associated with the inertia matrix, and therefore, a discretization of positions could be done for the inertia matrix. This is a very useful aspect because the position space for a revolute robot is compact and in consequence, the number of nodes is limited to approximate a non-linear function.

The plan of the chapter is as follows. In section 2 the representation properties for the Coriolis/centripetal matrix are analysed. An interpretation for the Coriolis/centripetal matrix is presented and the description by means of the Christoffel symbols of first kind and fundamental matrices are provided. In section 3, emulators are used to approximate the non-linearities of a robot using the properties presented in the previous section and the Kronecker product. The next section presents the design of the adaptive controller in terms of a control law and a parameter updating law. This section concludes with a theorem that guarantees the stability for the whole system and is based on the Lyapunov energy. Finally an example of a 2-dof robot arm is used to illustrate the theorem.

2. Representation of the Coriolis/Centripetal Matrix. Fundamental Matrices

In this section some notions regarding the representation of the Coriolis/centripetal matrices are introduced. All the ideas presented here constitute an original contribution and have many interesting implications in the field of robotics. To this end, fundamental matrices are introduced and described in terms of their structure. Moreover, some emerging properties are analyzed, allowing one to build the Coriolis/centripetal matrix in a simple way. Let start with the definition of the matrix M_D which from now on will be called the inertia derivative matrix.

Definition 1:

$$M_D(q,x) = \sum_{j=1}^{n} \frac{\partial M(q)}{\partial q_j} x e_j^T \tag{1}$$

where $M(q)$ is a generalized inertia matrix of dimension $n \times n$ a unitary vector of dimension n with a value 1 in the position j and x is an arbitrary vector of dimension n. It is noted that if x represents the generalized velocity vector, the matrix M_D will stand for the gradient of the generalized momentum with respect to the position coordinates q. This means that the gradient of the kinetic energy as a quadratic form $\frac{1}{2}x^T M(q) x$ relative to the joint position can be written as $\frac{1}{2}M_D^T(q,x)x$. Another representation of M_D is showed below.

Property 1: The matrix $M_D(q,x)$ can be expressed as

$$M_D(q,x) = \sum_{i=1}^{n} \frac{\partial M_i}{\partial q} x_i \tag{2}$$

Proof: It is very easy to show that $\frac{\partial M(q)}{\partial q_j} x e_j^T = \sum_{i=1}^{n} \frac{\partial M_i}{\partial q_j} x_i e_j^T$ since the following intermediate equation is obtained

$$\frac{\partial M(q)}{\partial q_j} x e_j^T = \left(\sum_{i=1}^{n} \frac{\partial M_i}{\partial q_j} e_i^T \right) x e_j^T = \sum_{i=1}^{n} \frac{\partial M_i}{\partial q_j} x_i e_j^T \tag{3}$$

Using the definition 1 and the identity (3) the hypothesis of the property is concluded

$$M_D(q,x) = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial M_i}{\partial q_j} x_i e_j^T = \sum_{i=1}^{n} \frac{\partial M_i}{\partial q} x_i \tag{4}$$

Q.E.D.

Now a new matrix will be introduced and from now on will be called as inertia velocity matrix, playing a central role in the representation theory.

Definition 2: Let define $M_v(q,x)$ in the following way

$$M_v(q,x) = \left(\frac{\partial M_1}{\partial q} x, \ldots, \frac{\partial M_n}{\partial q} x \right) = \sum_{i=1}^{n} \frac{\partial M_i}{\partial q} x e_i^T \tag{5}$$

The inertia velocity matrix $M_v(q,x)$ receives its name from the fact that when $x=\dot{q}$, the term $M_v(q,\dot{q})$ will be the time differentiation of the generalized inertia matrix, i.e. $\dot{M}(q)$. The following property provides an alternative way to write the matrix M_V.

Property 2: The inertia velocity matrix can be also expressed as

$$M_v(q,x)=\sum_{i=1}^{n}\frac{\partial M(q)}{\partial q_i}x_i \tag{6}$$

Proof: It is known that $\frac{\partial M_i}{\partial q}xe_i^T=\sum_{j=1}^{n}\frac{\partial M_i}{\partial q_j}x_j e_i^T$ and using the definition 2 the property is proved as follows

$$M_v(q,x)=\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{\partial M_i}{\partial q_j}x_j e_i^T=\sum_{j=1}^{n}\left(\sum_{i=1}^{n}\frac{\partial M_i}{\partial q_j}e_i^T\right)x_j=\sum_{j=1}^{n}\frac{\partial M(q)}{\partial q_j}x_j \tag{7}$$

Q.E.D.

2.1 Properties of the fundamental matrices

Subsequently, some properties related to the fundamental matrices are analyzed. Following a systematic methodology, the properties have been classified into two groups: commutation properties and representation properties.

2.1.1. Commutation properties

Commutation properties permit interchange of an external arbitrary vector y and a vector x passed to a fundamental matrix as an argument. The following property makes possible the commutation while keeping the type of the fundamental matrices. This means that the transpose of the inertia derivative matrix can be transformed into the same structure by simply interchanging the roles of x and y.

Property 3: $M_D^T(q,x)y=M_D^T(q,y)x$

The proof of the last property follows directly from the definition of M_D. The following property allows to pass from a type of fundamental matrix to another commuting the vectors x and y.

Property 4: $M_v(q,x)y=M_D(q,y)x$

Proof :

$$M_v(q,x)y=\sum_{i=1}^{n}\frac{\partial M_i}{\partial q}xe_i^T y=\sum_{i=1}^{n}\frac{\partial M_i}{\partial q}xy_i=M_D(q,y)x \tag{8}$$

Q.E.D.

2.1.2. Properties of representation of the Coriolis/centripetal matrix

These properties are very important to describe the Coriolis/centripetal matrix from the fundamental structures.

Property 5:

$$M_D^T(q,\dot{q}) = \sum_{i=1}^{n} \tfrac{\partial M_i^T}{\partial q} \dot{q} e_i^T \tag{9}$$

Proof: First of all, the transpose of the inertia derivative matrix can be represented as $M_D^T(q,\dot{q}) = \sum_{j=1}^{n} e_j \dot{q}^T \sum_{i=1}^{n} \tfrac{\partial M_i}{\partial q_j} e_i^T$ using the definition 1 and the fact that the differentiation of the inertia matrix with respect to the generalized coordinate q_j is $\tfrac{\partial M(q)}{\partial q_j} = \sum_{i=1}^{n} \tfrac{\partial M_i}{\partial q_j} e_i^T$. Since scalar product is commutative the following expression is derived, $\dot{q}^T \tfrac{\partial M_i}{\partial q_j} = \tfrac{\partial M_i^T}{\partial q_j} \dot{q}$, and as a result $M_D^T(q,\dot{q}) = \sum_{j=1}^{n} e_j \sum_{i=1}^{n} \tfrac{\partial M_i^T}{\partial q_j} \dot{q} e_i^T$. The order of summation is needed to be permutated to get the proposed identity.

$$M_D^T(q,\dot{q}) = \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{\partial M_i}{\partial q_j} e_j^T\right)^T \dot{q} e_i^T = \sum_{i=1}^{n} \frac{\partial M_i^T}{\partial q} \dot{q} e_i^T \tag{10}$$

Q.E.D.

Below, some representations of the Coriolis/centripetal matrix are introduced as properties derived from M_D and M_v .

Property 6: The matrix of Coriolis/centripetal effects can be expressed as

$$C(q,\dot{q}) = \frac{1}{2}\left(M_v(q,\dot{q}) + M_D(q,\dot{q}) - M_D^T(q,\dot{q})\right) = \frac{1}{2}\left(M_v(q,\dot{q}) + J(q,\dot{q})\right) \tag{11}$$

where $J(q,\dot{q}) = M_D(q,\dot{q}) - M_D^T(q,\dot{q})$ is a skew symmetric matrix, i.e. $J(q,\dot{q}) = -J^T(q,\dot{q})$.

Proof: This is an immediate consequence of the representation of $C(q,\dot{q})$ by means of the property 1 and the fact that the inertia velocity matrix is $M_v(q,\dot{q}) = \dot{M}(q)$.

In a general way, the following representation can be derived

$$C(q,x) = \frac{1}{2}\left(M_v(q,x) + J(q,x)\right) \tag{12}$$

where x is a vector of dimension n and $J(q,x) = M_D(q,x) - M_D^T(q,x)$ is a skew symmetric matrix. It is remarkable that the definition of the Coriolis/centripetal matrix presented above, is different from the definition given by (Wen,1988) in the identity 2, $C(q,z)z = \tfrac{1}{2}\left(M_D(q,z) - J(q,z)z\right)$. An interesting property which is a direct implication of the property 4 is that, by setting $x = y$ in $C(q,x)y$.

Property 7: The Coriolis/centripetal force can be represented as

$$C(q,\dot{q})\dot{q} = \left(\dot{M}(q) - \frac{1}{2} M_D^T(q,\dot{q})\right)\dot{q} \tag{13}$$

or in a general form as

$$C(q,x)x = M_v(q,x)x - \frac{1}{2}M_D^T(q,x)x \tag{14}$$

where x is an arbitrary vector of dimension n :

Property 8: The Coriolis/centripetal matrix commutes with external vectors

$$C(q,x)y = C(q,y)x \tag{15}$$

Proof: In order to see this point the representation of the Coriolis/centripetal matrix will be used as a sum of the inertia velocity matrix, $M_v(q,x)$ and the skew symmetric matrix $J(q,x)$ given by equation (12).

$$C(q,x)y = \frac{1}{2}\left(M_v(q,x)y + J(q,x)y\right) \tag{16}$$

On one hand, it is known that $J(q,x) = M_D(q,x) - M_D^T(q,x)$ .Using the commutation properties 2 and 3 the following expression is derived

$$C(q,x)y = \frac{1}{2}\left(M_D(q,y)x - M_D^T(q,y)x + M_D(q,x)y\right) \tag{17}$$

On other hand, $J(q,y)x = M_D(q,y)x - M_D^T(q,y)x$ and then applying the commutation property 3 to $M_D(q,x)y$ the following result is achieved as claimed

$$C(q,x)y = \frac{1}{2}\left(J(q,y)x + M_v(q,y)x\right) = C(q,y)x \tag{18}$$

Q.E.D

3. Design of Emulators for Robot Manipulators.

3.1 Functional and Linear Parameterization.

The approach that follows is founded on the idea to find an emulator as a function close to the non-linear terms involved in the dynamics equations of a robot manipulator. In order to get a model from a practical point of view, uncertainties in the nonlinear terms. getting arise from the partial information about the exact structure of the dynamics, must be taken into account. The inaccuracies of a model can be classified into two classes: structured and unstructured uncertainties. The first kind of uncertainties comes out from the inaccuracies of the parameters whereas the unstructured uncertainties are related to unmodeled dynamics, see (Slotine & Li,1991). Thus, the uncertainties can be adaptatively compensated by defining each coefficient as a separate parameter so that the dynamics can be expressed in the linear in the parameters (LIP) and this means that nonlinearities can be split up into an unknown vector of physical parameters P and a known matrix of basis nonlinear functions $\Psi(q,\dot{q},x,y)$ comprising the elements of $M(q)$, $C(q,\dot{q})$, $G(q)$ and $F(q,\dot{q})$, referred to as regression matrix. Therefore, the nonlinear function $f(x)$ can be written in this sense adding a term of error ε, see (Ge et al., 1998).

$$f(q,\dot{q},x,y)=\Psi(q,\dot{q},x,y)P+\varepsilon \tag{19}$$

The linearity of the parameters is the major structural property of robot manipulators and has been analyzed in (Lewis et al., 2003). This linear factorization is always possible to be done for the rigid body dynamics of a fixed-based manipulator as long as the physical uncertainty is on the mass properties of the robot links. Furthermore, linearity of the parameters is the first assumption in the most of adaptive controllers. An alternative representation of the nonlinear component is as follows

$$f(q,\dot{q},x,y)=R(q,\dot{q})v(x,y) \tag{20}$$

where $R(q,\dot{q})=\begin{pmatrix}M(q) & C(q,\dot{q}) & G(q)\end{pmatrix}\in\mathbb{R}^{n\times(2n+1)}$ and $v=\begin{pmatrix}y^T & x^T & 1\end{pmatrix}\in\mathbb{R}^{2n+1}$. This factorization is always attainable whereas the linearity in the parameters (LIP) is only obtained under some circumstances. In the literature, emulators based on regression matrices have been used to approximate the nonlinear dynamics as a whole, as follows

$$\mathcal{N}(q,\dot{q},\ddot{q})=M(q)\ddot{q}+C(q,\dot{q})\dot{q}+G(q) \tag{21}$$

As an attempt to obtain more efficient computation, the emulator approximating the nonlinearity $f(x)$ is split up into several smaller components:

$$f(q,\dot{q},x,y)=f_m(q,y)+f_c(q,\dot{q},x)+f_g(q)+f_f(q,\dot{q}) \tag{22}$$

The function $f_m(q,x)=M(q)y$, stands for the nonlinearity of inertial terms and can be written taking into account that the components of $M(q)$ are continuous functions of their arguments so that each component can be uniformly approximated on any compact subset of the state space by an appropriately designed emulator.

From now on we assume that the number of parameters to approximate the column i of a matrix is l_i.

3.2 Inertia matrix $M(q)$.

3.2.1 Ideal Case

The inertia matrix $M(q)$ consists of column vectors $M_i(q)$ that can be approximated by regression matrices

$$M_i(q)=\Psi_{m_i}(q)\xi_{m_i} \tag{23}$$

where $\Psi_{m_i}(q) \in \mathcal{M}_{n\times l_i}(\mathbb{R})$ is a regression matrix of known robot functions and $\xi_{m_i} \in \mathbb{R}^{l_i}$ is a vector of unknown parameters (e.g. masses, inertia moments and link parameters). Then, with an arbitrary vector $x \in \mathbb{R}^n$, there follows the decomposition

$$M(q)x = \sum_{i=1}^{n} M_i(q)x_i = \left(\sum_{i=1}^{n} \Psi_{m_i}(q)x_i\xi_{m_i}\right) \tag{24}$$

On the assumption that the same regression matrix serves for each column, i.e. $\Psi_{m_1}(q) = \Psi_{m_2}(q) = \cdots \Psi_{m_n}(q) = \Psi_m(q) \in \mathcal{M}_{n\times l}(\mathbb{R})$, one may rewrite equation (24) by resorting to the Kronecker product as

$$M(q)x = \Phi_m(q,x)\xi_m \tag{25}$$

where $\xi_m \in \mathbb{R}^l$ is an l-dimensional vector of parameters and $\Phi_m(q,x) = \left(x^T \otimes \Psi_m(q)\right) \in \mathcal{M}_{n\times l}(\mathbb{R})$ is the regression matrix of the generalized inertia matrix.

3.2.2 Real Case

Given a closed, bounded subset $\Omega \subseteq \mathbb{R}^n$, and a specified accuracy ε_{m_i}, there exist values for the design parameters l_i and ξ_{m_i} so that for all $q(t) \in \Omega$ the following inequality is satisfied:

$$\left\| M_i(q) - \Psi_{m_i}(q)\xi_{m_i} \right\| \le \varepsilon_{m_i} \tag{26}$$

For the sake of simplicity, we have assumed that $\Psi_{m_i}(q)$ are equal for all $i = 1,\ldots,n$. The inertia matrix consists of column vectors that can be approximated as

$$M(q) = \sum_{i=1}^{n} M_i(q)e_i^T = \left(\sum_{i=1}^{n} \Psi_{m_i}(q)\xi_{m_i}e_i^T\right) + \varepsilon_m(q) \tag{27}$$

where $\varepsilon_m(q) \in \mathbb{R}^{n\times n}$ is the functional reconstruction error.

It is convenient to underline that the vector function $\Phi_m(q,x)$ only can be expressed by the Kronecker product whenever the hypothesis $\Phi_m(q,x) = \Psi_{m_i}(q) = \Psi_m(q)$ holds for all $i = 1,\ldots,n$. As a consequence, a linear transformation with a change of basis can be derived in terms of the Kronecker product

$$M(q)x = \Phi_m(q,x)\xi_m + E_m(q,x) \tag{28}$$

The expansion given by (28) actually approximates the linear transformation $M(q)x$ to a certain degree of accuracy $E_m(q,x)=\varepsilon_m x$ by using an emulator with the position joint vector as the input. Specifically, the emulator in (28) has the vector $\left(q^T,x^T\right)^T$ as its input and n outputs, and a collection of simple computing elements $\Phi_m(q,x)$ are used to approximate the function $f_m(q,x)$.

3.3 Fundamental matrices.

For the sake of simplicity in the following, the matrix derivative of $\Psi_m(q)$ with $q\in\mathbb{R}^n$ will be denoted as

$$\frac{\partial\Psi_m(q)}{\partial q}\triangleq\begin{pmatrix}\frac{\partial\Psi_m(q)}{\partial q_1}\\ \cdots\\ \frac{\partial\Psi_m(q)}{\partial q_n}\end{pmatrix}\in\mathcal{M}_{n^2\times l}(\mathbb{R}) \tag{29}$$

3.3.1 Inertia Derivative Matrix $M_D(q,x)$

3.3.1.1 Real Case

It is possible to express the inertia derivative matrix as consisting of two terms, the approximation component $\hat{M}_D(q,x)$ and the reconstruction error $E_D(q,x)$.

$$M_D(q,x)=\sum_{j=1}^{n}\frac{\partial\hat{M}(q)}{\partial q_j}xe_j^T+\sum_{j=1}^{n}\frac{\partial E_m(q,x)}{\partial q_j}e_j^T \tag{30}$$

Following the discussion given in the ideal case, the approximation term $\hat{M}_D(q,x)y$ can be expressed in the regression form:

$$\hat{M}_D(q,x)y=\Phi_D(q,x,y)\xi_m \tag{31}$$

3.3.1.2 Ideal Case

The following lemmas are helpful for characterizing the inertia derivative matrix in LIP form.

Lemma 1 (LIP form of M_D). For arbitrary vectors $x,y\in\mathbb{R}^n$, $q\in\Omega_q\subseteq\mathbb{R}^n$, the inertia derivative matrix can be written in the LIP form as follows

$$M_D(q,x)y=\Phi_D(q,x,y)\xi_m \tag{32}$$

where $\xi_m\in\mathbb{R}^l$ is the parameter vector of the generalized inertia matrix and

$\Phi_D \in \mathcal{M}_{n\times l}(\mathbb{R})$ a regression matrix:

$$\begin{aligned}\Phi_D(q,x,y) &= \left(y^T \otimes I_n\right)\psi(q,x)\\ \psi(q,x) &= \left(x^T \otimes \frac{\partial \Psi_m(q)}{\partial q}\right)\end{aligned} \tag{33}$$

Proof: This fact can be straightforward proved by resorting to the definition of M_D given above:

$$M_D(q,x)y = \sum_{i=1}^{n} \frac{\partial M(q)}{\partial q_i} x y_i = \left(\sum_{i=1}^{n} \frac{\partial \Phi_m(q,x)}{\partial q_i} y_i\right)\xi_m = \left(x^T \otimes \sum_{i=1}^{n} \frac{\partial \Psi_m(q)}{\partial q_i} y_i\right)\xi_m = \Phi_D(q,x,y)\xi_m \tag{34}$$

By closely examining the structure of Φ_D and using the property

$$\sum_{i=1}^{n} \frac{\partial \Psi_m(q)}{\partial q_i} y_i = \left(y^T \otimes I_n\right)\frac{\partial \Psi_m(q)}{\partial q} \tag{35}$$

it is easy to conclude that

$$\begin{aligned}\Phi_D(q,x,y) &= x^T \otimes \left(\left(y^T \otimes I_n\right)\frac{\partial \Psi_m(q)}{\partial q}\right) = \left[x \otimes \left(\frac{\partial \Psi_m^T(q)}{\partial q}\left(y \otimes I_n\right)\right)\right]^T =\\ &= \left[\left(x \otimes \frac{\partial \Psi_m^T(q)}{\partial q}\right)\left(y \otimes I_n\right)\right]^T = \left(y^T \otimes I_n\right)\psi(q,x)\end{aligned} \tag{36}$$

3.3.2 Fundamental Matrix $M_v(q,x)$

3.3.2.1 Ideal Case

Lemma 2. For arbitrary vectors $x,y \in \mathbb{R}^n$, $q \in \Omega_q \subseteq \mathbb{R}^n$, the inertia velocity matrix can be written as

$$M_v(q,x)y = \Phi_v(q,x,y)\xi_m \tag{37}$$

where $\Phi_v(q,x,y) = \sum_{i=1}^{n} \frac{\partial \Phi_m(q,y)}{\partial q_i} x_i$.

Proof: Owing to the commutation property of fundamental matrices the result is direct.

Remark: It is easy to show that the Jacobian matrix $\Phi_v(q,x,y)$ also satisfies the commutation property, $\Phi_v(q,x,y) = \Phi_D(q,y,x)$ .This is an immediate consequence of the commutation property of $M_v(q,y)$ and $M_D(q,x)$ as given in the identity 3 .

3.3.2.2 Real Case

The inertia velocity matrix $M_v(q,x)$ can be seen as consisting of an approximation term $\hat{M}_v(q,x)$ and a reconstruction error $E_v(q,x)$. The approximation term $\hat{M}_v(q,x)$ can be written in terms of the Jacobian of $\Phi_v(q,x,y)$ as

$$\hat{M}_v(q,x)y = \Phi_v(q,x,y)\xi_m \tag{38}$$

and the error $E_v(q,x)$ is derived from the gradient of the inertia error $\varepsilon_m(q)$.

$$E_v(q,x) = \sum_{i=1}^{n} \frac{\partial \varepsilon_m}{\partial q_i} x_i = \sum_{i=1}^{n} \frac{\partial \varepsilon_m(q)}{\partial q_i} x_i \tag{39}$$

3.3.3 Transpose of the Inertia Derivative Matrix

Now that it is known how M_D can be written in LIP form, the main challenge, now, is to extend this result to its transpose. This is more difficult since involves a permutation matrix as stated in the following lemma.

Lemma 3: Let $x,y \in \mathbb{R}^n$, $q \in \Omega_q \subseteq \mathbb{R}^n$ arbitrary vectors with appropriate units and P a permutation matrix1 defined as follows

$$P(n,m) = \sum_{i=1}^{n}\sum_{j=1}^{n} E_{ij} \otimes E_{ji} \tag{40}$$

where $E_{ij} = e_i e_j^T$ is a unit matrix having 1 in position (i,j) and all other entries are zero. Under these conditions, the term $M_D^T(q,x)$ can be written in LIP form as follows

$$M_D^T(q,x)y = \left(x^T \otimes I_n\right)P(n,n)\psi(q,y)\xi_m \tag{41}$$

Proof: By virtue of definition 1 and lemma 1, the generalized force $M_D^T(q,x)y$ can be directly derived in the LIP form which can be detailed into

$$\begin{aligned} M_D^T(q,x)y &= \sum_{i=1}^{n} e_i \left(x^T \frac{\partial M(q)}{\partial q_i} y \right) = \left(I_n \otimes x^T\right)\frac{\partial M(q)}{\partial q} y = \\ &= \left(I_n \otimes x^T\right)\frac{\partial \Phi_m(q,y)}{\partial q}\xi_m = \left(I_n \otimes x^T\right)\left(y^T \otimes \frac{\partial \Psi_m(q)}{\partial q}\right)\xi_m \end{aligned} \tag{42}$$

[1] $P(m,n)$ is a matrix of zeroes and ones for which $P(m,n)^{-1} = P(m,n)^T = P(m,n)$.

With reference to the property $B\otimes A = P(m,p)^{T}(A\otimes B)P(n,q)$ for all $A\in\mathcal{M}_{m\times n}(\mathbb{R})$ and $B\in\mathcal{M}_{p\times q}(\mathbb{R})$, (see Corollary 4.3.10., p. 260, in (Horn & Johnson,1999)), the term $\left(x^{T}\otimes I_{n}\right)$ can be commutated as

$$\left(x^{T}\otimes I_{n}\right)=P(n,1)^{T}\left(I_{n}\otimes x^{T}\right)P(n,n) \tag{43}$$

The last identity can be simplified further by exploiting the structure of the permutation matrices. In particular, it is easy to show that $P(n,1)=P(1,n)=I_{n}$ (see problem 18, section 4.3, p. 265, in (Horn & Johnson, 1999)), which leads to

$$\left(I_{n}\otimes x^{T}\right)=\left(x^{T}\otimes I_{n}\right)P(n,n) \tag{44}$$

Therefore,

$$\begin{aligned} M_{D}^{T}(q,x)y &= \left(x^{T}\otimes I_{n}\right)P(n,n)\left(y^{T}\otimes\frac{\partial\Psi_{m}(q)}{\partial q}\right)\xi_{m} = \\ &= \left(x^{T}\otimes I_{n}\right)P(n,n)\psi(q,y)\xi_{m} \end{aligned} \tag{45}$$

Remark: The permutation matrix in the last lemma can be also written as

$$P(n,n)=\sum_{i=1}^{n}e_{i}\otimes I_{n}\otimes e_{i}^{T}=\sum_{i=1}^{n}e_{i}^{T}\otimes I_{n}\otimes e_{i} \tag{46}$$

For more details the reader is referred to the problem 21, section 4.3. p.286,in (Horn & Johnson, 1999).

3.3.4 Coriolis/Centripetal Matrix

On the basis of the description of M_{D} and M_{D}^{T} in LIP form, the skew-symmetric matrix J can be also represented as LIP.

Lemma 4: Let $x,y\in\mathbb{R}^{n}$, $q\in\Omega_{q}\subseteq\mathbb{R}^{n}$ arbitrary vectors with appropriate units. The skew-symmetric matrix $J(q,x)$ can be expressed is linear in the parameters,
$J(q,x)y=\Phi_{J}(q,x,y)\xi_{m}$
where $\Phi_{J}(q,x,y)=\left(y^{T}\otimes I_{n}\right)\left(I_{n^{2}}-P(n,n)\right)\psi(q,x)$.
Proof: It is straightforward that
$J(q,x)y=M_{D}(q,x)y-M_{D}^{T}(q,x)y=\left(y^{T}\otimes I_{n}\right)\left(I_{n^{2}}-P(n,n)\right)\psi(q,x)\xi_{m}$

Lemma 5: For arbitrary vectors $x,y\in\mathbb{R}^{n}$, $q\in\Omega_{q}\subseteq\mathbb{R}^{n}$, the inertia velocity matrix can be written as
$M_{v}(q,x)y=\Phi_{v}(q,x,y)\xi_{m}$

where $\Phi_v(q,x,y)=\sum_{i=1}^{n}\frac{\partial\Phi_m(q,y)}{\partial q_i}x_i$.

Proof: This is direct.

Remark: It is observed the following identity: $\Phi_D(q,x,y)=\Phi_v(q,y,x)$, which is consistent with the commutation property.

Remark: An alternative way for J is obtained by resorting to the commutation property:

$$
\begin{aligned}
M_v(q,x)y - M_D^T(q,x)y = M_D(q,y)x - M_D^T(q,y)x = \\
= \left(x^T \otimes I_n\right)\left(I_{n^2} - P(n,n)\right)\psi(q,y)\xi_m = J(q,y)x
\end{aligned}
$$

Remarkably, the above lemmas can be conveniently used to write the Coriolis/centripetal matrix in LIP form.

Proposition 1 (Coriolis/Centripetal matrix in LIP form): Let $x,y\in\mathbb{R}^n$, $q\in\Omega_q\subseteq\mathbb{R}^n$ arbitrary vectors, the Coriolis/centripetal effect $C(q,x)y$ can be linearly factorized as a regression matrix $\Phi_C(q,x,y)$ and a parameter vector ξ_m , i.e. $C(q,x)y=\Phi_C(q,x,y)\xi_m$, where Φ_C is given by

$$\Phi_C(q,x,y)=\frac{1}{2}\left[\left(x^T\otimes I_n\right)\psi(q,y)+\left(y^T\otimes I_n\right)\left(I_{n^2}-P(n,n)\right)\psi(q,x)\right]$$

Proof: By restoring to the LIP form of M_v and J , matrix C can be written as

$$
\begin{aligned}
C(q,x)y = \frac{1}{2}(M_v(q,x)y + J(q,x)y) = \\
= \frac{1}{2}\left[\left(x^T\otimes I_n\right)\psi(q,y)+\left(y^T\otimes I_n\right)\left(I_{n^2}-P(n,n)\right)\psi(q,x)\right]\xi_m
\end{aligned}
$$

3.4 Model Errors

The dynamic model of the robot manipulator is allowed to be imprecise since the nonlinear function $f(q,\dot{q},x,y)=M(q)x+C(q,\dot{q})y+G(q)$, (where $x,y\in\mathbb{R}^n$ are arbitrary vectors usually with units of acceleration and velocity respectively),is not exactly known. The imprecision comes from unstructured uncertainties, namely modeling errors caused by the truncation of the Gaussian expansion series. A detailed description of the approximation errors is demanding from a modeling viewpoint. To point out the fundamental aspects of error modeling, it is convenient to express the total error as composed by three terms, (Mulero-Martinez, 2007a),

$$E=E_m(q,x)+E_C(q,\dot{q},y)+E_G(q)$$

By referring to the expression of the Coriolis/centripetal matrix in (Mulero-Martínez,2007a) from the fundamental matrices M_D and M_V , the Coriolis/centripetal errors $E_C(q,\dot{q},y)$ can be written as

$$E_C(q,\dot{q},y)=\frac{1}{2}\left(E_D(q,\dot{q})-E_D^T(q,\dot{q})+E_V(q,\dot{q})\right)y$$

The error term $E_D(q,\dot{q})$ is the approximation error associated with the fundamental matrix $M_D(q,\dot{q})$ and $E_V(q,\dot{q})$ regards with the velocity matrix $M_V(q,\dot{q})$. For the sake of simplicity the gradient of the inertia error $\varepsilon_m(q)$ will be denoted as

$$\frac{\partial\varepsilon_m}{\partial q} \triangleq \left(\frac{\partial\varepsilon_m}{\partial q_1},\ldots,\frac{\partial\varepsilon_m}{\partial q_n}\right) \in \mathbb{R}^{n\times n^2}$$

These errors can be expressed in terms of the gradient of the inertia error $\varepsilon_m(q)$ as follows

$$\begin{aligned} E_D(q,\dot{q}) &= \sum_{i=1}^{n} \frac{\partial\varepsilon_m(q)}{\partial q_i}\dot{q}e_i^T \\ E_V(q,\dot{q}) &= \sum_{i=1}^{n} \frac{\partial\varepsilon_m(q)}{\partial q_i}\dot{q}_i \end{aligned} \tag{47}$$

For the following derivation, it is worth rewriting the mathematical errors in a more suitable form using the Kronecker product. This is written down in the following properties.

Claim 3: The linear transformation $E_D(q,\dot{q})\dot{q}_r$ can be formulated in terms of the Kronecker product as

$$E_D(q,\dot{q})\dot{q}_r = \frac{\partial\varepsilon_m(q)}{\partial q}(\dot{q}_r \otimes \dot{q}) \tag{48}$$

Proof: The proof is derived directly from the definition of $E_D(q,\dot{q})$

$$E_D(q,\dot{q})\dot{q}_r = \sum_{j=1}^{n} \frac{\partial\varepsilon_m(q)}{\partial q_j}\dot{q}e_j^T\dot{q}_r = \sum_{j=1}^{n} \frac{\partial\varepsilon_m(q)}{\partial q_j}\dot{q}\dot{q}_{r_j} = \frac{\partial\varepsilon_m(q)}{\partial q}(\dot{q}_r \otimes \dot{q})$$

Claim 4: The linear transformation $E_V(q,\dot{q})\dot{q}_r$ is expressed in terms of the Kronecker product as

$$E_V(q,\dot{q})\dot{q}_r = \frac{\partial\varepsilon_m(q)}{\partial q}(\dot{q} \otimes \dot{q}_r) \tag{49}$$

Proof: This fact can be trivially proved from the definition in equation (47)

$$E_V(q,\dot{q})\dot{q}_r = \sum_{i=1}^{n} \frac{\partial\varepsilon_m(q)}{\partial q_i}\dot{q}_i\dot{q}_r = \frac{\partial\varepsilon_m(q)}{\partial q}(\dot{q} \otimes \dot{q}_r)$$

Remark: Equations (48) and (49) are not the same since the Kronecker product is not commutative, i.e. $(\dot{q}_r \otimes \dot{q}) \neq (\dot{q} \otimes \dot{q}_r)$.

4. Design of the Adaptive Controller.

4.1 Error Dynamic Equation

In order to manage equilibrium points at the origin, it is necessary to make a coordinate transformation so that a position error variable is considered, $e(t) = q_d(t) - q(t)$. Thus, convergene of trajectory $q(t)$ to the desired trajectory $q_d(t)$ can be analysed observing position error trajectories $e(t)$ close to the origin in the phase space. The objective of the controller is both the stable tracking of trajectories and the rejection of disturbances. A good tracking performance means that the error converge to zero (asymptotic stability) or to a finite value, $\lim_{t\to\infty} e(t) = E < \infty$. This idea is also applied to $\dot{q}(t)$ and $\ddot{q}(t)$ because a position trajectory is given by three quantities $(q(t), \dot{q}(t), \ddot{q}(t))$. Measurements of velocities are easy to get by tachometers, but sensors of acceleration are noisy and are not used for implementation in robotics field. This fact conjures up to use a filtered error signal, $r(t)$, that is a derivative filter or PD (Slotine & Li, 1991):

$$r(t) = \dot{q}_r(t) - \dot{q}(t) \tag{50}$$

From (50) it is shown that $r(t)$ is a measurement of error between real velocity $\dot{q}(t)$ and reference velocity $\dot{q}_r(t)$. This reference velocity must not be confused with the desired velocity $\dot{q}_d(t)$. In fact, reference velocity is defined as follows

$$\dot{q}_r(t) = \dot{q}_d(t) + \Lambda e(t) \tag{51}$$

where Λ is a diagonal matrix of design parameters with big positive elements so that the system is BIBO stable. This matrix allows to filter errors so that no acceleration of errors $\ddot{e}(t)$ will appear in the error dynamic equation. The definition of filtered error $r(t)$ in terms of position and velocity errors can be obtained from (50) and (51)

$$r(t) = \dot{e}(t) + \Lambda e(t)$$

Substituting $q(t)$ from (50) into the plant, the error dynamic equation is derived.

$$M(q)\dot{r} = -C(q,\dot{q})r + f(x) - \tau + \tau_d$$

where $f(x) = M(q)\ddot{q}_r + C(q,\dot{q})\dot{q}_r + G(q) + F(q,\dot{q})$ stands for non-linear terms to be compensated. This non-linear function could be parametrised by $x^T = \left(q^T, \dot{q}^T, \dot{q}_r^T, \ddot{q}_r^T\right)^T$ or using the definition of $\dot{q}_r(t)$ in (51) by $x^T = \left(e^T, \dot{e}^T, q_d^T, \dot{q}_d^T, \ddot{q}_d^T\right)^T$.

A structural property of robot manipulators is the linearity of parameters (LIP) (Craig, 1989), (Sciavicco, 2002). This means that non-linearities can be split up into a parameter vector P and a vector of basis functions $\Psi(x)$. Therefore the non-linearity function $f(x)$ can be expressed in this sense adding a term of error ε.

$$f(x) = \Psi(q, \dot{q}, \dot{q}_r, \ddot{q}_r)P + \varepsilon$$

Linearity of parameters (LIP) is a first assumption in most of the adaptive controllers.

Proceeding in a similar way, the approximation of the non-linear function using estimation of the parameters $\hat{P}$ can be represented by means of linearity of parameters.

$$\hat{f}(x) = \Psi(q, \dot{q}, \dot{q}_r, \ddot{q}_r)\hat{P}$$

4.2 Controller Structure, Control Law and Updtating Law.

Up unto this point, LIP property has been analyzed via fundamental matrices. The control approach employs an *inertia-related linearization approach*, i.e. a conservation of energy formulation, as an attempt to derive update laws and control laws. To be specific, it is required to define an inertia-related Lyapunov function in the stability analysis which utilizes physical properties inherent to a mechanical manipulator (such as those presented above). Thus, the stability of the tracking error system is ensured by formulating the adaptive update rule and by analyzing the stability of the tracking error system at the same time. It is well known that dynamic models even though quite complex are anyhow an idealization of reality. Specifically, robots show uncertainties that are mainly found from two sources: variability of parameters and nonlinearity terms in the system. One way of dealing with parametric uncertainties would be to use the inertia-related approach, see (Lewis et al., 2003). The benefits of this approach as compared to others is that avoids a direct measurement of acceleration and the invertibility of the generalized inertia matrix, which are restrictions of some controllers such as those inspired in the adaptive computed-torque approaches.

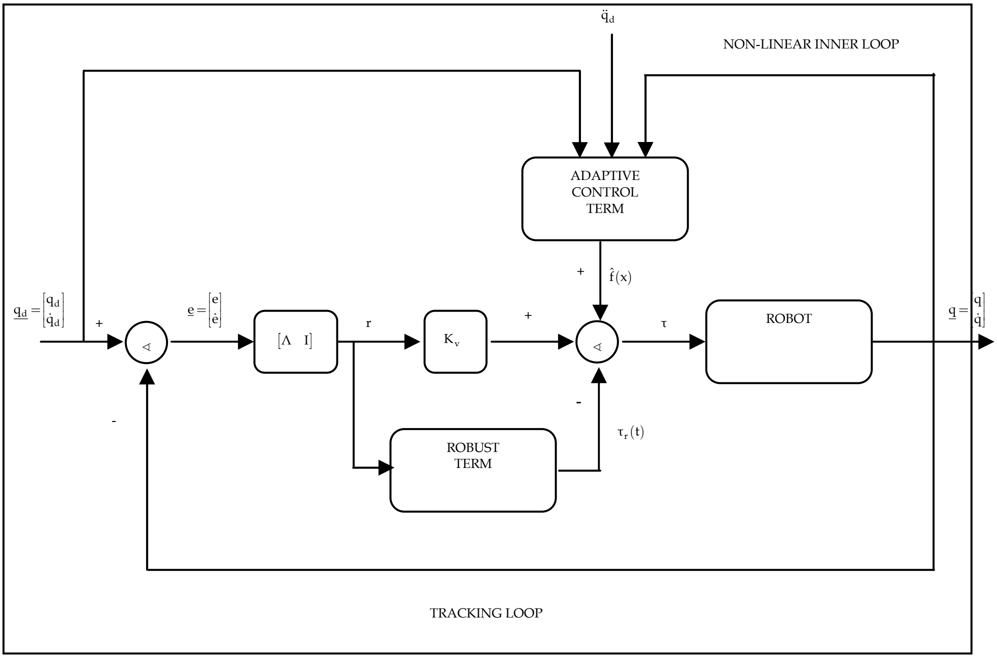
$\ddot{q}_d$
NON-LINEAR INNER LOOP
ADAPTIVE CONTROL TERM
+
$\hat{f}(x)$
$\underline{q}_d = \begin{bmatrix} q_d \\ \dot{q}_d \end{bmatrix}$
+
$\underline{e} = \begin{bmatrix} e \\ \dot{e} \end{bmatrix}$
$[\Lambda \quad I]$
r
K_v
+
τ
ROBOT
$\underline{q} = \begin{bmatrix} q \\ \dot{q} \end{bmatrix}$
-
-
$\tau_r(t)$
ROBUST TERM
TRACKING LOOP

Fig. 1. Adaptive Control Structure

The controller consists of three terms: a PD-controller to guarantee a good tracking performance, $\tau_{pd} = K_v r = K_v(\dot{e} + \Lambda e)$, a compensator of non-linearities, $\tau_{nl} = \hat{f}$ and a robust controller to absorb unmodelled dynamics τ_r .

$$\tau = K_v r + \hat{f} + \tau_r \tag{52}$$

where $K_v = K_v^T > 0$ is the gain matrix.

The control structure appears in figure 1. In this scheme, two loops can be spoted: an outer loop to track signals and an inner loop to compensate non-linearities. The inner loop is driven by an adaptive control and the outer loop is driven by a robust and PD terms.

An important feature of this class of controllers is that of being based on the all-important *closed-loop error dynamics*, which results from the substitution of the filtered tracking error into the robot dynamics

$$M(q)\dot{r} = -C(q,\dot{q})r + f(x) - \tau = -C(q,\dot{q})r - K_v r + \tilde{f}(x) - \tau_r \tag{53}$$

where $\tilde{f}(x) = f(x) - \hat{f}(x)$ stands for the functional estimation error with x being a vector of appropriate variables as shown below. The nominal nonlinearity f can be computed as

$f(q,\dot{q},\dot{q}_r,\ddot{q}_r) = M(q)\ddot{q}_r + C(q,\dot{q})\dot{q}_r + G(q)$

where $\ddot{q}_r$ is the reference acceleration, $\ddot{q}_r(t) \triangleq \ddot{q}_d(t) + \Lambda\dot{e}(t)$, and $\dot{q}_r$ the reference velocity, $\dot{q}_r(t) \triangleq \dot{q}_d(t) + \Lambda e(t)$. The nonlinearity f can be conveniently partitioned into several smaller terms, resulting into an added controller structure

$f(x) = f_m(x) + f_c(x) + f_g(x)$

where $f_g(x) = G(q)$. The terms f_m and f_c can be defined in terms of $\ddot{q}_r$ and $\dot{q}_r$ as follows

$$\begin{aligned} f_m(q,\ddot{q}_r) &= M(q)\ddot{q}_r = \Phi_m(q,\ddot{q}_r)\xi_m \\ f_c(q,\dot{q},\ddot{q}_r) &= C(q,\dot{q})\dot{q}_r = \Phi_c(q,\dot{q},\ddot{q}_r)\xi_m \\ f_g(q) &= G(q) = \Phi_g(q)\xi_g \end{aligned}$$

where $\xi_m \in \mathbb{R}^{l_m}$, $\xi_g \in \mathbb{R}^{l_g}$ are parameter vectors and Φ_m, Φ_c, Φ_g regression matrices.

Theorem 1: Let the desired trajectory $q_d(t)$ bounded by q_B , i.e. $\|q_d(t)\| \le q_B$. Suppose that the approximation error ε and unmodeled disturbances $\tau_d(t)$ are upper bounded by ε_N and d_B respectively. Let the control law given by (52) with a robust term, $\tau_r(t) = K_r \mathrm{sgn}(r)$ where $K_r = \mathrm{diag}(k_{r_{ii}}) > 0$ with $k_{r_{ii}} \ge \varepsilon_n + d_B$. The next parameter updating laws are considered