2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009

Pose Control of Robot Manipulators Using Different Orientation Representations: A Comparative Review

Ricardo Campa and Hussein de la Torre

Abstract— The pose of a rigid–body in 3-D space is described by a set of six independent variables, being three for position and three for orientation. In pose control tasks it is useful to define a pose error representing the deviation between the desired and actual pose of the body. Nevertheless, due to the peculiar properties of the orientation manifold, the orientation error is not well defined as a vector difference. This paper deals with some of those properties, and rewiews various definitions of the orientation error found in the literature. Then, some simulations are carried out on a robotic spherical wrist in order to compare the performance of each approach in a simple orientation control task.

I. INTRODUCTION

The term pose is employed in Mechanics to represent both the position and orientation of a body. It is well–known that the number of degrees of freedom required to define the pose of an object in a three-dimensional space is six: three for position and three for orientation.

Chasles' theorem [1] states that the position and orientation parts of the pose can be treated independently. Position is well described by a vector $\boldsymbol{p} \in \mathbb{R}^3$, usually in Cartesian coordinates. In the case of orientation, however, there is not a generalized method to describe it, and this is mainly due to the fact that the orientation manifold is not a vector space, but a Lie group.

Minimal representations of orientation are defined by three parameters, e.g., Euler angles. But in spite of their popularity, Euler angles suffer the drawbacks of representation singularities and inconsistency with the task geometry [2]. There are other nonminimal parameterizations of orientation which use a set of $3+k$ parameters, related by k holonomic constraints, in order to keep the required three degrees of freedom. Common examples of these are the rotation matrices, the angle-axis pair, and the Euler parameters.

The pose control problem consists on making the actual pose of a body reach the desired (possibly time–varying) pose. This problem has become more relevant in the recent years due to: (a) the use of new electronic devices for direct sensing of position/orientation signals; (b) the application of mathematical tools —such as unit quaternions and screws [3]—, which facilitate the analysis of the pose configuration space [4].

In pose control tasks, it is common to define a pose error which gives a measure of the deviation of the body's actual pose from the desired one. As position belongs to a vector space, the position error is merely the difference between the desired and actual position. But the definition of the orientation error is not so simple, and depends on the parameterization of orientation used.

Applications of pose modeling and control are mainly found in Robotics, where the desired motion of a moving body —such as a mobile vehicle or a manipulator's end–effector— is usually expressed in terms of its relative pose with respect to a fixed frame.

The main concern of this paper is to review the most common parameterizations of orientation and the different definitions of the orientation error. Then, in order to make a comparison among these, some simulations are carried out using a prototypical robotic spherical wrist and a simple orientation control task.

Throughout this paper bold letters indicate column vectors. The inner product of $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$ is given by $\boldsymbol{a}^T\boldsymbol{b} = \boldsymbol{b}^T\boldsymbol{a} \in \mathbb{R}$, where $\boldsymbol{a}^T$ represents the transpose of $\boldsymbol{a}$. The Euclidean norm of $\boldsymbol{a}$ is thus defined as $\|\boldsymbol{a}\| = \sqrt{\boldsymbol{a}^T\boldsymbol{a}}$.

II. THE POSE MANIFOLD OF A RIGID BODY

Every possible location of an object in space is known as a configuration. According to this, Samson et al [4] define the configuration space of a mechanical system as the set of all possible configurations of such a system.

As an example, a set of n free particles in a 3-dimensional space has a configuration space isomorphic to $\mathbb{R}^{3n}$; if those particles are restricted by k holonomic constraints, then the configuration space would be an r-dimensional manifold, $\mathrm{M}^r \subset \mathbb{R}^{3n}$, where $r = 3n - k$.

In general, given a manifold M^r and a point $\boldsymbol{z} \in \mathrm{M}^r$, $\boldsymbol{z}$ can be described either by using a minimal set of r local coordinates or a non-minimal set of $r+k$ parameters related by k constraints [5]. This is what happens when locating a point on the surface of a sphere —which is isomorphic to $\mathrm{S}^2 \subset \mathbb{R}^3$—; locally, any point on the sphere can be located using only two coordinates, however, for an observer which is out of the sphere, three parameters (e.g. Cartesian coordinates) are required.

Now let us consider the case of a rigid–body —i.e. an object in which any pair of particles in it is constrained to keep a constant distance—. It can be shown [1] that a rigid–body has a pose configuration space (manifold) of dimension 6. Let this configuration space be

$$\mathcal{P} \equiv \mathbb{R}^3 \times \mathrm{M}^3 \subseteq \mathbb{R}^{3+m} \tag{1}$$

Work supported by DGEST and CONACyT (grant 60230).

The authors are with the Postgraduate and Research Division of the Instituto Tecnológico de la Laguna, Apdo. Postal 516, 27000, Torreón, Coah., Mexico. recampa@itlalaguna.edu.mx

978-1-4244-4524-0/09/$25.00 ©2009 AACC

where $\mathbb{R}^3$ corresponds to the position part and $\mathrm{M}^3 \subseteq \mathbb{R}^m$ to the orientation part of the pose; $m \geq 3$ is the number of parameters used to describe the orientation.

In order to specify the pose of a rigid–body let us define an inertial coordinate frame Σ_o fixed in space. Also, we define a frame Σ_b which is bound to the rigid–body —i.e. this frame moves together with the body—.

Let $\boldsymbol{p} \in \mathbb{R}^3$ be the position vector describing the relative position of the origin of Σ_b with respect to Σ_o. Even if the orientation does not belong to a vector space, it is convenient to define an "orientation vector" $\boldsymbol{\phi} \in \mathrm{M}^3 \subseteq \mathbb{R}^m$ The pose of the body is then given by

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{\phi} \end{bmatrix} \in \mathcal{P}. \tag{2}$$

Among the different parameterizations of the orientation manifold, the four most relevant in orientation control applications are:

1) Euler angles: $\mathrm{M}^3 \equiv \mathbb{R}^3$.
2) Rotation matrix: $\mathrm{M}^3 \equiv \mathrm{SO}(3) \subset \mathbb{R}^{3\times 3}$.
3) Angle–axis pair: $\mathrm{M}^3 \equiv \mathbb{R}\times \mathrm{S}^2 \subset \mathbb{R}^4$.
4) Euler parameters: $\mathrm{M}^3 \equiv \mathrm{S}^3 \subset \mathbb{R}^4$.

These parameterizations are explained in the following section. Of particular interest are cases 2 and 4, because they happen to form Lie groups.

The special orthogonal group $\mathrm{SO}(3) \subset \mathbb{R}^{3\times 3}$ is defined

$$\mathrm{SO}(3) = \{R \in \mathbb{R}^{3\times 3} : R^T R = I, \det(R) = 1\}, \tag{3}$$

while the hypersphere of dimension three $\mathrm{S}^3 \subset \mathbb{R}^4$ is

$$\mathrm{S}^3 = \{\boldsymbol{x} \in \mathbb{R}^4 : \|\boldsymbol{x}\| = 1\}. \tag{4}$$

III. PARAMETERIZATIONS OF ORIENTATION

A. Euler angles

Leonard Euler (1707–1783) first stablished and proved that any two independent orthonormal coordinate frames with a common origin can be related by a sequence of no more than three rotations around the coordinate axes. That means that if the secuence of axes to rotate is known, we only need three Euler angles to completely define the whole rotation. Even though it is not a standard notation, here we denote the three Euler angles as α, β and γ, which can grouped in a vector

$$\boldsymbol{\phi}(\alpha,\beta,\gamma) = \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} \in \mathbb{R}^3 \equiv \mathrm{M}^3 \tag{5}$$

becoming a minimal parameterization of orientation.

Given α, β, γ, and the sequence of rotations, the corresponding orientation is well defined, however, the inverse problem is not. In fact, it is well–known that every sequence of Euler angles has inherently singular points [6], i.e., there is always a particular set of orientations for which the selected set of Euler angles is not uniquely defined. This is due to the fact that Euler angles are local coordinates of the orientation manifold, and they cannot "cover" the whole manifold.

In spite of this drawback, the use of Euler angles for representing orientation is still very common, mainly because of the simplicity of using a minimal vector such as (5). In Robotics, defining the pose of a manipulator by means of a vector $\boldsymbol{x} \in \mathbb{R}^6$ is the base of what is called the operational space approach [7].

B. Rotation matrix

Rotation matrices are perhaps the most extended method for describing orientation, mainly due to the convenience of matrix algebra operations. Let us consider again the fixed Σ_o and moving Σ_b coordinate frames. As only orientation is of concern, let us assume that the origins of both frames coincide. The rotation matrix of Σ_b with respect to Σ_o is named $R \in \mathrm{SO}(3)$.

Now let $\boldsymbol{r}_i$ be the i-th column of matrix R. Bach and Paielli [8] proposed an interesting representation of rotation matrices, in which the three columns of a rotation matrix are stacked in a column vector, i.e.

$$\boldsymbol{\phi}(R) = \begin{bmatrix} \boldsymbol{r}_1 \\ \boldsymbol{r}_2 \\ \boldsymbol{r}_3 \end{bmatrix} \in \mathrm{M}^3 \subset \mathbb{R}^9. \tag{6}$$

Vector $\boldsymbol{\phi}(R)$ is more adecuate for control applications, since it allows to handle the whole pose manifold $\mathcal{P}$ as a vector (see equation (2)).

C. Angle–axis pair

Euler also showed that the general displacement of a rigid–body with a fixed point is a rotation around some axis [3]. This is known as the Euler's rotation theorem, and implies that an arbitrary rotation between two coordinate frames is equivalent to a single rotation around an axis. Thus, to completely define a rotation (or relative orientation between two frames) we need an angle $\theta \in \mathbb{R}$ and a unit vector $\boldsymbol{u} \in \mathrm{S}^2 \subset \mathbb{R}^3$ in the direction of the rotation axis.

Hence, the so–called angle–axis pair uses four parameters and a unit norm constraint ($\|\boldsymbol{u}\| = 1$) to define a rotation. Expressed as a vector

$$\boldsymbol{\phi}(\theta,\boldsymbol{u}) = \begin{bmatrix} \theta \\ \boldsymbol{u} \end{bmatrix} \in \mathrm{M}^3 \equiv \mathbb{R} \times \mathrm{S}^2 \subset \mathbb{R}^4. \tag{7}$$

Given the angle θ and the rotation axis $\boldsymbol{u}$, the corresponding orientation is well defined, and can be expressed as a rotation matrix $R(\theta,\boldsymbol{u})$ given by [3]:

$$R(\theta,\boldsymbol{u}) = \cos(\theta)I + \sin(\theta)S(\boldsymbol{u}) + [1-\cos(\theta)]\boldsymbol{u}\boldsymbol{u}^T \tag{8}$$

where, for a vector $\boldsymbol{v} = [\,v_1 \quad v_2 \quad v_3\,]^T \in \mathbb{R}^3$, the matrix operator $S(\boldsymbol{v})$ is given by

$$S(\boldsymbol{v}) = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix} \in \mathrm{so}(3); \tag{9}$$

so(3) is the space of skew–symmetric 3×3 matrices.

Given two vectors $\boldsymbol{v}, \boldsymbol{w} \in \mathbb{R}^3$, the following useful properties of the skew–symmetric operator stand:

$$\begin{aligned} S(-\boldsymbol{v}) &= -S(\boldsymbol{v}), & (10)\\ S(\boldsymbol{v})\boldsymbol{w} &= -S(\boldsymbol{w})\boldsymbol{v}, & (11)\\ S(\boldsymbol{v})S(\boldsymbol{v}) &= \boldsymbol{v}\boldsymbol{v}^T - \boldsymbol{v}^T\boldsymbol{v}I. & (12)\end{aligned}$$

From (8), and considering (10), it should be noticed that $R(-\theta, -\boldsymbol{u}) = R(\theta, \boldsymbol{u})$, so that, for a given orientation, represented by R, the angle–axis pair is not unique, but double, whenever $\theta \neq 2n\pi$. In fact, the angle–axis pair is what is called a double cover of the rotation group SO(3) [5]. The inverse mapping $\mathrm{SO}(3) \to \mathbb{R} \times \mathrm{S}^2$ is not well defined when $\theta = 2n\pi$. In such a case $R = I$ (both frames coincide), and $\boldsymbol{u}$ can be chosen arbitrarily.

D. Euler parameters

Another way of describing orientation is by means of the so–called Euler parameters, denoted here as $\eta \in \mathbb{R}$ and $\boldsymbol{\varepsilon} \in \mathbb{R}^3$. Those parameters satisfy a unit norm condition given by

$$\eta^2 + \boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon} = 1, \qquad (13)$$

so that they can be seen as unit quaternions which belong to the unit hypersphere S^3. Thus, the orientation vector $\boldsymbol{\phi}$ using Euler parameters is given by

$$\boldsymbol{\phi}(\eta, \boldsymbol{\varepsilon}) = \begin{bmatrix} \eta \\ \boldsymbol{\varepsilon} \end{bmatrix} \in \mathrm{M}^3 \equiv \mathrm{S}^3 \subset \mathbb{R}^4. \qquad (14)$$

Euler parameters are closely related to the angle–axis pair, and can be computed directly from them using

$$\eta = \cos\left(\frac{\theta}{2}\right), \quad \boldsymbol{\varepsilon} = \sin\left(\frac{\theta}{2}\right)\boldsymbol{u}. \qquad (15)$$

For a given η and $\boldsymbol{\varepsilon}$ the corresponding orientation is defined by means of a rotation matrix as follows [10]:

$$R(\eta, \boldsymbol{\varepsilon}) = (\eta^2 - \boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon})I + 2\eta S(\boldsymbol{\varepsilon}) + 2\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^T,$$

with the skew–symmetric operator $S(\boldsymbol{\varepsilon})$ defined as in (9). Note that $R(\eta, \boldsymbol{\varepsilon}) = R(-\eta, -\boldsymbol{\varepsilon})$, meaning that S^3 is also a double cover of SO(3). But, unlike the angle–axis pair, Euler parameters truly give a global (though double) parameterization of orientation.

A more detailed description of Euler parameters and their applications to Robotics can be found in [9].

IV. LINEAR AND ANGULAR VELOCITIES

In a pure translational motion only a linear velocity vector is present, indicating the speed and direction of the displacement. In a pure rotational motion there is an angular velocity vector, whose magnitud and direction indicate, respectively, the rate of change of the angular displacement and the instantaneous axis of rotation.

The linear velocity of the body, denoted by $\boldsymbol{v} \in \mathbb{R}^3$, is simply the time derivative of the position vector, that is $\boldsymbol{v} = \dot{\boldsymbol{p}}$. However, the relation between the angular velocity $\boldsymbol{\omega} \in \mathbb{R}^3$ and the time derivative of the orientation vector $\dot{\boldsymbol{\phi}}$ is not so straight, because it depends on the current orientation of the body. In general we have

$$\boldsymbol{\omega} = J_\phi(\boldsymbol{\phi})\dot{\boldsymbol{\phi}} \qquad (16)$$

where $J_\phi \in \mathbb{R}^{3\times m}$ is called the representation Jacobian, which depends on the parameterization employed for describing orientation. Those orientations $\boldsymbol{\phi}$ in which J_ϕ loses rank are known as representation singularities [10].

A. Jacobian for Euler angles, $J_\phi(\alpha, \beta, \gamma)$

Suppose that the orientation of a moving rigid–body is described by means of the Euler angles $\alpha(t)$, $\beta(t)$ and $\gamma(t)$, around some general frame unit axes $\boldsymbol{w}_\alpha$, $\boldsymbol{w}_\beta$, and $\boldsymbol{w}_\gamma$, respectively, which depend of the convention (or sequence of rotations) chosen for the Euler angles.

The angular velocity of the moving body can be decomposed as follows [10]:

$$\boldsymbol{\omega} = \dot{\alpha}\boldsymbol{w}_\alpha + \dot{\beta}\boldsymbol{w}_\beta + \dot{\gamma}\boldsymbol{w}_\gamma = [\,\boldsymbol{w}_\alpha \;\; \boldsymbol{w}_\beta \;\; \boldsymbol{w}_\gamma\,]\begin{bmatrix}\dot{\alpha}\\ \dot{\beta}\\ \dot{\gamma}\end{bmatrix}$$

so that

$$J_\phi(\alpha, \beta, \gamma) = [\,\boldsymbol{w}_\alpha \;\; \boldsymbol{w}_\beta \;\; \boldsymbol{w}_\gamma\,] \in \mathbb{R}^{3\times 3}. \qquad (17)$$

As an example, consider the representation Jacobian for the common ZYZ-convention Euler angles, which is

$$J_\phi(\alpha, \beta, \gamma) = \begin{bmatrix} 0 & -S_\alpha & S_\beta C_\alpha \\ 0 & C_\alpha & S_\beta S_\alpha \\ 1 & 0 & C_\beta \end{bmatrix}.$$

Notice that J_ϕ is singular whenever $\beta = n\pi$. As mentioned before, all of the Euler angles conventions have representation singularities.

B. Jacobian for rotation matrix, $J_\phi(R)$

If the orientation is described by means of $R \in \mathrm{SO}(3)$, then the relation between $\boldsymbol{\omega}$ and $\dot{R}$ is given by [10]:

$$\dot{R} = S(\boldsymbol{\omega})R \qquad (18)$$

where $S(\cdot)$ is the matrix operator defined in (9). Notice that (18) can be rewritten as

$$\begin{bmatrix}\dot{\boldsymbol{r}}_1\\ \dot{\boldsymbol{r}}_2\\ \dot{\boldsymbol{r}}_3\end{bmatrix} = S(\boldsymbol{\omega})\begin{bmatrix}\boldsymbol{r}_1\\ \boldsymbol{r}_2\\ \boldsymbol{r}_3\end{bmatrix} = -\begin{bmatrix}S(\boldsymbol{r}_1)\\ S(\boldsymbol{r}_2)\\ S(\boldsymbol{r}_3)\end{bmatrix}\boldsymbol{\omega} \qquad (19)$$

where property (11) has been used. Also, notice that

$$[\,S(\boldsymbol{r}_1) \quad S(\boldsymbol{r}_2) \quad S(\boldsymbol{r}_3)\,]\begin{bmatrix}S(\boldsymbol{r}_1)\\ S(\boldsymbol{r}_2)\\ S(\boldsymbol{r}_3)\end{bmatrix} = -2I \in \mathbb{R}^{3\times 3}$$

so it is possible to resolve $\boldsymbol{\omega}$ from (19) to get

$$\boldsymbol{\omega} = \frac{1}{2}[\,S(\boldsymbol{r}_1) \quad S(\boldsymbol{r}_2) \quad S(\boldsymbol{r}_3)\,]\begin{bmatrix}\dot{\boldsymbol{r}}_1\\ \dot{\boldsymbol{r}}_2\\ \dot{\boldsymbol{r}}_3\end{bmatrix}.$$

Considering the orientation vector $\boldsymbol{\phi}(R)$ defined in (6), it is clear that the representation Jacobian (16) is

$$J_\phi(R) = \frac{1}{2}[\,S(\boldsymbol{r}_1) \quad S(\boldsymbol{r}_2) \quad S(\boldsymbol{r}_3)\,] \in \mathbb{R}^{3\times 9}. \qquad (20)$$

Note that $J_\phi(R)$ has no representation singularities.

C. Jacobian for Euler parameters, $J_\phi(\eta, \boldsymbol{\varepsilon})$

The derivative of Euler parameters is given by the so–called quaternion propagation rule [10]:

$$\begin{bmatrix} \dot{\eta} \\ \dot{\boldsymbol{\varepsilon}} \end{bmatrix} = \frac{1}{2} E(\eta, \boldsymbol{\varepsilon}) \boldsymbol{\omega}, \tag{21}$$

where $\boldsymbol{\omega}$ is the angular velocity, and

$$E(\eta, \boldsymbol{\varepsilon}) = \begin{bmatrix} -\boldsymbol{\varepsilon}^T \\ \eta I - S(\boldsymbol{\varepsilon}) \end{bmatrix} \in \mathbb{R}^{4\times 3}. \tag{22}$$

By using property (12) and the unit norm constraint (13) it is easy to prove that $E(\eta,\boldsymbol{\varepsilon})^T E(\eta,\boldsymbol{\varepsilon}) = I$, so that $\boldsymbol{\omega}$ can be resolved from (21) as

$$\boldsymbol{\omega} = 2E(\eta, \boldsymbol{\varepsilon})^T \begin{bmatrix} \dot{\eta} \\ \dot{\boldsymbol{\varepsilon}} \end{bmatrix} \tag{23}$$

indicating that the representation Jacobian is

$$J_\phi(\eta, \boldsymbol{\varepsilon}) = 2E(\eta, \boldsymbol{\varepsilon})^T = 2\left[\, -\boldsymbol{\varepsilon} \quad \eta I + S(\boldsymbol{\varepsilon}) \,\right], \tag{24}$$

which is full rank for all $[\,\eta \quad \boldsymbol{\varepsilon}^T\,]^T \in \mathrm{S}^3$.

D. Jacobian for angle–axis pair, $J_\phi(\theta, \boldsymbol{u})$

By taking the time derivative of the Euler parameters given by (15), and comparing with (21), we get

$$\begin{bmatrix} \dot{\theta} \\ \dot{\boldsymbol{u}} \end{bmatrix} = F(\theta, \boldsymbol{u}) \boldsymbol{\omega} \tag{25}$$

where $F(\theta, \boldsymbol{u}) \in \mathbb{R}^{4\times 3}$ is given by

$$F(\theta, \boldsymbol{u}) = \begin{bmatrix} \boldsymbol{u}^T \\ -\frac{1}{2\sin\left(\frac{\theta}{2}\right)} \left[\sin\left(\frac{\theta}{2}\right) I + \cos\left(\frac{\theta}{2}\right) S(\boldsymbol{u})\right] S(\boldsymbol{u}) \end{bmatrix}.$$

By using the properties of the $S(\cdot)$ operator, and the unit norm constraint of $\boldsymbol{u}$ it is possible to show that $J_\phi(\theta, \boldsymbol{u}) F(\theta, \boldsymbol{u}) = I$, where $J_\phi(\theta, \boldsymbol{u}) \in \mathbb{R}^{3\times 4}$ is

$$J_\phi(\theta, \boldsymbol{u}) = \left[\, \boldsymbol{u} \quad \sin(\theta)\left[I - \boldsymbol{u}\boldsymbol{u}^T\right] + [1-\cos(\theta)] S(\boldsymbol{u}) \,\right] \tag{26}$$

so that, resolving $\boldsymbol{\omega}$ from (25), we finally get

$$\boldsymbol{\omega} = J_\phi(\theta, \boldsymbol{u}) \begin{bmatrix} \dot{\theta} \\ \dot{\boldsymbol{u}} \end{bmatrix}.$$

Notice that $J_\phi(\theta, \boldsymbol{u})$ is singular when $\theta = 2n\pi$.

V. POSE ERROR FOR CONTROL TASKS

Given the actual pose $\boldsymbol{x} = [\,\boldsymbol{p} \quad \boldsymbol{\phi}\,]^T \in \mathcal{P}$ and the desired pose $\boldsymbol{x}_d = [\,\boldsymbol{p}_d \quad \boldsymbol{\phi}_d\,]^T \in \mathcal{P}$ of a rigid–body, the pose control aim is

$$\lim_{t\to\infty} \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{\phi} \end{bmatrix} = \begin{bmatrix} \boldsymbol{p}_d \\ \boldsymbol{\phi}_d \end{bmatrix}.$$

In control applications it is common to define an error variable, which indicates the deviation between the variable to be controlled and its desired value. Then, this error is used by the controller for measuring the difference between those two values (a null error means that the variable has reached its desired value).

The position error vector, $\widetilde{\boldsymbol{p}}$ is simply defined as $\widetilde{\boldsymbol{p}} = \boldsymbol{p}_d - \boldsymbol{p}$. But in the case of orientation an inherent difficulty arises, due to the fact that the orientation error should be defined in terms of the algebra of the rotation group and not of the vector algebra.

Let us consider the frames shown in Figure 1. As always, Σ_o and Σ_b are the inertial and rigid–body frames. The desired pose can be expressed as an additional frame Σ_d. Vectors $\boldsymbol{\phi}$ and $\boldsymbol{\phi}_d$ represent the orientation of Σ_b and Σ_d with respect to the inertial frame. The error vector, however, can be defined in two ways, according to which coordinate frame is used as reference.

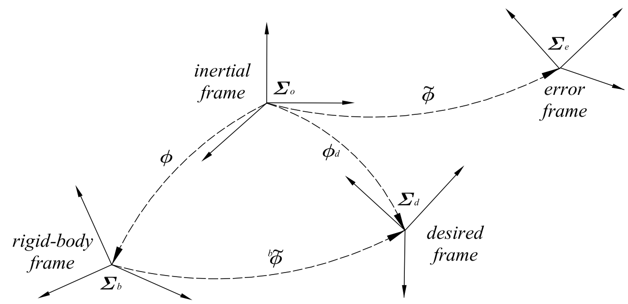

Fig. 1. Coordinate frames for orientation control.

The relative orientation of Σ_d with respect to Σ_b, is called ${}^b\widetilde{\boldsymbol{\phi}}$. This orientation error is generally used in pose control applications for autonomous vehicles, where the variables are generally referred to the rigid–body frame. However, for the cases where it is required to have the orientation error referred to the inertial frame, we use vector $\widetilde{\boldsymbol{\phi}}$. It can be shown [6] that ${}^b\widetilde{\boldsymbol{\phi}}$ and $\widetilde{\boldsymbol{\phi}}$ actually represent the same relative orientation, but referred to different frames. In the rest of the paper we consider only the case where the orientation error is given by $\widetilde{\boldsymbol{\phi}}$.

In the following subsections we present some orientation error definitions for the four parameterizations of orientation given in Section III.

A. Orientation error for Euler angles, $\widetilde{\boldsymbol{\phi}}(\widetilde{\alpha}, \widetilde{\beta}, \widetilde{\gamma})$

In the operational space approach [7] Euler angles are used as coordinates of proper vectors in $\mathbb{R}^3$. Thus, if $[\,\alpha \quad \beta \quad \gamma\,]^T$ and $[\,\alpha_d \quad \beta_d \quad \gamma_d\,]^T$ represent the orientation of the actual and desired frames of the body, then the orientation error vector $\widetilde{\boldsymbol{\phi}}(\widetilde{\alpha}, \widetilde{\beta}, \widetilde{\gamma})$ is simply

$$\widetilde{\boldsymbol{\phi}}(\widetilde{\alpha}, \widetilde{\beta}, \widetilde{\gamma}) = \begin{bmatrix} \widetilde{\alpha} \\ \widetilde{\beta} \\ \widetilde{\gamma} \end{bmatrix} = \begin{bmatrix} \alpha_d - \alpha \\ \beta_d - \beta \\ \gamma_d - \gamma \end{bmatrix} \in \mathbb{R}^3. \tag{27}$$

It should be kept in mind, however, that a minimal representation such as that given by Euler angles, is valid only in a local sense.

B. Orientation error for rotation matrix, $\widetilde{\boldsymbol{\phi}}(\widetilde{R})$

Let us considere that $R, R_d \in \mathrm{SO}(3)$ describe the actual and desired orientation of a rigid body with respect of the inertial frame, as in Fig. 1, then the error rotation matrix, expressing the relative orientation from Σ_d to Σ_b, with respect to Σ_o, is given by [10]:

$$\widetilde{R} = R_d R^T \in \mathrm{SO}(3). \tag{28}$$

However, the use of $\widetilde{R}$ in orientation control tasks is very limited due to the difficulty of handling the nine elements of the matrix. It is far better to use an orientation error vector, in terms of $\widetilde{R}$.

For the purpose of this paper we use the following definition, first proposed by Luh et al [11]:

$$\widetilde{\boldsymbol{\phi}}(\widetilde{R}) = \frac{1}{2}\begin{bmatrix} \widetilde{r}_{32} - \widetilde{r}_{23} \\ \widetilde{r}_{13} - \widetilde{r}_{31} \\ \widetilde{r}_{21} - \widetilde{r}_{12} \end{bmatrix} \in \mathrm{I\!R}^3, \tag{29}$$

where $\widetilde{r}_{ij}$ is the ij–th element of $\widetilde{R}$. Moreover, it is possible to show [2] that (29) can be written as

$$\widetilde{\boldsymbol{\phi}}(\widetilde{R}) = \frac{1}{2}\left[\, S(\boldsymbol{r}_{d1})^T \quad S(\boldsymbol{r}_{d2})^T \quad S(\boldsymbol{r}_{d3})^T \,\right]\begin{bmatrix} \boldsymbol{r}_1 \\ \boldsymbol{r}_2 \\ \boldsymbol{r}_3 \end{bmatrix} \tag{30}$$

where $\boldsymbol{r}_{di}$ and $\boldsymbol{r}_i$ stand for the i–th column of R_d and R, respectively. Notice that $\widetilde{\boldsymbol{\phi}}(\widetilde{R}) = 0$ whenever $R = R_d$.

C. Orientation error for Euler parameters, $\widetilde{\boldsymbol{\phi}}(\widetilde{\eta}, \widetilde{\boldsymbol{\varepsilon}})$

Let $[\,\eta \quad \boldsymbol{\varepsilon}^T\,]^T$ and $[\,\eta_d \quad \boldsymbol{\varepsilon}_d^T\,]^T$ be the Euler parameters of the actual and desired orientation of the rigid body, respectively. It is possible to show that the Euler parameters corresponding to $\widetilde{R}$ in (28) are given by [10]

$$\begin{bmatrix} \widetilde{\eta} \\ \widetilde{\boldsymbol{\varepsilon}} \end{bmatrix} = \begin{bmatrix} \eta\eta_d + \boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon}_d \\ \eta\boldsymbol{\varepsilon}_d - \eta_d\boldsymbol{\varepsilon} + S(\boldsymbol{\varepsilon})\boldsymbol{\varepsilon}_d \end{bmatrix} \in \mathrm{S}^3, \tag{31}$$

When using Euler parameters, it is common to choose the error vector $\widetilde{\boldsymbol{\phi}}(\widetilde{\eta}, \widetilde{\boldsymbol{\varepsilon}})$ as the vector part of the quaternion, i.e.

$$\widetilde{\boldsymbol{\phi}}(\widetilde{\eta}, \widetilde{\boldsymbol{\varepsilon}}) = \widetilde{\boldsymbol{\varepsilon}} \tag{32}$$

Notice that $\widetilde{\boldsymbol{\phi}} = \mathbf{0}$ when the actual and desired orientations coincide (i.e. when $[\,\eta \quad \boldsymbol{\varepsilon}^T\,]^T = \pm[\,\eta_d \quad \boldsymbol{\varepsilon}_d^T\,]^T$).

D. Orientation error for angle–axis pair, $\widetilde{\boldsymbol{\phi}}(\widetilde{\theta}, \widetilde{\boldsymbol{u}})$

Let $[\,\theta \quad \boldsymbol{u}^T\,]^T$ and $[\,\theta_d \quad \boldsymbol{u}_d^T\,]^T$ be the angle–axis pairs representing the actual and desired orientation of the body, respectively, and $[\,\widetilde{\theta} \quad \widetilde{\boldsymbol{u}}^T\,]^T$ be the corresponding angle–axis pair for the orientation error, which satisfy

$$\widetilde{\eta} = \cos\left(\tfrac{\widetilde{\theta}}{2}\right), \quad \widetilde{\boldsymbol{\varepsilon}} = \sin\left(\tfrac{\widetilde{\theta}}{2}\right)\widetilde{\boldsymbol{u}}. \tag{33}$$

By equating (33) and (31) we are able to obtain:

$$\widetilde{\theta} = 2\arccos\left(\mathrm{c}\left(\tfrac{\theta}{2}\right)\mathrm{c}\left(\tfrac{\theta_d}{2}\right) + \mathrm{s}\left(\tfrac{\theta}{2}\right)\mathrm{s}\left(\tfrac{\theta_d}{2}\right)\boldsymbol{u}^T\boldsymbol{u}_d\right)$$

$$\widetilde{\boldsymbol{u}} = \frac{\mathrm{c}\left(\frac{\theta}{2}\right)\mathrm{s}\left(\frac{\theta_d}{2}\right)\boldsymbol{u}_d - \mathrm{c}\left(\frac{\theta_d}{2}\right)\mathrm{s}\left(\frac{\theta}{2}\right)\boldsymbol{u} + \mathrm{s}\left(\frac{\theta}{2}\right)\mathrm{s}\left(\frac{\theta_d}{2}\right)S(\boldsymbol{u})\boldsymbol{u}_d}{\mathrm{s}\left(\frac{\widetilde{\theta}}{2}\right)}$$

where $\mathrm{s}(\cdot)$, $\mathrm{c}(\cdot)$ stand for $\sin(\cdot)$, $\cos(\cdot)$, respectively.

The orientation error vector $\widetilde{\boldsymbol{\phi}}(\widetilde{\theta}, \widetilde{\boldsymbol{u}}) \in \mathrm{I\!R}^3$ can take the general form $\widetilde{\boldsymbol{\phi}}(\widetilde{\theta}, \widetilde{\boldsymbol{u}}) = f(\widetilde{\theta})\widetilde{\boldsymbol{u}}$, where $f(\widetilde{\theta})$ is a scalar continuous function such that $f(0) = 0$. Several cases can be found in the literature. For the purpose of this paper, let us take $f(\widetilde{\theta}) = \widetilde{\theta}$ so that

$$\widetilde{\boldsymbol{\phi}}(\widetilde{\theta}, \widetilde{\boldsymbol{u}}) = \widetilde{\theta}\widetilde{\boldsymbol{u}} \tag{34}$$

VI. APPLICATION TO ROBOTICS

In order to validate the previous analysis on the definition of the orientation error in a pose control application, we carried out some Matlab simulations. We chose to use the model of a real 3–dof spherical wrist built at the Robotics Lab of CICESE Research Center [12].

A. Robot modeling

Let $\boldsymbol{q} = [\,q_1 \quad q_2 \quad \ldots \quad q_n\,]^T \in \mathrm{I\!R}^n$ be the vector of joint variables of a robot manipulator with n degrees of freedom, and $\boldsymbol{x} = [\,\boldsymbol{p}^T \quad \boldsymbol{\phi}^T\,]^T \in \mathcal{P}$ be the pose of its end–effector. The relation between the joint and pose variables is given by the forward kinematics function:

$$\boldsymbol{x} = \boldsymbol{h}(\boldsymbol{q}) \tag{35}$$

By taking the time derivative of (35) we get

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{p}} \\ \dot{\boldsymbol{\phi}} \end{bmatrix} = \frac{\partial h(\boldsymbol{q})}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}} = J_a(\boldsymbol{q})\dot{\boldsymbol{q}}$$

where $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$ are, respectively, the vectors of joint and pose velocities; $J_a(\boldsymbol{q}) \in \mathrm{I\!R}^{(3+m)\times n}$ is known as the analytic Jacobian of the manipulator.

On the other hand, the relation between the joint velocities, $\dot{\boldsymbol{q}} \in \mathrm{I\!R}^n$ and the linear and angular velocities of the end–effector is given by [10]:

$$\begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{\omega} \end{bmatrix} = J(\boldsymbol{q})\dot{\boldsymbol{q}}$$

where $J(\boldsymbol{q})$ is the so–called manipulator geometric Jacobian. Notice that $J(\boldsymbol{q})$ can be computed from

$$J(\boldsymbol{q}) = \begin{bmatrix} I & 0 \\ 0 & J_\phi(\boldsymbol{\phi}) \end{bmatrix} J_a(\boldsymbol{q})$$

where $J_\phi(\boldsymbol{\phi})$ is the representation Jacobian, which, depending on the chosen parameterization, can be either (17), (20), (26), or (24). It is worth noticing that $J(\boldsymbol{q})$ is independent of the orientation parameterization.

For the case of the spherical wrist used for the simulations, as $n = 3$ and only orientation is of concern, we have $\boldsymbol{x} = \boldsymbol{\phi}$, which in the case of using Euler parameters becomes [13]:

$$\boldsymbol{\phi}(\eta, \boldsymbol{\varepsilon}) = \begin{bmatrix} \eta(\boldsymbol{q}) \\ \varepsilon_1(\boldsymbol{q}) \\ \varepsilon_2(\boldsymbol{q}) \\ \varepsilon_3(\boldsymbol{q}) \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{q_2}{2}\right)\cos\left(\frac{q_1+q_3}{2}\right) \\ -\sin\left(\frac{q_2}{2}\right)\sin\left(\frac{q_1-q_3}{2}\right) \\ \sin\left(\frac{q_2}{2}\right)\cos\left(\frac{q_1-q_3}{2}\right) \\ \cos\left(\frac{q_2}{2}\right)\sin\left(\frac{q_1+q_3}{2}\right) \end{bmatrix}$$

In dynamic control it is also required the dynamic model of the manipulator, given by [14]:

$$M(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} \tag{36}$$

where $M(\boldsymbol{q}) \in \mathrm{I\!R}^{n\times n}$ is the inertia matrix, $C(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathrm{I\!R}^{n\times n}$ is the matrix of centrifugal and Coriolis terms, and $\boldsymbol{g}(\boldsymbol{q}) \in \mathrm{I\!R}^n$ is the vector of gravitational forces; $\boldsymbol{\tau} \in \mathrm{I\!R}^3$ is the vector of external forces applied to the robot joints.

The entries of the wrist dynamics (36) are not included for reasons of space, but the can be found in [12]:

B. Pose control and simulation results

To test and compare the performance of a pose controller using different descriptions of the orientation error, we chose a simple orientation regulation task in which the wrist is intended to reach a desired fixed orientation, then we applied a pose controller based in the inverse dynamics methodology [14].

Four simulations were carried out in Matlab, each with one of the orientation error vector definitions presented in Section V, that is, $\widetilde{\boldsymbol{\phi}}(\widetilde{\alpha},\widetilde{\beta},\widetilde{\gamma})$, $\widetilde{\boldsymbol{\phi}}(\widetilde{R})$, $\widetilde{\boldsymbol{\phi}}(\widetilde{\eta},\widetilde{\boldsymbol{\varepsilon}})$, $\widetilde{\boldsymbol{\phi}}(\widetilde{\theta},\widetilde{\boldsymbol{u}})$, given by equations (27),(30), (32) and (34), respectively.

For the case when using Euler angles, we considered the ZYZ convention, and employed the operational space linearizing controller first proposed in [7], which in the case of regulation, and considering only the orientation part of the pose, is given by

$$\boldsymbol{\tau} = M(\boldsymbol{q})J_a^{-1}(\boldsymbol{q})\left[K_p\widetilde{\boldsymbol{\phi}} - K_v\dot{\boldsymbol{\phi}} - \dot{J}_a(\boldsymbol{q})\dot{\boldsymbol{q}}\right] + C(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q})$$

where $J_a(\boldsymbol{q})$ is the analytic Jacobian of the manipulator, and $\dot{J}_a(\boldsymbol{q})$ its time derivative; K_p and K_v are positive definite control gains.

For the other three cases, i.e. when $\widetilde{\boldsymbol{\phi}}(\widetilde{R})$, $\widetilde{\boldsymbol{\phi}}(\widetilde{\eta},\widetilde{\boldsymbol{\varepsilon}})$, and $\widetilde{\boldsymbol{\phi}}(\widetilde{\theta},\widetilde{\boldsymbol{u}})$, we used the well–know resolved acceleration controller (RAC), first proposed by Luh et al [11]. RAC controller is widely utilized in pose control task, and there exist different versions of it, according to the definition of the orientation error utilized (see e.g. [15]). The RAC controller used in the simulations is given by

$$\boldsymbol{\tau} = M(\boldsymbol{q})J^{-1}(\boldsymbol{q})\left[K_p\widetilde{\boldsymbol{\phi}} - K_v\boldsymbol{\omega} - \dot{J}(\boldsymbol{q})\dot{\boldsymbol{q}}\right] + C(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q})$$

where now the geometric jacobian is used, and $\dot{\boldsymbol{\phi}}$ is replaced by $\boldsymbol{\omega}$.

The initial configuration for the simulations, given in terms of the joint variables, is $\boldsymbol{q}|_{t=0} = [\,0 \quad \frac{\pi}{2} \quad \frac{\pi}{2}\,]^T$ while the desired orientation is given by $\boldsymbol{q}_d = [\,\frac{\pi}{2} \quad \frac{\pi}{4} \quad 0\,]^T$. To measure the orientation error we use the Euclidean norm of $\widetilde{\boldsymbol{\phi}}$. The initial value of $\|\widetilde{\boldsymbol{\phi}}\|$ for each parameterization is

$$\|\widetilde{\boldsymbol{\phi}}(\widetilde{\alpha},\widetilde{\beta},\widetilde{\gamma})\|\Big|_{t=0} = 2.3562, \quad \|\widetilde{\boldsymbol{\phi}}(\widetilde{R})\|\Big|_{t=0} = 0.9892,$$

$$\|\widetilde{\boldsymbol{\phi}}(\widetilde{\theta},\widetilde{\boldsymbol{u}})\|\Big|_{t=0} = 1.7177, \quad \|\widetilde{\boldsymbol{\phi}}(\widetilde{\eta},\widetilde{\boldsymbol{\varepsilon}})\|\Big|_{t=0} = 0.7571.$$

Gain matrices where chosen to be diagonal, with suitable values so as to have a good performance in each of the simulations. Figure 2 shows the time evolution of $\|\widetilde{\boldsymbol{\varepsilon}}\|$ for the four cases; notice that in all of them the error measure converges to zero, meaning that the wrist end–effector orientation reaches the desired orientation.

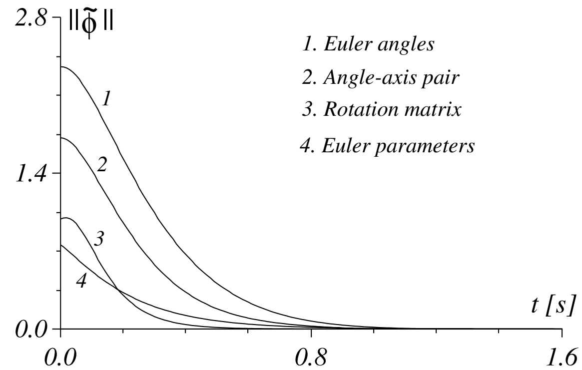

Fig. 2. Evolution of the orientation error measure.

VII. CONCLUSION

This paper has presented a review of some topics related with the different parameterizations of the orientation manifold. Four cases were considered: a) the Euler angles, b) the rotation matrix, c) the angle–axis pair, and d) the Euler parameters. For each case, some formulas were given to obtain an orientation vector $\boldsymbol{\phi}$, and the representation Jacobian, which relates $\dot{\boldsymbol{\phi}}$ with the body angular velocity. Also an orientation error vector was defined for each parameterization. Finally, some simulations were carried out on a prototypical wrist, and showed the fulfillment of the pose control aim using the different parameterizations of orientation.

References

[1] H. Goldstein, C. P. Poole, and J. L. Safko, Classical Mechanics, Addison-Wesley, 3rd. ed., 2001.

[2] C. Natale, Interaction Control of Robot Manipulators: Six degrees–of–freedom Tasks, Springer, 2003.

[3] R. M. Murray, Z. Li, and S. S. Sastry, A Mathematical Introduction to Robotic Manipulation. CRC Press, 1994.

[4] C. Samson, M. Le Borgne, and B. Espiau. Robot Control. The Task Function Approach, Clarendon Press, 1991.

[5] J. Gallier, Geometric Methods and Applications for Computer Science and Engineering. Springer–Verlag, 2000.

[6] J. B. Kuipers, Quaternions and rotation sequences, Princeton University Press, 1999.

[7] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation". IEEE Journal of Robotics and Automation. Vol. 3, pp. 43-53, 1987.

[8] R. Bach, and R. Paielli, "Linearization of attitude–control error dynamics", IEEE Transactions on Automatic Control, Vol. 38, pp. 1521–1525, 1993.

[9] R. Campa, and K. Camarillo, "Unit quaternions: A mathematical tool for modeling, path planning and control of robot manipulators", in Robot manipulators, M. Ceccarelli (ed.), In-Teh, pp. 21-48, 2008.

[10] L. Sciavicco, and B. Siciliano, Modeling and Control of Robot Manipulators, Springer-Verlag, 2000.

[11] J. Y. S. Luh, M. W. Walker, and R. P. C. Paul, "Resolved-acceleration control of mechanical manipulators". IEEE Transactions on Automatic Control. Vol. 25, pp. 486–474, 1980.

[12] R. Campa, R. Kelly, and V. Santibáñez, "Windows-based real-time control of direct-drive mechanisms: platform description and experiments', Mechatronics, Vol 14, No. 9, pp. 1021–1036, 2004.

[13] R. Campa, K. Camarillo, and L. Arias, "Kinematic modeling and control of robot manipulators via unit quaternions: application to a spherical wrist", Proc. IEEE Conference on Decision and Control, San Diego, CA, December 2006.

[14] M. W. Spong, S. Hutchinson, and M. Vidyasagar, Robot Modeling and Control, John Wiley & Sons, 2006.

[15] F. Caccavale, C. Natale, B. Siciliano, and L. Villani, "Resolved–acceleration control of robot manipulators: A critical review with experiments", Robotica, Vol. 16, pp. 565–573, 1998.