

# Redundancy Parametrization for Flexible Motion Control

Brian Moore\* and Erhan Oztop†

## Abstract

*Our overall research interest is in synthesizing human like reaching and grasping using anthropomorphic robot hand-arm systems, as well as understanding the principles underlying human control of these actions. When one needs to define the control and task requirements in the Cartesian space, the problem of inverse kinematics needs to be solved. For non-redundant manipulators, a desired end-effector position and orientation can be achieved by a finite number of solutions. For redundant manipulators however, there are in general infinitely many solutions where the cardinality of the solution set must be made finite by imposing certain constraints.*

*In this paper, we consider the Mitsubishi PA10 manipulator which is similar to the human arm, in the sense that both wrist and shoulder joints can be considered to emulate a 3DOF ball joint. We explicitly derive the analytic solution for the inverse kinematics using quaternions. Then, we derive a parameterization in terms of a pure quaternion called the swivel quaternion. The swivel quaternion is similar to the elbow swivel angle used in most approaches, but avoid the computation of inverse trigonometric functions. This parameterization of the self-motion manifold is continuous with any end-effector motion. Given the pose of the end-effector and the swivel quaternion (or swivel angle), the algorithm derives all solution of the inverse kinematics (finite number). We then show how the parameterization of the elbow self-motion can be used for the real-time control of the PA10 manipulator in the presence of obstacles.*

## KEYWORDS

*Inverse kinematics, redundancy resolution, self-motion, anthropomorphic manipulator, motion parameterization*

---

\*Brian Moore is with the ATR Cognitive Mechanisms Laboratories, 2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0288, Japan (e-mail: moore@atr.jp). He is supported by the Japan Society for the Promotion of Science.

†Erhan Oztop is with the ATR Cognitive Mechanisms Laboratories, 2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0288, Japan (e-mail: erhan@atr.jp), the Osaka University and the NICT Biological ICT Group.

## INTRODUCTION

Our motivation for studying redundancy in redundant manipulators is twofold. Firstly, we would like to develop a framework that facilitates the exploitation of the redundancy to its full extent for dexterous robot control. Secondly, we would like to have this framework to be instrumental for the analysis of human exploitation of redundancy in human reach and grasp. In general, redundant manipulators offer many advantages over non redundant systems. The redundancy can greatly improve the overall performance and versatility of the system by offering the possibility to avoid obstacles, compensating for the limitations imposed by the individual joint ranges and increasing dexterity. Anthropomorphic manipulators with 7-DOF which are similar to the human arm are very useful as their kinematics are open to analytic treatment. Moreover, due to their architectural similarities with the human arm, such manipulators are particularly suitable for programming by demonstration [1].

Our overall research interest is in synthesizing human like reaching and grasping [2] using anthropomorphic robot hand-arm systems, as well as understanding the principles underlying human control of these actions. We approach this problem from several directions including learning from human guided robot control [3, 4] and analysis of human reach and grasp [5]. Inverse kinematics is often necessary when one needs to define the control and task requirements in the Cartesian space. For non-redundant manipulators, the solution of the inverse kinematic for a generic pose (position and orientation) of the end-effector yields a finite number of solutions. For redundant manipulators, one has to solve the inverse kinematic problem and choose a solution in the joint space, or in other word, resolve the redundancy by introducing some constraints. Most approaches focus on the instantaneous relationship between the end-effector velocities and the joint velocities by looking at the Jacobian [6]. Optimization in the null space of the Jacobian is the standard way for exploiting the redundancy offered by the manipulator. However there are some drawbacks of these velocity based approaches: the manipulator can encounter singularities along the assigned trajectory, they are local and do not provide all solutions, and finally, since the approach involves the intensive computation of inverse trigonometric functions and matrix inversions, numerical error tend to accumulate and computation is often slow [7]. The other approach to inverse kinematics is to study the robotic manipulator at the position level by trying to derive an analytic representation of the self-motion of the manipulator [8, 9]. That is, those motions of the robot joints that leave the position of the end-effector unchanged. Such methods emphasize the global aspect rather than the local aspect which is of utmost importance for dynamic motion-planning and control. Once a closed form general solution to the inverse kinematics problem has been found, one can easily convert a desired Cartesian trajectory to a joint trajectory fulfilling a desired optimization criteria, such as obstacle avoidance. Alternatively, one can use the closed form solution in a real-time fashion, for example as in real-time human control of the robot, to impose certain optimization criteria, such as minimum joint angle change.

Finding an analytic solution for the inverse kinematic of redundant manipulators is not always possible in practice. Even for (non-redundant) 6-DOF manipulators, the problem is difficult. Recent work have looked at novel ways to compute all solutions (finite number) of the inverse kinematics of 6R chains taking advantage of symbolic computation using kinematic mapping [10] or double quaternions [11]. For redundant manipulators, the set of solutions (i.e. self motion) for the inverse kinematics lie on several manifolds in the joint space. The topology of these self motion manifolds have been studied in [8,12,13]. For redundant manipulators with special architecture, it is sometimes possible to derive an analytic solution of the inverse kinematic. This is the case for anthropomorphic arms where, for a fixed pose of the hand (i.e. end-effector), the elbow is free to rotate on a circle, see for instance [14,15].

In this paper, we consider the Mitsubishi PA10 manipulator (see Figure 1) which is similar to the human arm (i.e. both wrist and shoulder joints can be considered to emulate a 3DOF ball joint). We explicitly derive the analytic solution for the inverse kinematics based on previous work on the inverse kinematics of anthropomorphic arms [15–17] using quaternions. Then, we derive a parameterization in terms of a pure quaternion called the swivel quaternion. The swivel quaternion is similar to the elbow swivel angle used in most approaches, but avoid the computation of inverse trigonometric functions. This parameterization of the self-motion manifold is continuous with any end-effector motion.

Given the pose of the end-effector and the swivel angle, the algorithm derives all solution of the inverse kinematics (finite number). We then show how the parameterization of the elbow self-motion can be used for the real-time control of the PA10 manipulator following a human arm motion in presence of fixed or moving obstacles.

## INVERSE KINEMATICS OF THE PA10

### Notation

The Mitsubishi PA10 is a 7-DOF redundant arm which is widely used [18,19]. The joints alternate between roll joints and pitch joints: that is, joints 1,3,5 and 7 are roll joints and joints 2,4 and 6 are pitch joints as shown in Figure 2. Denote by  $\theta_i$  ( $i = 1..7$ ) the joint angles values and let  $\boldsymbol{\theta} = [\theta_1, \dots, \theta_7]$ . The lengths of the links are denoted by  $L_j$  ( $j = 1..4$ ). In Figure 2, the zero configuration of the manipulator, that is the configuration where  $\theta_i = 0$  for  $i = 1, \dots, 7$ , is shown.

Here is a short review of quaternions which are used in this paper to represent orientation. We refer to classical textbook including [20] for more details on this topic. Let  $\mathbf{q} = q_s + q_x \mathbf{i} + q_y \mathbf{j} + q_z \mathbf{k}$  with  $q_s, q_x, q_y, q_z \in \mathbb{R}$  and  $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$ . If  $q_s^2 + q_x^2 + q_y^2 + q_z^2 = 1$ ,  $q$  is called a *unit quaternion*. Alternatively, a unit quaternion  $\mathbf{q}$  can be written as

$$\mathbf{q} = \cos\left(\frac{\alpha}{2}\right) + q_x \sin\left(\frac{\alpha}{2}\right)\mathbf{i} + q_y \sin\left(\frac{\alpha}{2}\right)\mathbf{j} + q_z \sin\left(\frac{\alpha}{2}\right)\mathbf{k} \quad (1)$$

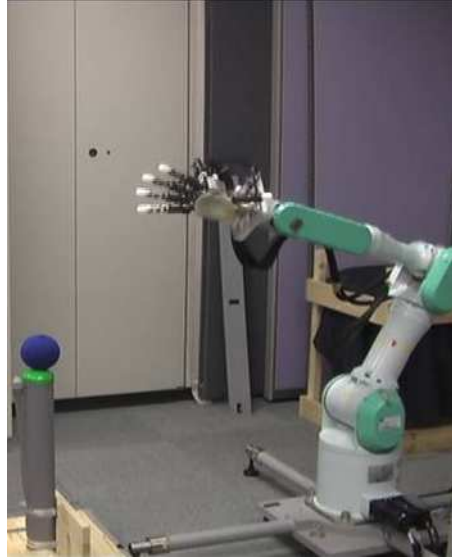


Fig. 1: Mitsubishi PA10

where  $\alpha$  is the rotation angle and the unit vector  $(q_x, q_y, q_z)$  represents the rotation axis. A vector  $(v_x, v_y, v_z) \in \mathbb{R}^3$  can be represented by a *pure quaternion*  $\mathbf{v} = v_x \mathbf{i} + v_y \mathbf{j} + v_z \mathbf{k}$ , that is, we regards  $\mathbf{i}$ ,  $\mathbf{j}$  and  $\mathbf{k}$  as the  $x$ ,  $y$  and  $z$  axis respectively. The rotation of  $v$  by an angle  $\theta$  about the axis  $(q_x, q_y, q_z)$  is given by:

$$\mathbf{v}' = \mathbf{q}\mathbf{v}\bar{\mathbf{q}} \quad (2)$$

where  $\bar{\mathbf{q}}$  is the conjugate of  $\mathbf{q}$  given by  $\bar{\mathbf{q}} = q_s - q_x \mathbf{i} - q_y \mathbf{j} - q_z \mathbf{k}$ . This rotation can also be represented by a rotation matrix  $R$  and the relation between  $R$  and  $\mathbf{q}$  is given by:

$$R = \begin{bmatrix} q_s^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_s q_z) & 2(q_x q_z + q_s q_y) \\ 2(q_x q_y + q_s q_z) & q_s^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_y q_z - q_s q_x) \\ 2(q_x q_z - q_s q_y) & 2(q_y q_z + q_s q_x) & q_s^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix} \quad (3)$$

For implementation purposes, it is important to notice that every rotation in 3D corresponds to two unit quaternions  $\mathbf{q}$  and  $-\mathbf{q}$  which yield the same rotation [20, 21].

### Forward kinematics

For given values of the joint angles  $\boldsymbol{\theta}$ , the forward (or direct) kinematic function  $f$  computes the position  $\mathbf{p}$  and orientation  $\mathbf{q}$  of the end-effector, where  $\mathbf{p}$  is a pure quaternion and  $\mathbf{q}$  is a unit quaternion:

$$\begin{bmatrix} \mathbf{p} \\ \mathbf{q} \end{bmatrix} = \begin{bmatrix} f_p(\boldsymbol{\theta}) \\ f_q(\boldsymbol{\theta}) \end{bmatrix} = f(\boldsymbol{\theta}) \quad (4)$$

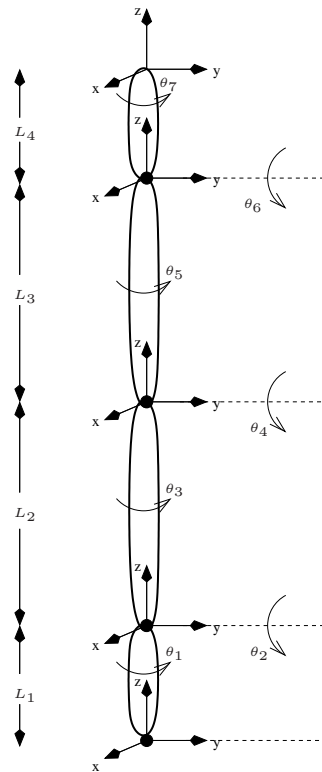


Fig. 2: Notation used for the PA10.

Let  $\mathbf{Q}_y(\theta_i)$  and  $\mathbf{Q}_z(\theta_i)$  representing a rotation of  $\theta_i$  about the  $y$  and  $z$  axis respectively, that is:

$$\begin{aligned}\mathbf{Q}_y(\theta_i) &= \cos\left(\frac{\theta_i}{2}\right) + \sin\left(\frac{\theta_i}{2}\right)\mathbf{j} \\ \mathbf{Q}_z(\theta_i) &= \cos\left(\frac{\theta_i}{2}\right) + \sin\left(\frac{\theta_i}{2}\right)\mathbf{k}\end{aligned}\quad (5)$$

then

$$f_q(\boldsymbol{\theta}) = \mathbf{Q}_z(\theta_1)\mathbf{Q}_y(\theta_2)\mathbf{Q}_z(\theta_3)\mathbf{Q}_y(\theta_4)\mathbf{Q}_z(\theta_5)\mathbf{Q}_y(\theta_6)\mathbf{Q}_z(\theta_7) \quad (6)$$

From (6), denote by  $\mathbf{q}_i$  the transformation up to joint angle  $i$ . For instance:

$$\mathbf{q}_3 = \mathbf{Q}_z(\theta_1)\mathbf{Q}_y(\theta_2)\mathbf{Q}_z(\theta_3) \quad (7)$$

Let

- $\mathbf{p}_S$ : position of the shoulder joint
- $\mathbf{p}_E$ : position of the elbow joint
- $\mathbf{p}_W$ : position of the wrist joint

as shown in Figure 3. The positions of these joints are given by:

$$\begin{aligned}\mathbf{p}_S &= L_1\mathbf{k} \\ \mathbf{p}_E &= \mathbf{p}_S + L_2\mathbf{q}_2\mathbf{k}\overline{\mathbf{q}}_2 \\ \mathbf{p}_W &= \mathbf{p}_E + L_3\mathbf{q}_4\mathbf{k}\overline{\mathbf{q}}_4\end{aligned}\quad (8)$$

The position of the end-effector is then given by:

$$f_p(\boldsymbol{\theta}) = \mathbf{p}_W + L_4\mathbf{q}_6\mathbf{k}\overline{\mathbf{q}}_6 \quad (9)$$

## Redundancy parameterization

In this section, we consider a fixed pose (position  $\mathbf{p}$  and orientation  $\mathbf{q}$ ) of the end-effector and we parameterize the set of all possible positions of the elbow. In this case, since the pose of the end-effector is known, the position of the wrist  $\mathbf{p}_W$  can be derived from (9) using the fact that  $\mathbf{q}\mathbf{k}\overline{\mathbf{q}} = \mathbf{q}_6\mathbf{k}\overline{\mathbf{q}}_6$ :

$$\mathbf{p}_W = \mathbf{p} - L_4(\mathbf{q}\mathbf{k}\overline{\mathbf{q}}) \quad (10)$$

Since  $\mathbf{p}_S$  is fixed for any pose of the end-effector, the only moving joint is the elbow joint (see Figure 3). Let  $A$  be the line joining  $\mathbf{p}_S$  and  $\mathbf{p}_W$ . From the geometry of the problem, it is clear that the elbow is restricted to move on a circle with center  $C$  located on  $A$  and the circle being perpendicular to  $A$ . Alternatively, this circle can be represented by the position of its centre  $C$ , its normal vector  $\mathbf{n}$  parallel to  $A$  and its radius  $\rho$  (see Figure 3).

An alternative and equivalent way to analyze the motion of the elbow is to replace the PA10 manipulator by an equivalent 7R manipulator consisting of two spherical joints (at the shoulder and wrist) and a revolute joint at the elbow. Actually, combining the three first joints, we obtain a rotation described

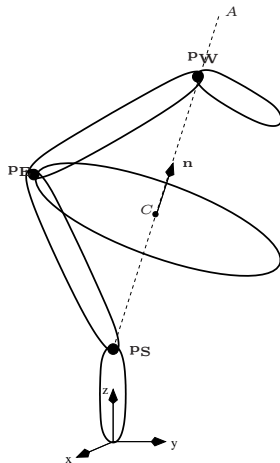


Fig. 3: Elbow self motion for a fixed pose of the end-effector.

by  $ZYZ$  angles. It is also possible to find the inverse relation, that is, given a rotation  $\mathbf{q}_3 = q_s + q_x \mathbf{i} + q_y \mathbf{j} + q_z \mathbf{k}$  (see (7)), we can recover the angles  $\theta_1, \theta_2, \theta_3$ . The explicit equations are given in [22]<sup>1</sup>:

$$\begin{aligned} \theta_1 &= \text{Atan2}(R[2, 3], R[1, 3]) \\ \theta_2 &= \text{Atan2}(\sqrt{R(1, 3)^2 + R(2, 3)^2}, R(3, 3)) \\ \theta_3 &= \text{Atan2}(R(3, 2), -R(3, 1)) \end{aligned} \quad (11)$$

for  $\theta_3 \in [0, \pi]$  and by

$$\begin{aligned} \theta_1 &= \text{Atan2}(-R[2, 3], -R[1, 3]) \\ \theta_2 &= \text{Atan2}(-\sqrt{R(1, 3)^2 + R(2, 3)^2}, R(3, 3)) \\ \theta_3 &= \text{Atan2}(-R(3, 2), -R(3, 1)) \end{aligned} \quad (12)$$

for  $\theta_3 \in [-\pi, 0]$ , where  $R$  is the rotation corresponding to  $\mathbf{q}_3$  (see (7)). Using (3), (11) and (12) can be rewritten in terms of  $q_s, q_x, q_y, q_z$ .

In this case, the motion of the elbow can be seen as the intersection of two spheres  $S_1$  and  $S_2$ . The first sphere,  $S_1$  is a fixed sphere (i.e.: not depending on the end-effector pose) centered at  $\mathbf{p}_S$  of radius  $L_2$ .  $S_2$  is a sphere centered at  $\mathbf{p}_W$  of radius  $L_3$  which is moving according to the pose of the end-effector. This idea of splitting the manipulator in two sub-chains is described for instance in [10, 16]. Therefore, for any motion of the end-effector, the set of possible positions of the elbow is a moving circle on the fixed sphere  $S_1$  (see Figure 4).

Since the self-motion manifold of the elbow is a circle, it can be parameterized. Consider a unit vector  $\mathbf{u}$  (seen as a unit pure quaternion) in the  $xy$  plane making an angle  $s$  with the  $x$  axis as shown in Figure 5.  $\mathbf{u}$  is given by

$$\mathbf{u} = \mathbf{Q}_z(s) \mathbf{i} \overline{\mathbf{Q}_z(s)} \quad (13)$$

<sup>1</sup> See p. 30-32

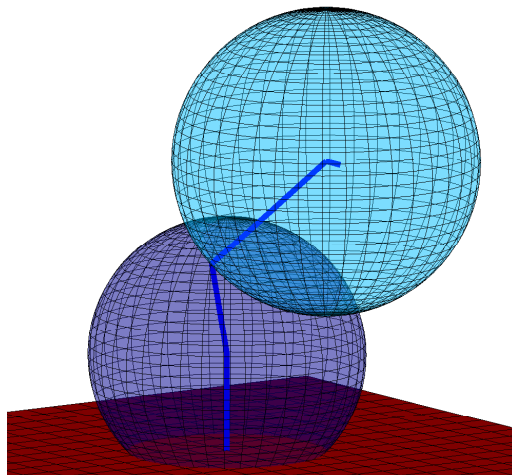


Fig. 4: Intersection of two spheres in one circle

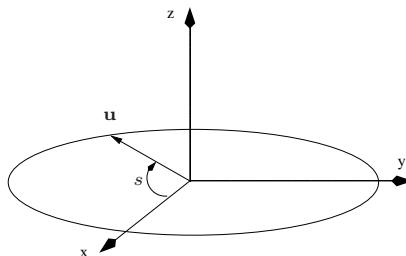


Fig. 5: Parameterization of the self-motion manifold of the elbow either using the swivel angle  $s$  or the associated pure unit quaternion  $\mathbf{u}$ .

Both  $\mathbf{u}$  and  $s$  can be used for parameterizing the self motion of the elbow. Using  $\mathbf{u}$  has several advantages over the use of  $s$ . First, each unit pure quaternion in the  $xy$  plane corresponds to one value of the parameterization contrarily to the use of  $s$  where  $s = k(2\pi)$ , for  $k \in \mathbb{Z}$ . Moreover, using  $\mathbf{u}$ , the computation of unnecessary trigonometric functions can be avoided therefore reducing computational cost and numerical errors.

This parameterization should be continuous: for any continuous motion of the end-effector and any continuous choice of  $\mathbf{u}$ , the motion of the elbow should be continuous as well. For a continuous motion of the end-effector, the motion of  $\mathbf{n}$  is also continuous. Therefore consider  $\mathbf{q}_A$  such that

$$\mathbf{n} = \mathbf{q}_A \mathbf{k} \overline{\mathbf{q}_A} \quad (14)$$

the *shortest rotation* rotating  $\mathbf{k}$  onto  $\mathbf{n}$ . That is, the rotation about  $\mathbf{k} \times \mathbf{n}$  of an angle  $s$ . Figure 6 illustrates the idea of the mapping. Given  $\mathbf{u}$ , we can use this

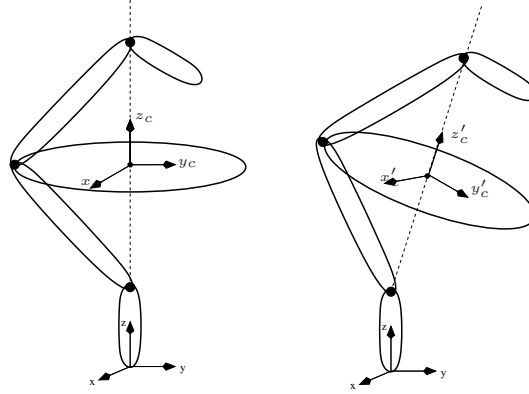


Fig. 6: Notation for the PA10

mapping to obtain the position of the elbow:

$$\mathbf{p}_E = C + \rho \mathbf{q}_A \mathbf{u} \overline{\mathbf{q}_A} \quad (15)$$

Once  $\mathbf{p}_E$  is known then we can obtain the orientation of the elbow joint starting from the end-effector position and orientation. As will be shown in the next section this is the key to the solution of the inverse kinematics as the orientation of elbow joint can also be computed starting from the base. Using (15) it is straightforward to compute the value of  $\mathbf{u}$  given the position of the elbow.

$$\mathbf{u} = \overline{\mathbf{q}_A} \left( \frac{\mathbf{p}_E - C}{\rho} \right) \mathbf{q}_A \quad (16)$$

Eq. (16) can be used to study how the redundancy is used for a particular given motion.

### Inverse kinematics

The inverse kinematic problem can be formulated as follow: given a pose of the end-effector and  $\mathbf{u}$  a unit pure quaternion in the  $xy$  plane, find  $\boldsymbol{\theta}$ . Here is a summary of the algorithm:

1. Compute  $\mathbf{p}_W$  from (10).
2. Compute the circle representing the elbow self-motion.
3. Given  $\mathbf{u}$ (or  $s$ ), compute  $\mathbf{p}_E$  from (15).
4. Compute  $\theta_4$  using the triangle made up by  $\mathbf{p}_S$ ,  $\mathbf{p}_E$  and  $\mathbf{p}_W$ .
5. Compute the desired value of  $\mathbf{q}_3$ .
6. Compute  $\theta_1$ ,  $\theta_2$  and  $\theta_3$  using (11), (12) and  $\mathbf{q}_3$ .

7. Compute  $\mathbf{q}_4$
8. Compute  $\theta_5$ ,  $\theta_6$  and  $\theta_7$  using (11), (12) and by seeing  $\mathbf{q}_4$  as the orientation to be achieved by the spherical joint at the wrist.

Using this algorithm, we obtain a valid solution for the inverse kinematics. However, one has to consider that for each point on this circle, there are many different joint angles solving the inverse kinematics. In fact, one can use the following equivalence relationship<sup>2</sup>:

1.  $\theta_4 \rightarrow -\theta_4$ ,  $\theta_3 \rightarrow \theta_3 \pm \pi$ ,  $\theta_5 \rightarrow \theta_5 \pm \pi$
2.  $\theta_2 \rightarrow -\theta_2$ ,  $\theta_1 \rightarrow \theta_1 \pm \pi$ ,  $\theta_3 \rightarrow \theta_3 \pm \pi$
3.  $\theta_6 \rightarrow -\theta_6$ ,  $\theta_7 \rightarrow \theta_7 \pm \pi$ ,  $\theta_5 \rightarrow \theta_5 \pm \pi$

In other words, every point on the circle have multiplicity 8 ( $2*2*2$ ) if all joint angles are limited to  $[-\pi, \pi]$  but using customized min-max can modify the number of solutions.

### Example

The following example illustrates the steps required to compute the inverse kinematics for the following pose of the end effector

$$\mathbf{p} = 0.5\mathbf{i} + 0.1\mathbf{j} + \mathbf{k} \quad \mathbf{q} = 0.7071 + 0.7071\mathbf{j} \quad (17)$$

and  $\mathbf{u} = \mathbf{i}$  (i.e.  $s = 0$ ). The lengths of the links are given by  $L_1 = 0.315$ ,  $L_2 = 0.45$ ,  $L_3 = 0.5$  and  $L_4 = 0.08$ . The shoulder is fixed and its position is given by  $\mathbf{p}_S = 0.315\mathbf{k}$ . Here are the main steps of the algorithm described in section 2.4.

1. The first step is to compute the position of the wrist using equation (10):

$$\mathbf{p}_W = 0.42\mathbf{i} + 0.1\mathbf{j} + \mathbf{k} \quad (18)$$

2. Using the geometry of the manipulator and considering the possible position of the elbow located on the intersection of the sphere  $S_1$  and  $S_2$ , we obtain  $C = 0.1948\mathbf{i} + 0.04638\mathbf{j} + 0.6327\mathbf{k}$ ,  $\mathbf{n} = 0.5187\mathbf{i} + 0.1235\mathbf{j} + 0.846\mathbf{k}$  and  $\rho = 0.2480$ .

3.  $\mathbf{q}_A$  as described in (14) is given by:

$$\mathbf{q}_A = 0.9607 - 0.06428\mathbf{i} + 0.27\mathbf{j} \quad (19)$$

Note that the vector part of  $\mathbf{q}_A$  corresponds to the cross-product of  $\mathbf{k}$  by  $\mathbf{n}$ . Given  $\mathbf{q}_A$ , the position of the elbow joint can be computed using equation (15)

$$\mathbf{p}_E = 0.4066\mathbf{i} + 0.03777\mathbf{j} + 0.5041\mathbf{k} \quad (20)$$

---

<sup>2</sup> In this paper, we assume that  $0 \leq \theta_4 < \pi$  to mimic the human elbow motion.

4. We obtain that

$$\theta_4 = \text{acos}\left(\frac{\rho}{L_3}\right) + \text{acos}\left(\frac{\rho}{L_2}\right) - \pi = -1.1025 \quad (21)$$

5. Since the elbow joint is a revolute joint which rotation axis is perpendicular to both links which are attached to it. It is possible to find  $\mathbf{q}_3$  since:

$$\mathbf{q}_3 \mathbf{k} \overline{\mathbf{q}}_3 = \text{unit}(\mathbf{p}_E - \mathbf{p}_S) \quad (22)$$

and

$$\mathbf{q}_3 \mathbf{j} \overline{\mathbf{q}}_3 = (\mathbf{p}_E - \mathbf{p}_S) \times (\mathbf{p}_E - \mathbf{p}_W) \quad (23)$$

Using this information, we obtain:

$$\mathbf{q}_3 = 0.8425 - 0.06156\mathbf{i} + 0.5349\mathbf{j} + 0.01849\mathbf{k} \quad (24)$$

6. Using (11) and (12), the first three angles can be computed:

$$\theta_1 = 0.0926, \theta_2 = 1.1372, \theta_3 = -0.1365 \quad (25)$$

7. Since  $\theta_4$  is also known, we obtain

$$\mathbf{q}_4 = 0.9978 - 0.06212\mathbf{i} + 0.01445\mathbf{j} + 0.01649\mathbf{k} \quad (26)$$

8. From (6), we can write the ZYZ rotation of the last three joints as:

$$\overline{\mathbf{q}}_4 \mathbf{q} = 0.7158 + 0.05559\mathbf{i} + 0.6954\mathbf{j} + 0.03227\mathbf{k} \quad (27)$$

since  $\mathbf{Q}_z(\theta_5)\mathbf{Q}_y(\theta_6)\mathbf{Q}_z(\theta_7) = \overline{\mathbf{q}}_4 \mathbf{q}$ . Using (11) and (12), the last three angles can be computed:

$$\theta_5 = -0.0347, \theta_6 = 1.5440, \theta_7 = 0.1248 \quad (28)$$

## MOTION CONTROL AND OBSTACLE AVOIDANCE

In the previous section, a method to compute the complete solution of the inverse kinematics problem with the arbitrary open parameter  $\mathbf{u}$  (or  $s$ ) was given. In this section, we illustrate how to use this parametrization to our advantage. For simplicity, we describe the approach using  $s$  and show in the example how  $\mathbf{u}$  can be used instead. In motion planning, we are given a desired Cartesian trajectory (a time changing position and orientation of the end-effector) and the goal is to find a smooth joint trajectory with certain constraints. The constraints might be due to the environment (e.g. obstacle avoidance) or intrinsic (e.g. self collision avoidance and singularity avoidance). The goal is then to find a function  $s(t)$  such that the constraints will be satisfied over the whole trajectory. Let the functional

$$C = \int E(g(t), s(t)) dt \quad (29)$$

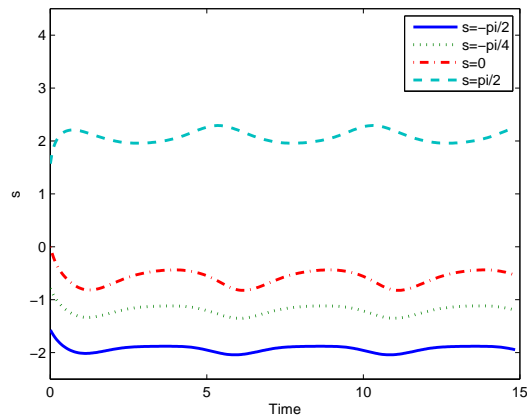


Fig. 7: Example 1: Values of the parameter  $s$  for different fixed  $s_{\text{des}}$  in the presence of obstacle.

define the cost required to follow the Cartesian trajectory  $g(t)$  with  $s(t)$ , where  $E$  is the instantaneous cost. In this cost one encodes the constraint satisfaction criteria. Then the problem consists in finding the function  $s(t)$  that minimizes the functional  $C$ . This problem can be solved using variational calculus akin to optimal control problems. In the scenario where the desired Cartesian trajectory is not available in advance, this method however, is not applicable. This is the case when the human hand position and orientation are to be imitated by the robot end effector in real-time. For this case, one can revert to a local approach by defining  $s$  as a dynamical system

$$\tau \dot{s} = \lambda_1 h(s_{\text{des}} - s) + \sum_{i=2}^N \lambda_i F_i \quad (30)$$

where  $h$  is a non-decreasing *odd function*<sup>3</sup> and  $\sum_{i=1}^N \lambda_i = 1$ , with  $\lambda_i \geq 0$ , for  $i = 1..N$ . For example, one can use

$$\tau \dot{s} = \lambda_1 h(s_{\text{des}} - s) + \lambda_2 F_{\text{obst}} + \lambda_3 F_{\text{singl}} + \lambda_4 F_{\text{limits}} \quad (31)$$

where  $F_{\text{obst}}$ ,  $F_{\text{singl}}$ ,  $F_{\text{limits}}$  are forces pushing the robot away from obstacles, singularities and joint limits respectively, by increasing or decreasing the value of  $s$ .

As an example, consider the following obstacle avoidance problem. Let an object position be defined by  $O$  and the minimum euclidean distance between the robot and the object given by a function  $d(\mathbf{p}, \mathbf{q}, s, O)$  with  $\mathbf{p}$  and  $\mathbf{q}$  being the position and orientation of the end-effector. Let the dynamics of  $s$  be defined

<sup>3</sup>  $-h(x) = h(-x)$

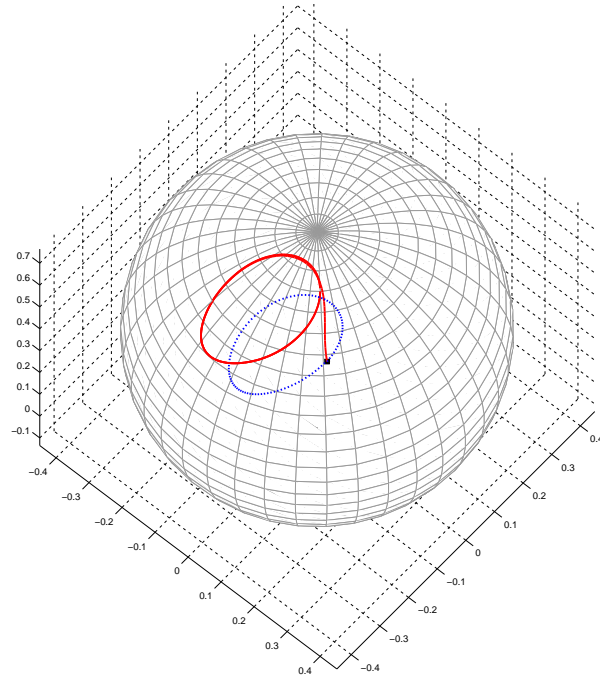


Fig. 8: Example 1: Motion of the elbow on the sphere  $S_1$  with obstacle (in red) and without obstacle (in dashed blue) for fixed  $s_{\text{des}} = \frac{-\pi}{2}$ .

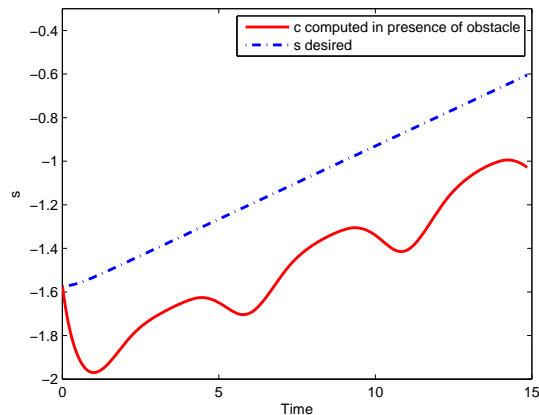


Fig. 9: Example 2: Values of the parameter  $s$  (in red) in the presence of an obstacle based on a linearly increasing  $s_{\text{des}}$  (in dashed blue).

as:

$$\tau \dot{s} = \lambda \tanh(s_{\text{des}} - s) + (1 - \lambda) F_{\text{obst}} \quad (32)$$

where  $F_{\text{obst}} \propto \frac{\partial d}{\partial s}$ . Computationally, the minimum distance  $d$  between the obstacles and the robot for small increments and decrements of  $s$  is used to determine the direction of  $F_{\text{obst}}$ . The amplitude of  $F_{\text{obst}}$  is computed using the inverse of the absolute current distance to the obstacles. Using  $\mathbf{u}$ , the exact same approach can be used by computing the distance to the obstacles using a small rotational change of  $\mathbf{u}$  by  $2\epsilon$  given by  $\mathbf{r}_+\mathbf{u}$  and  $\mathbf{r}_-\mathbf{u}$ , where  $\mathbf{r}_\pm = (\cos(\pm\epsilon) + \sin(\pm\epsilon)\mathbf{k})$  for a small value  $\epsilon$ . Since  $\epsilon$  is chosen a priori as a constant, no trigonometric computation is involved in the computation of  $\mathbf{r}_\pm\mathbf{u}$ .

Consider that the end-effector is moving around a circle (three times) in the  $xy$  plane with constant orientation in presence or absence of a spherical obstacle. In the first experiment,  $s_{\text{des}}$  is fixed and kept constant during the motion in presence of the obstacle. Figure 7 illustrates the changes of  $s$  in presence of obstacles for different fixed  $s_{\text{des}}$  while Figure 8 shows how the motion of the elbow changes to avoid the obstacle. In the second example, desired  $s(t)$  is given as a linearly increasing ramp. In the presence of an obstacle, the dynamics of  $s$  to avoid the obstacle while at the same time trying to track the desired ramp as shown in Figure 9. The motion of the elbow with this desired  $s(t)$  is shown in Figure 10 both with and without the obstacle.

## CONCLUSION AND FUTURE WORK

In this paper, we presented a generic framework to analytically compute the inverse kinematics of the anthropomorphic PA10 redundant manipulators and parameterizing the motion of the elbow using the swivel quaternion. We have

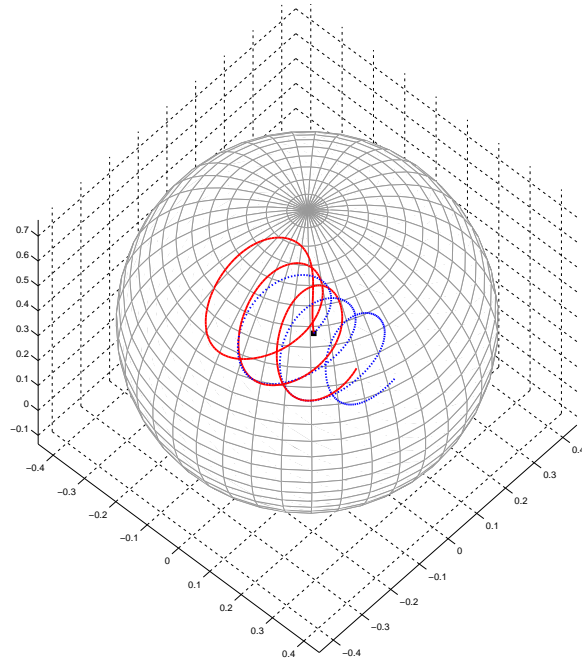


Fig. 10: Example 2: Motion of the elbow on the sphere  $S_1$  with obstacle (in red) and without obstacle (in dashed blue) for linearly increasing  $s_{\text{des}}$ .

also shown how this approach can be used for obstacle avoidance by using the parameterization of the elbow position as a dynamical system.

We believe that the formalism presented in this paper will be useful for our current work in order to understand, model and reproduce human like reaching and grasping using anthropomorphic robotic arm/hand.

## ACKNOWLEDGEMENT

The authors would like to thank Emre Ugur who has developed the simulation tools used for this project and for his technical support.

## References

- [1] Calinon, S., 2009. *Robot programming by demonstration: A probabilistic approach*. EPFL Press.
- [2] Ugur, E., Sahin, E., and Oztop, E., 2009. “Affordance learning from range data for multi-step planning”. In Ninth International Conference on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems.
- [3] Oztop, E., Lin, L., Kawato, M., and Cheng, G., 2006. “Dexterous skills transfer by extending human body schema to a robotic hand”. In IEEE-RAS International Conference on Humanoid Robots, Genova, Italy.
- [4] Oztop, E., Lin, L., Kawato, M., and Cheng, G., 2007. “Extensive human training for robot skill synthesis: Validation on a robotic hand”. In IEEE International Conference on Robotics and Automation, Roma, Italy.
- [5] Ozyer, B., and Oztop, E., 2008. “Task dependent human-like grasping”. In IEEE-Ras International Conference on Humanoid Robots(Humanoids 2008), pp. 177–182.
- [6] Sciavicco, L., and Siciliano, B., 1988. “A solution algorithm to the inverse kinematic problem for redundant manipulators”. *IEEE Journal of Robotics and Automation*, **4**(4), pp. 403–410.
- [7] Tarokh, M., Keerthi, K., and Lee, M., 2010. “Classification and characterization of inverse kinematics solutions for anthropomorphic manipulators”. *Robotics and Autonomous Systems*, **58**, pp. 115–120.
- [8] Burdick, J., 1989. “On the inverse kinematics of redundant manipulators: characterization of the self-motion manifolds”. In Proceedings of the 1989 IEEE International Conference on Robotics and Automation, pp. 264–270.
- [9] Kreutz, K., Long, M., and Seraji, H., 1992. “Kinematic analysis of 7 dof manipulators”. *International Journal of Robotics Research*, **11**(5), pp. 469–481.

- 
- [10] Husty, M., Pflurner, M., and Schröcker, H.-P., 2007. “A new and efficient algorithm for the inverse kinematics of a general serial 6r manipulator”. *Mechanism and Machine Theory*, **42**(1), pp. 66–81.
- [11] Qiao, S., Liao, Q., Wei, S., and Su, H.-J., 2010. “Inverse kinematic analysis of the general 6r serial manipulators based on double quaternions”. *Mechanism and Machine Theory*, **45**(2), pp. 193–199.
- [12] Thomas, F., 1993. *Computational kinematics*. Kluwer Academic Publishers, Dordrecht, Angeles, J., Hommel, G., Kovács, P. (Eds.), ch. The self-motion manifold of the N-bar mechanism, pp. 95–107.
- [13] DeMers, D., and Kreutz-Delgado, K., 1994. “Canonically parameterized families of inverse kinematic functions for redundant manipulators”. In ICRA, pp. 1881–1886.
- [14] Hollerbach, J., 1984. “Optimum kinematic design for a seven degree of freedom manipulator”. In Proc. 2nd Int. Symp. Robotics Research, pp. 215–222.
- [15] Tolani, D., Goswami, A., and Tomović, R., 2000. “Real-time inverse kinematics techniques for anthropomorphic limbs”. *Graph. Models*, **62**, pp. 353–388.
- [16] Korein, J., 1985. “A geometric investigation of reach”. PhD thesis, University of Pennsylvania.
- [17] Kallmann, M., 2008. “Analytical inverse kinematics with body posture control”. *Computer animation and virtual worlds*, **19**, pp. 79–91.
- [18] Peters, J., and Schaal, S., 2008. “Learning to control in operational space”. *The International Journal of Robotics Research*, **27**(2), pp. 197–212.
- [19] Klanke, S., Lebedev, D., Haschke, R., Steil, J., and Ritter, H., 2006. “Dynamic path planning for a 7-dof robot arm”. In Proc. International Conference on Intelligent Robots and Systems (IROS), pp. 3879–3884.
- [20] McCarthy, J., 1990. *Introduction to theoretical kinematics*. MIT Press, Cambridge.
- [21] Selig, J., 2000. *Geometric fundamentals of robotics*. Springer.
- [22] Sciavicco, L., and Siciliano, B., 2000. *Modeling and control of robot manipulators*. Springer-Verlag Advanced Textbooks in Control and Signal Processing Series, London.