INTECH
open science | open minds

OPEN ACCESS
ARTICLE

International Journal of Advanced Robotic Systems

Inverse Kinematics of a Humanoid Robot with Non-Spherical Hip: A Hybrid Algorithm Approach

Regular Paper

Rafael Cisneros Limón[1,2,*], Juan Manuel Ibarra Zannatha[3] and Manuel Ángel Armada Rodríguez[4]

1 University of Tsukuba, Tsukuba, Japan
2 CNRS-AIST JRL (Joint Robotics Laboratory) UMI3218/CRT, Tsukuba, Japan
3 Department of Automatic Control - Cinvestav, Mexico City, Mexico
4 Centre for Automation and Robotics - CAR (CSIC - UPM), Madrid, Spain
* Corresponding author E-mail: rafael.cisneros@aist.go.jp

Received 28 May 2012; Accepted 10 Dec 2012

DOI: 10.5772/55464

Abstract This paper describes an approach to solve the inverse kinematics problem of humanoid robots whose construction shows a small but non negligible offset at the hip which prevents any purely analytical solution to be developed. Knowing that a purely numerical solution is not feasible due to variable efficiency problems, the proposed one first neglects the offset presence in order to obtain an approximate "solution" by means of an analytical algorithm based on screw theory, and then uses it as the initial condition of a numerical refining procedure based on the Levenberg-Marquardt algorithm. In this way, few iterations are needed for any specified attitude, making it possible to implement the algorithm for real-time applications. As a way to show the algorithm's implementation, one case of study is considered throughout the paper, represented by the SILO2 humanoid robot.

Keywords Humanoid Robots, Inverse Kinematics, Non-Spherical Hip

1. Introduction

The kinematic modeling of humanoid robots is in fact a more difficult problem when compared to the classical industry manipulators, due to the locomotion that these robots are expected to perform while maintaining, at the same time, their postural stability. It is a formidable computational challenge, especially for real-time applications, due to the high number of degrees of freedom, the complex kinematics configurations that may lead to non-analytical solutions and the constantly changing point of support that is not rigidly attached to the floor [1]. In order to control the motion of these humanoid robots, specific trajectories in task space are required; that is, a desired sequence of attitudes for the feet and the body are required for accomplishing the desired stable locomotion task.

In the context of rigid multi-body systems, as is the case of robots, task space is different from joint space where motor commands are issued. Hence, movement planning requires either a preset sequence generated with *play-back* techniques or a suitable coordinate transformation from task to joint space. The former method is very simple as it does not require a complex mathematical analysis; however, it is not systematic as it is based on empirical tests for each movement, which make it tedious and non practical. The latter one, based on coordinates transformation, is supposed to be found by solving the so called Inverse Kinematics (IK) problem, which arises from the fact that the inverse transformation is often complex and ill-posed [2].

It is known that the IK problem cannot be solved by means of systematic processes as it is highly dependent on the robot's kinematic structure, and that it also either yields to infinite, multiple, one or none solutions. Only in simple cases, as in the case when the kinematic chain has no more than 6 degrees-of-freedom (dof) and three joint axes intersect, the IK problem for position and orientation can be decoupled, as stated by the so popular method proposed by Pieper in his PhD thesis [3] [4]. This is the case of many industrial manipulators and some of the most popular humanoid robots, i.e. ASIMO of Honda, HRP-2 of AIST and Kawada Industries or HOAP-2 of Fujitsu [5]. However, even for singularity-free mechanisms, the presence of offsets prevents this fact. Therefore, closed-form solutions are difficult, if not impossible, to find [6]. Or if they exist, they are very specific for a given kinematic chain [7] [8] [9].

For this reason, numerical methods have been commonly used to solve the IK problem, by using the pseudoinverse of the Jacobian (or a modification of it) [2] [10], the Cyclic Coordinate Descent (CCD) method [11] [12] or a procedure based on iterative triangulation [13] [14]. Some authors also use neural approaches [15] [16] or computational approaches based on conformal geometric algebra [17] or symbolic preprocessing [18]. However, these ones cannot be as efficient as the analytical ones, as they may require several iterations (not known *a priori*) in order to reach a particular precision, leading in many cases to computationally expensive solutions, which can be numerically unstable [19] [20]. Not to mention that a numerical algorithm may converge to a *local minima* instead of the desired global one, represented by the desired solution [21].

This makes it even more difficult to develop real-time applications, as these ones require fast and efficient algorithms for computing the gait patterns required to achieve dynamical stability. Therefore, finding an analytical solution to the IK problem, or a solution that takes a minimum of iterations, makes it possible to develop controllers able to compensate dynamical effects by changing motion patterns in real-time.

2. SILO2 Humanoid Robot

SILO2 stands for Locomotion System based on two limbs. This humanoid robot belongs to a family of walking prototypes that have been developed in the Department of Automatic Control at the Institute of Industrial Automation in Madrid, Spain [22].

This one in particular is currently on development and it is planned to be upgraded until getting a complete humanoid kinematic structure of 23 dof. However, in the current stage only the locomotor unit has been entirely developed (Figure 1).

As can be seen on this picture, this robot possesses 6 dof per leg: 3 at the hip, 1 at the knee, and 2 at the ankle. That is, each leg's mechanism is determined, so that each foot (or the body, depending on the basement link's election) can reach any desired attitude inside its workspace, defined by geometrical and mechanical constraints.

Joints are driven with two different types of actuators, as can be also noticed: Standard DC servomotors placed at the joint axis and non-linear transmission ratio actuators, used to improve the mechanical design by decreasing the energy consumption. The latter ones, patented as SMART (*Special Mechatronic Actuator for Robot JoinTs*) by the Department of Automatic Control at the Institute of Industrial Automation, are implemented as standard DC servomotors and four-bar linkages which provide the so called non-linear transmission ratio between the input angle driven by the servo and the output angle related to the joint variable.

Actually, three joints on each leg are driven by SMART actuators: One at the hip (abduction / adduction), one at the knee (flexion / extension), and one at the toe (flexion / extension), as can be seen on Figure 2a, 2b, and 2c.

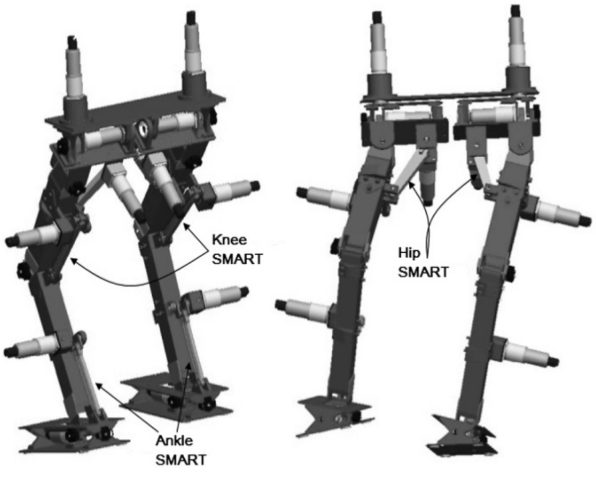


Figure 1. SILO2 Humanoid Robot [23]

3. Humanoid Forward Kinematics

3.1 Kinematic diagram

In order to build the kinematic diagram of this robot it is worth to notice that the presence of SMART actuators does not add complexity to the kinematic model, as could be thought in a first sight. The four-bar linkages are just transmission mechanisms placed between the servomotors and the actual joints, such that the joint variable is indeed a function of the corresponding servomotor's angle.

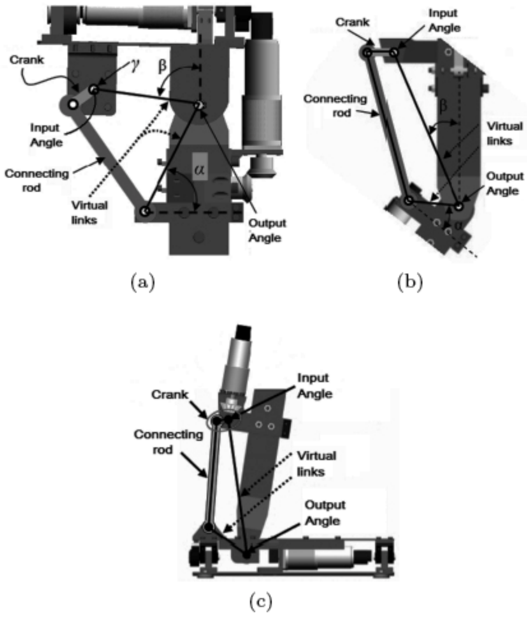


Figure 2. SMART actuators at (a) the hip, (b) the knee, (c) the ankle [23]

Keeping this in mind, the kinematic structure of this robot can be represented with the diagram shown in Figure 3. In this diagram, joints are represented by cylinder-shaped icons, whose arrow represents its positive sense of motion. Each joint is labeled according to the following nomenclature: J_{si}, where

$$\mathrm{s}: \mathrm{l}(\text{left}) \text{ or } \mathrm{r}(\text{right}).$$

i : A number starting at 1 from the joint next to the body frame (base frame) and increasing until reaching the last joint in each kinematic chain.

Joint variables are labeled as θ_{si} according to the corresponding joint J_{si}, such that each leg has a configuration determined by the vector $\boldsymbol{\theta}_{\mathbf{s}} = \{\theta_{\mathrm{s1}}, \theta_{\mathrm{s2}}, \dots, \theta_{\mathrm{s6}}\}$.

Links follow a similar nomenclature, L_{si}, where L_{s0} refers to the base link (the body). In this way, joint J_{si} directly drives link L_{si}.

Link dimensions are displayed along with the corresponding constant denoted by l_{i}, where side information is omitted due to symmetry reasons.

The body frame $\{\mathrm{D}\}$ of the robot is oriented in such a way that the y axis points up and the x axis points in its movement direction. The z axis is then established to complete a right-hand frame.

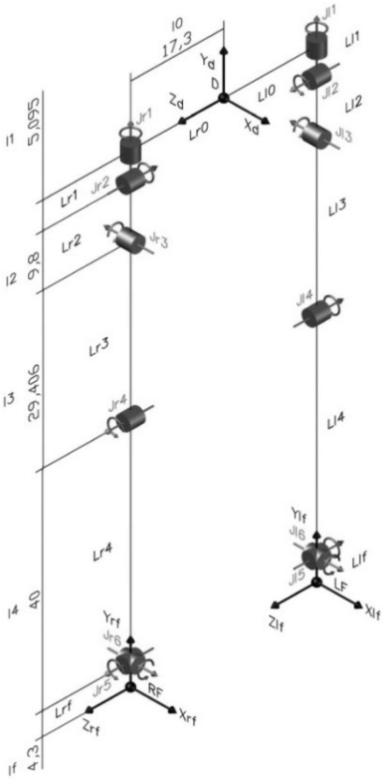


Figure 3. SILO2's kinematic diagram

At the reference attitude $(\boldsymbol{\theta}_{\mathbf{s}} = \mathbf{0})$, the feet frames (referred as $\{\mathrm{S}_{\mathrm{F}}\}$) have the same orientation of the body frame, so that their configuration with respect to the frame $\{\mathrm{D}\}$, $\bar{\mathrm{g}}_{\mathrm{d,sf}}(\mathbf{0})$ [24], can be described as

$$\bar{\mathrm{g}}_{\mathrm{d,sf}}(0) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -(\mathrm{l}_1 + \mathrm{l}_2 + \mathrm{l}_3 + \mathrm{l}_4 + \mathrm{l}_{\mathrm{f}}) \\ 0 & 0 & 1 & \zeta \mathrm{l}_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

Where

$$(\mathrm{s} = \mathrm{r}) \Rightarrow (\zeta = 1) \qquad (\mathrm{s} = \mathrm{l}) \Rightarrow (\zeta = -1) \tag{2}$$

3.2 Screw motion

Forward kinematics can be described in a more geometric way if a screw motion is associated to every joint [24]. A screw motion represents a rotation by an amount of θ about an axis $\mathbf{l}$ followed by a translation by an amount d parallel to that axis, as can be seen on Figure 4. This axis is represented as a directed line ω through a point $\mathbf{q}$; such that $\mathbf{l} = \{\mathbf{q} + \lambda\omega \mid \lambda \in \mathbb{R}\}$. The ratio $h = d/\theta$ is known as the *pitch* of the screw.

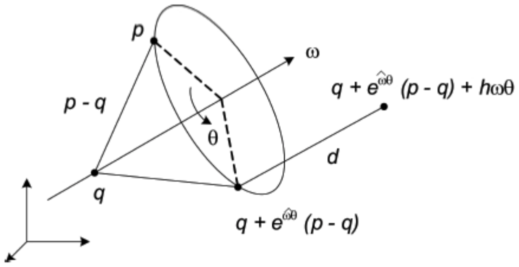

p
ω
$q + e^{\hat{\omega}\theta}(p - q) + h\omega\theta$
p - q
θ
d
q
$q + e^{\hat{\omega}\theta}(p - q)$

Figure 4. Generalized Screw Motion [24]

In this way, if a point $\mathbf{p}$ is operated with the equivalent rigid motion transformation g, its final location (Figure 4) is given by

$$g\mathbf{p} = \mathbf{q} + e^{\hat{\omega}\theta}(\mathbf{p} - \mathbf{q}) + h\theta\omega \tag{3}$$

Where $e^{\hat{\omega}\theta}$ is the exponential matrix of the skew-symmetric matrix $\hat{\omega}$ corresponding to the unit vector ω; that is, $e^{\hat{\omega}\theta}$ is just another way for representing the rotation matrix $R(\omega,\theta)$.

By representing (3) in homogeneous coordinates, we have

$$\bar{g}\bar{\mathbf{p}} = \begin{bmatrix} e^{\hat{\omega}\theta} & \left(I - e^{\hat{\omega}\theta}\right)\mathbf{q} + h\theta\omega \\ 0^{\mathbf{T}} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} \tag{4}$$

The transformation $\bar{g}$ in (4) can also be represented as an exponential matrix, $\bar{g} = e^{\hat{\xi}\theta}$, of a matrix defined as

$$\hat{\xi} = \begin{bmatrix} \hat{\omega} & -\omega \times \mathbf{q} + h\omega \\ 0^{\mathbf{T}} & 1 \end{bmatrix} \tag{5}$$

In this way, a screw motion associated to a rotational joint $(h = 0)$ with joint variable θ is given by

$$e^{\hat{\xi}\theta} = \begin{bmatrix} e^{\hat{\omega}\theta} & \left(I - e^{\hat{\omega}\theta}\right)\mathbf{q} \\ 0^{\mathbf{T}} & 1 \end{bmatrix} \tag{6}$$

3.3 Product of exponentials formula

The transformation $\bar{g}_{d,sf}(\mathbf{0})$ shown in (1) describes the attitude of the corresponding foot with respect to the body frame.

Suppose that we fix every joint except the last one (J_{s6}), which directly affects the foot's attitude, then the new attitude can be calculated as

$$\bar{g}_{d,sf}(\theta_{s6}) = e^{\hat{\xi}_{s6}\theta_{s6}}\bar{g}_{d,sf}(0) \tag{7}$$

In the same way, we can fix every joint except J_{s5}, knowing that the foot's attitude has been modified by the J_{s6}'s action, then

$$\begin{aligned} \bar{g}_{d,sf}(\theta_{s5},\theta_{s6}) &= e^{\hat{\xi}_{s5}\theta_{s5}}\bar{g}_{d,sf}(\theta_{s6}) \\ \bar{g}_{d,sf}(\theta_{s5},\theta_{s6}) &= e^{\hat{\xi}_{s5}\theta_{s5}}e^{\hat{\xi}_{s6}\theta_{s6}}\bar{g}_{d,sf}(0) \end{aligned} \tag{8}$$

By composition, after taking into account every joint in the corresponding kinematic chain the foot configuration becomes

$$\bar{g}_{d,sf}(\boldsymbol{\theta}_{\mathbf{s}}) = e^{\hat{\xi}_{s1}\theta_{s1}}e^{\hat{\xi}_{s2}\theta_{s2}}\ldots e^{\hat{\xi}_{s6}\theta_{s6}}\bar{g}_{d,sf}(0) \tag{9}$$

The expression in (9) is called the *product of exponentials formula* for the humanoid forward kinematics; that is, it describes the foot's attitude as a function of every joint in the kinematic chain [24].

3.4 Forward Kinematics parameters

As can be seen from expressions (9) and (6), in order to calculate the forward kinematics it is necessary to specify each exponential matrix $e^{\hat{\xi}_{si}\theta_{si}}$; that is, apart from the joint variable we need to define each axis direction $\boldsymbol{\omega}_{\mathbf{si}}$ and any point on each axis $\mathbf{q}_{\mathbf{si}}$, as described in the body's reference frame $\{D\}$.

From the kinematic diagram shown in Figure 3, each axis direction is defined as

$$\boldsymbol{\omega}_{\mathbf{s1}} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^{T} \tag{10}$$

$$\boldsymbol{\omega}_{\mathbf{s2}} = \begin{bmatrix} 0 & 0 & -\zeta \end{bmatrix}^{T} \tag{11}$$

$$\boldsymbol{\omega}_{\mathbf{s3}} = \begin{bmatrix} -1 & 0 & 0 \end{bmatrix}^{T} \tag{12}$$

$$\boldsymbol{\omega}_{\mathbf{s4}} = \begin{bmatrix} 0 & 0 & \zeta \end{bmatrix}^{T} \tag{13}$$

$$\boldsymbol{\omega}_{\mathbf{s5}} = \begin{bmatrix} 0 & 0 & \zeta \end{bmatrix}^{T} \tag{14}$$

$$\boldsymbol{\omega}_{\mathbf{s6}} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^{T} \tag{15}$$

And a point on each axis is proposed as

$$\mathbf{q}_{\mathbf{s1}} = \begin{bmatrix} 0 & 0 & \zeta l_0 \end{bmatrix}^{T} \tag{16}$$

$$\mathbf{q}_{\mathbf{s2}} = \begin{bmatrix} 0 & -l_1 & 0 \end{bmatrix}^{T} \tag{17}$$

$$\mathbf{q}_{\mathbf{s3}} = \begin{bmatrix} 0 & -(l_1 + l_2) & \zeta l_0 \end{bmatrix}^{T} \tag{18}$$

$$\mathbf{q}_{\mathbf{s4}} = \begin{bmatrix} 0 & -(l_1 + l_2 + l_3) & 0 \end{bmatrix}^{T} \tag{19}$$

$$\mathbf{q}_{\mathbf{s5}} = \begin{bmatrix} 0 & -(l_1 + l_2 + l_3 + l_4) & 0 \end{bmatrix}^{T} \tag{20}$$

$$\mathbf{q_{s6}} = \begin{bmatrix} 0 & -(l_1+l_2+l_3+l_4) & \zeta l_0 \end{bmatrix}^T \tag{21}$$

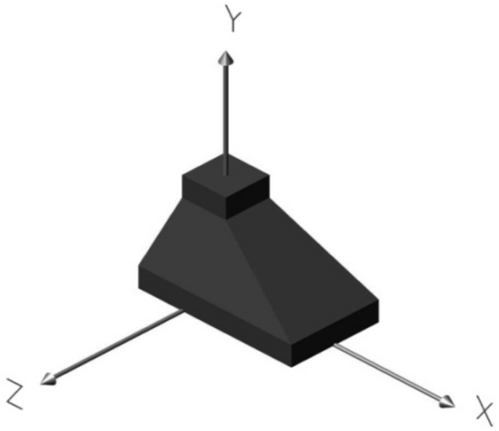


Figure 5. Foot's frame definition

3.5 Orientation coordinates

Consider the representation of the humanoid's foot (and its frame) as shown in Figure 5. The chosen orientation coordinates are three angles, taken as follows: First, consider a rotation of θ_s about the Z axis (*pitch*), followed by a rotation of θ_f about the X axis of the current frame (*roll*). Finally, consider a rotation of θ_t about the Y axis (*yaw*) of the current frame; that is, about a normal vector to the foot's sole. In this way, the resulting rotation matrix R_{foot} is given by

$$\begin{aligned} R_{foot} &= R(Z,\theta_s)R(X,\theta_f)R(Y,\theta_t) \\ R_{foot} &= \begin{bmatrix} c_s c_t - s_s s_f s_t & -s_s c_f & c_s c_t + s_s s_f c_t \\ s_s c_t - c_s s_f s_t & c_s c_f & s_s s_t - c_s s_f c_t \\ -c_f s_t & s_f & c_f c_t \end{bmatrix} \end{aligned} \tag{22}$$

Such that, from an arbitrary rotation matrix these rotation coordinates can be calculated as

$$\theta_f = \arctan 2\left(r_{32}, \pm\sqrt{1-r_{32}^2}\right) \tag{23}$$

$$\theta_s = \arctan 2\left(-r_{12}, r_{22}\right) \tag{24}$$

$$\theta_t = \arctan 2\left(-r_{31}, r_{33}\right) \tag{25}$$

4. Humanoid Inverse Kinematics

By looking at the kinematic diagram shown in Figure 3, it is possible to notice that the implementation of the humanoid's hip is not *spherical*; that is, the first three axes do not intersect in one point. There is an *offset* represented by the link L_{s2} which prevents the usage of the kinematic decoupling method proposed by Pieper [3] to solve the IK problem. In this way, alternative geometric analytical solutions were also sought without success. However, a purely numerical solution is not feasible either due to the existence of a variable efficiency on its implementation.

For this reason, a hybrid approach is proposed. It consists on first finding an approximate analytical "solution" for the inverse kinematics problem by neglecting the presence of the offset at the hip, as it is small when compared to the actual size of the robot, and then to use this "solution" as the initial condition of a numerical method, so that the number of iterations needed be drastically reduced.

This idea is similar to the one proposed in [6]. However, the last one only deals with the case of a double universal joint placed at the wrist. On the other hand, the following procedure is not restricted to any particular kinematic configuration, as long as the offset is small compared to the size of the robot.

4.1 Approximate analytical "solution"

First, suppose that the three axes at the hip intersect in just one point, represented by the actual intersection of the first two axes. Likewise, suppose that the length of the third link is not l_3, but $l_2 + l_3$, in order to diminish the error.

By doing this, a purely analytical "solution" exists, so that it can be found by using the so-called Paden-Kahan methodology [24], which is based on the product of exponentials formula shown in (9) [1] [25].

The equation to be solved is

$$\bar{g}_{d,sf}(\boldsymbol{\theta}) = e^{\hat{\xi}_{s1}\theta_{s1}} e^{\hat{\xi}_{s2}\theta_{s2}} \dots e^{\hat{\xi}_{s6}\theta_{s6}} \bar{g}_{d,sf}(0) := \bar{g}_d \tag{25}$$

$$e^{\hat{\xi}_{s1}\theta_{s1}} e^{\hat{\xi}_{s2}\theta_{s2}} \dots e^{\hat{\xi}_{s6}\theta_{s6}} = \bar{g}_d \bar{g}_{d,sf}^{-1}(0) := \bar{g}_1 \tag{26}$$

To do that, let us propose the following four points based on the reference attitude of the robot shown in Figure 3:

- A point $\mathbf{p_H}$ located at the position of the supposed spherical hip; that is, at the intersection of the axes of the first two joints:

$$\mathbf{p_H} := \begin{bmatrix} 0 & -l_1 & l_0 \end{bmatrix}^T \tag{28}$$

- A point $\mathbf{p_A}$ located at the position of the ankle; that is, at the intersection of the axes of the two last joints:

$$\mathbf{p_A} := \begin{bmatrix} 0 & -(l_1+l_2+l_3+l_4) & l_0 \end{bmatrix}^T \tag{29}$$

- A point $\mathbf{p_L}$ located on the axis of the first joint:

$$\mathbf{p_L} := \begin{bmatrix} 0 & 0 & l_0 \end{bmatrix}^T \tag{30}$$

- A point $\mathbf{p_D}$ located at the origin of the body frame:

$$\mathbf{p_D} := \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \tag{31}$$

Having done this, the joint variables can be calculated as it is explained in the following [24]:

Calculation of θ_{s4} Let us apply both sides of (27) to the point $\mathbf{p_A}$ and then, find the Euclidian distance between the transformation's result and the point $\mathbf{p_H}$. Notice that the point $\mathbf{p_A}$ is located on the axes of the joints J_{s5} and J_{s6}, so that the corresponding operators have no effect on this point and can be omitted. That is,

$$\begin{gathered}\left\| e^{\hat{\xi}_{s1}\theta_{s1}} \ldots e^{\hat{\xi}_{s5}\theta_{s5}} e^{\hat{\xi}_{s6}\theta_{s6}} \mathbf{p_A} - \mathbf{p_H} \right\| = \left\| \overline{g}_1 \mathbf{p_A} - \mathbf{p_H} \right\| \\ \left\| e^{\hat{\xi}_{s1}\theta_{s1}} \ldots e^{\hat{\xi}_{s4}\theta_{s4}} \mathbf{p_A} - \mathbf{p_H} \right\| = \delta\end{gathered} \tag{32}$$

In the same way, given that we have a similar situation at the hip

$$\mathbf{p_H} = e^{\hat{\xi}_{s1}\theta_{s1}} e^{\hat{\xi}_{s2}\theta_{s2}} e^{\hat{\xi}_{s3}\theta_{s3}} \mathbf{p_H} \tag{33}$$

Equation (32) can be rewritten as

$$\left\| e^{\hat{\xi}_{s1}\theta_{s1}} e^{\hat{\xi}_{s2}\theta_{s2}} e^{\hat{\xi}_{s3}\theta_{s3}} \left(e^{\hat{\xi}_{s4}\theta_{s4}} \mathbf{p_A} - \mathbf{p_H} \right) \right\| = \delta \tag{34}$$

Now then, recalling that a rigid body transformation preserves distances between points, the joint variable θ_{s4} can be found by recognizing the equivalent expression as the third Paden-Kahan sub problem (P-K-3) [24], whose solution is already known:

$$\left\| e^{\hat{\xi}_{s4}\theta_{s4}} \mathbf{p_A} - \mathbf{p_H} \right\| = \delta \ \underline{\text{P}-\text{K}-3} \ \theta_{s4} \tag{35}$$

Calculation of θ_{s5} and θ_{s6} Let us reorder (27) and apply both sides to the point $\mathbf{p_H}$, considering the fact shown in (33). In this way,

$$\begin{gathered} e^{\hat{\xi}_{s5}\theta_{s5}} e^{\hat{\xi}_{s6}\theta_{s6}} \mathbf{g}_1^{-1} \mathbf{p_H} = e^{-\hat{\xi}_{s4}\theta_{s4}} e^{-\hat{\xi}_{s3}\theta_{s3}} \ldots e^{-\hat{\xi}_{s1}\theta_{s1}} \mathbf{p_H} \\ e^{\hat{\xi}_{s5}\theta_{s5}} e^{\hat{\xi}_{s6}\theta_{s6}} \mathbf{g}_1^{-1} \mathbf{p_H} = e^{-\hat{\xi}_{s4}\theta_{s4}} \mathbf{p_H} \end{gathered} \tag{36}$$

Now then, as θ_{s4} is already known, it is possible to define the following points:

$$\mathbf{p} := g_1^{-1} \mathbf{p_H} \tag{37}$$

$$\mathbf{q} := e^{-\hat{\xi}_{s4}\theta_{s4}} \mathbf{p_H} \tag{38}$$

So that, both θ_{s5} and θ_{s6} can be found by recognizing the expression in (36) as the second Paden-Kahan sub problem (P-K-2), after the proper substitution of $\mathbf{p}$ and $\mathbf{q}$ [24]; that is,

$$e^{\hat{\xi}_{s5}\theta_{s5}} e^{\hat{\xi}_{s6}\theta_{s6}} \mathbf{p} = \mathbf{q} \ \underline{\text{P}-\text{K}-2} \ \{\theta_{s5}, \theta_{s6}\} \tag{39}$$

Calculation of θ_{s2} and θ_{s3} Let us reorder again (27) and apply both sides to the point $\mathbf{p_L}$. Remember that this point is located on the axis of J_{s1}, so that the joint has no effect on this point. In this way,

$$\begin{gathered} e^{\hat{\xi}_{s2}\theta_{s2}} e^{\hat{\xi}_{s3}\theta_{s3}} e^{\hat{\xi}_{s4}\theta_{s4}} \ldots e^{\hat{\xi}_{s6}\theta_{s6}} \mathbf{g}_1^{-1} \mathbf{p_L} = e^{-\hat{\xi}_{s1}\theta_{s1}} \mathbf{p_L} \\ e^{\hat{\xi}_{s2}\theta_{s2}} e^{\hat{\xi}_{s3}\theta_{s3}} \left(e^{\hat{\xi}_{s4}\theta_{s4}} \ldots e^{\hat{\xi}_{s6}\theta_{s6}} \mathbf{g}_1^{-1} \mathbf{p_L} \right) = \mathbf{p_L} \end{gathered} \tag{40}$$

Now then, given that θ_{s4}, θ_{s5}, and θ_{s6} are known, it is possible to define

$$\mathbf{p} = e^{\hat{\xi}_{s4}\theta_{s4}} e^{\hat{\xi}_{s5}\theta_{s5}} e^{\hat{\xi}_{s6}\theta_{s6}} \mathbf{g}_1^{-1} \mathbf{p_L} \tag{41}$$

In such a way that θ_{s2} y θ_{s3} can be found by recognizing the expression in (40) as the second Paden-Kahan sub problem (P-K-2) again, after the proper substitution of $\mathbf{p}$ [24]; that is,

$$e^{\hat{\xi}_{s2}\theta_{s2}} e^{\hat{\xi}_{s3}\theta_{s3}} \mathbf{p} = \mathbf{p_L} \ \underline{\text{P}-\text{K}-2} \ \{\theta_{s2}, \theta_{s3}\} \tag{42}$$

Calculation of θ_{s1} Reorder again (27) and apply both sides to the point $\mathbf{p_D}$. In this way,

$$e^{\hat{\xi}_{s1}\theta_{s1}} \mathbf{p_D} = g_1 e^{-\hat{\xi}_{s6}\theta_{s6}} \ldots e^{-\hat{\xi}_{s2}\theta_{s2}} \mathbf{p_D} \tag{43}$$

Now then, as all the other angles are already known, it is possible to define

$$\mathbf{q} = g_1 e^{-\hat{\xi}_{s6}\theta_{s6}} \ldots e^{-\hat{\xi}_{s2}\theta_{s2}} \mathbf{p_D} \tag{44}$$

In such a way that θ_{s1} can be finally found by recognizing the expression in (43) as the first Paden-Kahan sub problem (P-K-1), after the substitution of $\mathbf{q}$ [24]; that is,

$$e^{\hat{\xi}_{s1}\theta_{s1}} \mathbf{p_D} = \mathbf{q} \ \underline{\text{P}-\text{K}-1} \ \theta_{s1} \tag{45}$$

It is worth to remark that this procedure may lead to a maximum of eight different joint configurations, due to the multiple solutions of P-K-2 and P-K-3 for equations (35), (39), and (42) (each one has two different solutions). However, it is possible to get rid of four among these ones, as it is natural to pre-select the knee-front configuration, and then to notice that the remaining four are not all valid ones, as not every constraint is considered in every calculation (the procedure required to get rid of information on constraints in order to apply the Paden-Kahan sub-problems).

By choosing the knee-front configuration only two solutions are valid, which can be selected by calculating the four candidate joint configurations and measuring the attitude's error generated by each one. Ideally, the two valid ones must lead to a zero attitude error; but, as this method is based on an approximation, only a minimum error may be achieved. In this way, the joint configuration which lead to lowest error is then selected as the "solution" of the problem.

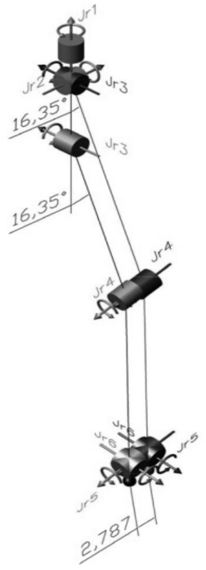


Figure 6. Real kinematic structure (lighter one) vs. supposed one (darker one)

4.2 Numerical refining

The approximate "solution" obtained by supposing a spherical hip generates significant errors that can not be neglected when dealing with gait generation. However, these ones are small compared to the actual size of the robot. This fact can be depicted in Figure 6, where both the real kinematic structure and the supposed one are shown having the same joint values.

As can be seen, for $\theta_{s3} = 16.35°$ (which is actually the joint limit), there is an error of 2.787 cm with respect to the end-effector's position, whereas the robot's size is approximately 1 m. In this way, if such approximate "solution" is used as the initial condition of a numerical method, the algorithm will *likely* converge to the desired solution instead of getting trapped in a local *minima*. Furthermore, the number of iterations needed for accomplishing a negligible error can be drastically reduced in most of the cases. These statements rely in the fact that the convergence of any numerical method almost entirely depends on the selection of the initial values [20].

Numerical algorithms

A numerical algorithm for this problem basically increments (or decrements) the joint configuration $\boldsymbol{\theta}$ by means of an increment vector $\boldsymbol{\Delta\theta}$, in a certain number of iterations; that is, $\boldsymbol{\theta} := \boldsymbol{\theta} + \boldsymbol{\Delta\theta}$ in each one. To compute this $\boldsymbol{\Delta\theta}$ there are many available methods which primarily use the Jacobian J, by taking advantage of the fact that [19]

$$\boldsymbol{\Delta}\mathbf{x} \approx \mathrm{J}\boldsymbol{\Delta\theta} \tag{46}$$

Where $\boldsymbol{\Delta}\mathbf{x}$ is an increment of the end-effector's attitude. One way of doing this is using the Moore-Penrose's pseudoinverse of the Jacobian $\mathrm{J}^{\dagger}$, such that

$$\boldsymbol{\Delta\theta} = \mathrm{J}^{\dagger}\mathbf{e} \tag{47}$$

Where $\mathbf{e}$ is the error vector between the desired attitude and the actual one.

The pseudoinverse provides the "best fit" solution to a system of linear equations that lacks of unique solution. However, it is prone to be unstable in the neighborhood of singularities, as the singular values of $\mathrm{J}^{\dagger}$ are calculated from the inverse of the singular values of J, which are small in the neighborhood of singularities [19].

The Levenberg-Marquardt's method (also known as the damped least squares' method) improves the pseudoinvere's behavior near the singularities, resulting in a stable way for computing $\boldsymbol{\Delta\theta}$. This is done by including a damping factor λ, such that [19]

$$\boldsymbol{\Delta\theta} = \left(\mathrm{J}^{\mathrm{T}}\mathrm{J} + \lambda^2\mathrm{I}\right)^{-1}\mathrm{J}^{\mathrm{T}}\mathrm{e} \tag{48}$$

The reason for such improvement is because the singular values $\mathrm{f_i}$ of the matrix $\left(\mathrm{J}^{\mathrm{T}}\mathrm{J} + \lambda^2\mathrm{I}\right)^{-1}\mathrm{J}^{\mathrm{T}}$ can also be expressed as

$$\mathrm{f_i} = \frac{\sigma_{\mathrm{i}}}{\sigma_{\mathrm{i}}^2 + \lambda^2} \tag{49}$$

Where $\sigma_{\mathrm{i}}, \mathrm{i} = 1,\ldots,\rho\{\mathrm{J}\}$ are singular values of J.

So, given that λ is small, if $\sigma_{\mathrm{i}} \gg \lambda$, then $\frac{\sigma_{\mathrm{i}}}{\sigma_{\mathrm{i}}^2 + \lambda^2} \approx \frac{1}{\sigma_{\mathrm{i}}}$, such that both methods behave in a similar way. But, if $\sigma_{\mathrm{i}} \approx \lambda$, then $\forall \lambda > 0, \frac{\sigma_{\mathrm{i}}}{\sigma_{\mathrm{i}}^2 + \lambda^2} \to 0$ as $\sigma_{\mathrm{i}} \to 0$, behaving in a stable manner [19]. This fact gives us the clue to calculate λ as

$$\lambda = \min\{\sigma_{\mathrm{i}}, \lambda_{\max}\} \tag{50}$$

Attitude's error computation

In order to apply the Levenberg-Marquardt's method it is first necessary to define the error vector $\mathbf{e} \in \mathbb{R}^6$, which is obtained by a concatenation of the positioning error $\mathbf{e_t} \in \mathbb{R}^3$ and the orientation error $\mathbf{e_R} \in \mathbb{R}^3$; that is

$$\mathbf{e} = \begin{bmatrix} \mathbf{e_t}^{\mathrm{T}} & \mathbf{e_R}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{51}$$

Recall that the Jacobian relates differential increments in position and orientation of the end-effector to the differential increments of joint variables. Position's differential increments are effectively vectors with differential magnitude pointing in the movement's direction. In the other hand, orientation's ones are also vectors with differential magnitude, but pointing in the direction of the axis of rotation (unitary angular velocity vector).

Numerical algorithms deal with finite increments (errors), instead of differential ones (as an approximation). In this way, position and orientation errors are vectors with a finite magnitude, that point in the direction of the increment in the case of the former ones or in the direction of the axis of rotation in the case of latter ones.

So, whereas position error's calculation is straightforward $\mathbf{e_t} = \left[\Delta X \ \ \Delta Y \ \ \Delta Z\right]^T$, orientation error's one is not so obvious. This is because the chosen orientation coordinates are not the Cartesian components of a rotational increment vector.

In order to obtain the desired orientation error from the chosen orientation coordinates error, it is necessary first to obtain an expression that relates the rate of change of those coordinates $\left(\dot{\theta}_s, \dot{\theta}_f, \dot{\theta}_t\right)$ with respect to the angular velocity vector $\mathbf{\Omega}$ of the corresponding foot.

Consider the matrix in (22), the correspondent skew-symmetric matrix of $\mathbf{\Omega}$, written as $\widehat{\Omega}$, can be calculated from [24]

$$\widehat{\Omega} = \dot{R}_{foot} R_{foot}^T \tag{52}$$

From which the vector $\mathbf{\Omega}$ can be extracted, so that

$$\mathbf{\Omega} = \begin{bmatrix} 0 & c_s & -s_s c_f \\ 0 & s_s & c_s c_f \\ 1 & 0 & s_f \end{bmatrix} \begin{bmatrix} \dot{\theta}_s \\ \dot{\theta}_f \\ \dot{\theta}_t \end{bmatrix} \tag{53}$$

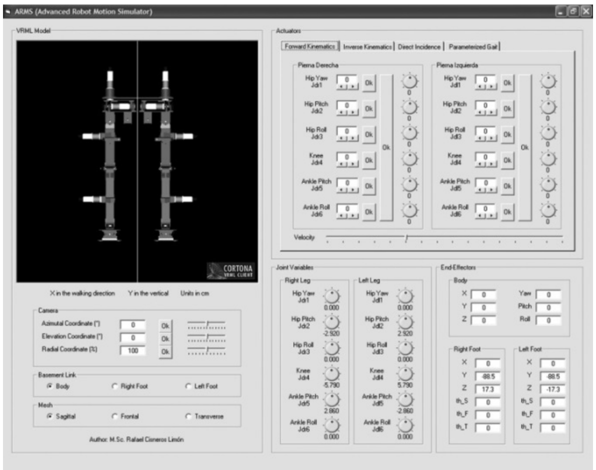
Having done this, the orientation error can be obtained by arbitrarily replacing the orientation coordinates rate of change with the corresponding error to build $\mathbf{e_R}$; that is,

$$\mathbf{e_R} = \begin{bmatrix} 0 & c_s & -s_s c_f \\ 0 & s_s & c_s c_f \\ 1 & 0 & s_f \end{bmatrix} \begin{bmatrix} \Delta\theta_s \\ \Delta\theta_f \\ \Delta\theta_t \end{bmatrix} \tag{54}$$

4.3 Workspace

Let us consider two types of workspace: the ideal one, W_i, defined by geometrical constraints; that is, by considering non-limited joints, and the real one, W_r, which takes into account the joint limits, so that $W_r \subset W_i$.

By having an analytical algorithm for the IK problem, it is possible to check *a priori* if a given attitude is inside the ideal workspace or not, in order to assure that a "solution" exists and avoid failures at the time of calculation.



Piema Derecha: Hip Yaw Jdi1 0; Hip Pitch Jdi2 0; Hip Roll Jdi3 0; Knee Jdi4 0; Ankle Pitch Jdi5 0; Ankle Roll Jdi6 0

Piema Izquierda: Hip Yaw Jdl1 0; Hip Pitch Jdl2 0; Hip Roll Jdl3 0; Knee Jdl4 0; Ankle Pitch Jdl5 0; Ankle Roll Jdl6 0

X in the walking direction Y in the vertical Units in cm

Camera: Azimutal Coordinate (°) 0; Elevation Coordinate (°) 0; Radial Coordinate (%) 100

Basement Link: ● Body ○ Right Foot ○ Left Foot

Mesh: ● Sagittal ○ Frontal ○ Transverse

Author: M.Sc. Rafael Cisneros Limón

Joint Variables

Right Leg: Hip Yaw Jdi1 0.000; Hip Pitch Jdi2 -2.920; Hip Roll Jdi3 0.000; Knee Jdi4 -5.790; Ankle Pitch Jdi5 2.860; Ankle Roll Jdi6 0.000

Left Leg: Hip Yaw Jdl1 0.000; Hip Pitch Jdl2 2.920; Hip Roll Jdl3 0.000; Knee Jdl4 5.790; Ankle Pitch Jdl5 -2.860; Ankle Roll Jdl6 0.000

End-Effectors

Body: X 0; Y 0; Z 0; Yaw 0; Pitch 0; Roll 0

Right Foot: X 0; Y -88.5; Z 17.3; th_S 0; th_F 0; th_T 0

Left Foot: X 0; Y -88.5; Z -17.3; th_S 0; th_F 0; th_T 0

Figure 7. ARMS Kinematic Simulator

In this case, an approximate workspace can be found by noticing that only the solution for P-K-3 considers a restricted-domain function, represented by the arccosine, $\arccos(.) \ : \left[-1,1\right] \rightarrow \left[0,\pi\right]$. So that if an unreachable attitude is specified, the argument of the arccosine function $\left(D\right)$ will be a number outside its domain. In this way, if $-1 < D < 1$ an approximate analytical "solution" exists.

5. Simulation results

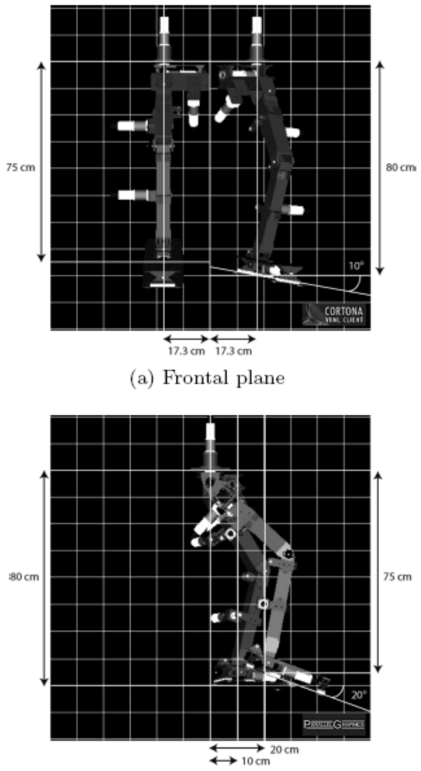
In order to visually validate the algorithm proposed in this paper, a kinematic simulator was developed under the name of ARMS (*Advanced Robot Motion Simulator*) (Figure 7) [7].

This simulator uses a VRML (*Virtual Reality Modeling Language*) model of the humanoid robot, which is driven by specifying its joint coordinates. These ones are directly specified by the user, or calculated from a desired attitude which is specified in task space.

For testing purposes a series of attitudes were specified for each foot, by considering the body as the base link. Some attitudes were proposed as lying inside the real workspace and others, outside, or as lying only inside the ideal workspace but not inside the real one, in order to test the algorithm's behavior. One example of each case is reported, along with some screenshots of the corresponding results.

5.1 Attitude inside the real workspace

Figure 8 shows the attitude reached by the model of the robot (frontal and sagittal views) after specifying that the right foot's sole should be located 75 cm below with respect to the pelvis and 20 cm just in front of it, while maintaining the reference attitude's distance between this foot and the sagittal plane of symmetry (17.3 cm). The right foot's orientation is also specified with a *pitch* angle of −20□ In the other hand, the location of the left foot's sole is specified to be 80 cm below with respect to the pelvis, 10 cm in front of it and 17.3 cm away from the sagittal plane of symmetry, while its orientation is specified with a *roll* angle of −10°, followed by a *yaw* angle of 30°.

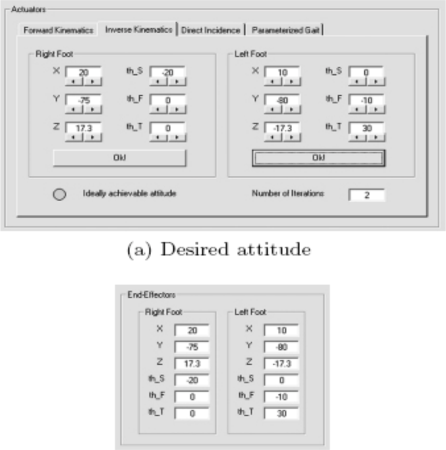


(a) Frontal plane

(b) Sagittal plane

Figure 8. Resulting configuration after specifying an attitude inside the real workspace

As can be seen with the aid of superimposed labels and the screenshot of the specified attitude and the obtained one (Figure 9) the virtual robot not only reaches the desired goal, but it can be achieved with only 2 iterations.



Right Foot

X 20 th_S -20

Y -75 th_F 0

Z 17.3 th_T 0

Left Foot

X 10 th_S 0

Y -80 th_F -10

Z -17.3 th_T 30

○ Ideally achievable attitude

Number of Iterations 2

(a) Desired attitude

End-Effectors

Right Foot		Left Foot	
X	20	X	10
Y	-75	Y	-80
Z	17.3	Z	-17.3
th_S	-20	th_S	0
th_F	0	th_F	-10
th_T	0	th_T	30

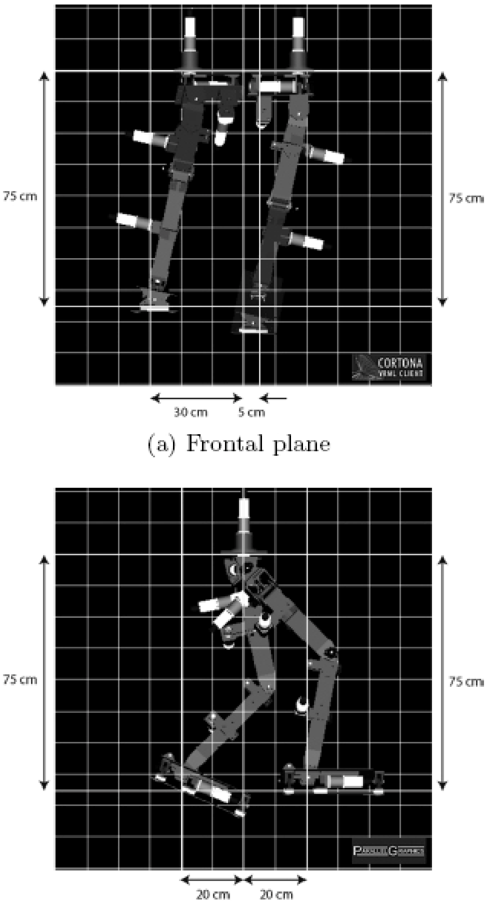
(b) Obtained attitude

Figure 9. Control and feedback panels associated With Figure 8

5.2 Attitude inside the ideal workspace, but outside the real one defined by mechanical constraints

Figure 10 shows again an attitude reached by the model of the robot. In this case, the right foot's sole was specified to be located 75 cm below with respect to the pelvis and 20 cm just in front of it as before, but now it is also desired a distance of 30 cm between this foot and the sagittal plane of symmetry. In the other hand, the left foot's location is specified to be a the same height but 20 cm behind the pelvis, at a distance of 5 cm from the sagittal plane of symmetry. Both feet are expected to maintain the reference orientation; that is, 0° for all the orientation coordinates.

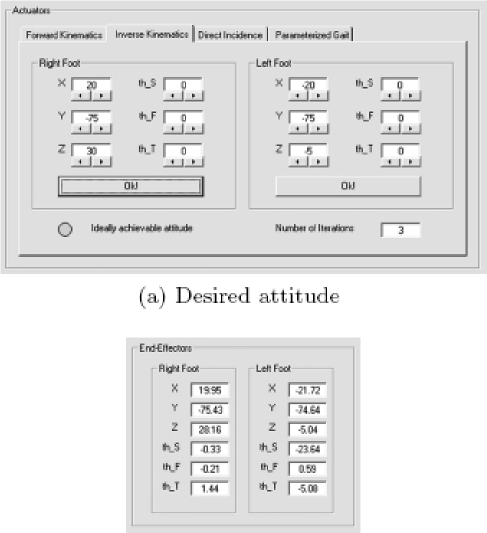
However, the desired attitude cannot be reached as it can be confirmed by noticing that the right foot's sole is a little below of the desired one (marked with the aid of superimposed labels), and because the orientation of the left foot is not the reference one. This can also be confirmed in Figure 11, where the difference between the desired attitude and the obtained one is noticeable, due to the saturation of joint's values. The process needed 3 iterations in this case.



(a) Frontal plane

(b) Sagittal plane

Figure 10. Resulting configuration after specifying an attitude inside the ideal workspace, but outside the real one



Actuators

Right Foot: X 20, Y -75, Z 30, th_S 0, th_F 0, th_T 0

Left Foot: X -20, Y -75, Z -5, th_S 0, th_F 0, th_T 0

○ Ideally achievable attitude

Number of Iterations: 3

(a) Desired attitude

End-Effectors

Right Foot: X 19.95, Y -75.43, Z 28.16, th_S -0.33, th_F -0.21, th_T 1.44

Left Foot: X -21.72, Y -74.64, Z -5.04, th_S -23.64, th_F 0.59, th_T -5.08

(b) Obtained attitude

Figure 11. Control and feedback panels associated with Figure 10

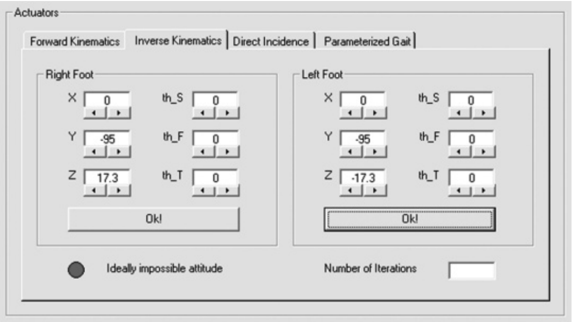
5.3 *Attitude outside the ideal workspace*

Consider now the case when a desired attitude is outside the ideal workspace, represented by the case when the feet are specified to reach a distance impossible to achieve, even at full extension of the legs. In this case the kinematic simulator performs no action at all, as indicated by the label "*Ideally impossible attitude*" (Figure 12).

6. Evaluation

The performance of the proposed hybrid algorithm was evaluated with respect to the number of iterations needed to accomplish a desired precision by means of statistical comparison with a purely numerical solution, as well as with respect to the mean time required for performing the requested computations in both cases, as a way to measure the achieved improvement of performance.

This evaluation was carried out over different sets of data consisting on 1000 random 6-tuples of joint variables $\{\theta_{s1}, \theta_{s2}, \ldots, \theta_{s6}\}$ each one. The leg was also chosen in a random manner.



Actuators

Right Foot: X 0, Y -95, Z 17.3, th_S 0, th_F 0, th_T 0

Left Foot: X 0, Y -95, Z -17.3, th_S 0, th_F 0, th_T 0

● Ideally impossible attitude

Number of Iterations:

Figure 12. The attitude is not inside the ideal workspace

Each 6-tuple, along with the leg's choice, was used to obtain the corresponding foot's attitude by means of applying the forward kinematics algorithm. The resulting attitude was then taken as the input of the IK algorithm. In this way, it was assured that this input would lead to a solution, which was already known for testing and comparison purposes.

When testing the proposed algorithm, the foot's attitude was used as the input of the approximate analytical algorithm in such a way that the approximate "solution" was taken as the initial condition of the Levenberg-Marquardt method.

When testing the purely numerical solution, the initial condition of the Levenberg-Marquardt method was set arbitrarily as $\boldsymbol{\theta}_{\mathrm{s}} = 0$.

In both cases, an error vector magnitude less than 0.1 or a maximum of 1500 iterations were used as stop conditions. The maximum value for the parameter $\lambda_{\max}$ was also set to 0.01 (as this value led to the best performance through a set of tests). Then, the error between the desired attitude and the obtained one was computed and recorded, as well as the number of iterations required and the mean computation time.

This evaluation consisted on two different statistical populations described as follows:

1. The first one considered randomly generated attitudes inside the ideal workspace, but not necessarily inside the real one. These ones were obtained by constraining the random joint variables' values to −90□and 90□.
2. The second one considered randomly generated attitudes only inside the real workspace. In this case, the joint variables' values were constrained to the actual joint limits of the robot (Table 1).

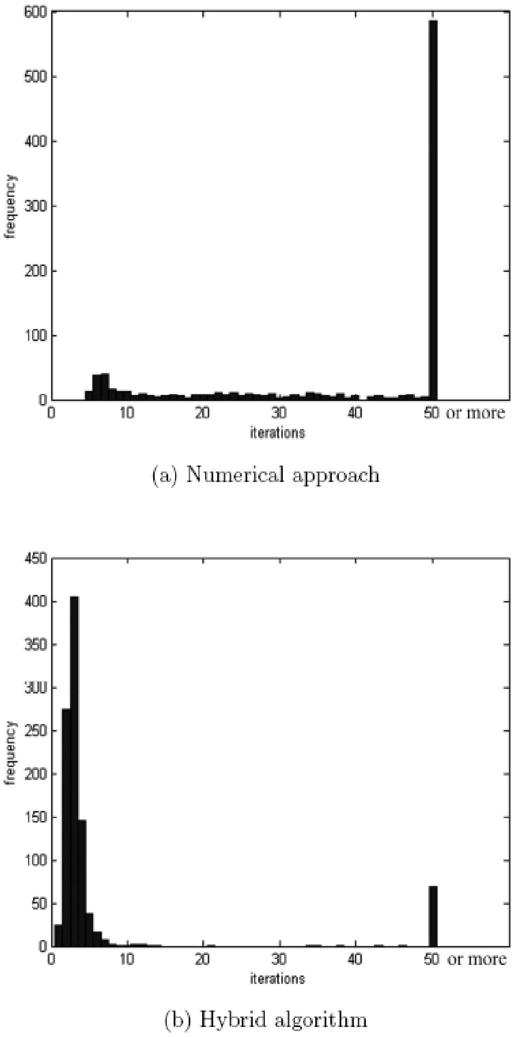
Joint		Range of motion
Hip rotation	Left Right	$-180° \le \theta_{l1} \le 15°$ $-15° \le \theta_{r1} \le 180°$
Hip flexion / extension	Left Right	$-70° \le \theta_{l2} \le 70°$ $-70° \le \theta_{r2} \le 70°$
Hip abduction / adduction	Left Right	$73° \le \theta_{l3} \le 106°$ $-106° \le \theta_{r3} \le 73°$
Knee	Left Right	$-64° \le \theta_{l4} \le -3°$ $-64° \le \theta_{r4} \le -3°$
Ankle flexion / extension	Left Right	$-15° \le \theta_{l5} \le 20°$ $-15° \le \theta_{r5} \le 20°$
Ankle abduction / adduction	Left Right	$-12° \le \theta_{l6} \le 12°$ $-12° \le \theta_{r6} \le 12°$

Table 1. Range of motion of each joint of SILO2 [23]

This evaluation tests were implemented in Matlab R2012a, and performed on a Laptop Dell XPS L702X, which has an Intel(R) Core(TM) i7-2670QM CPU @ 2.20 GHz, 8 GB in RAM, and a 64-bit Operating System (Windows 7 Utimate).

6.1 Random attitudes generated inside the ideal workspace, but not necessarily inside the real one

Figure 13a shows the histogram generated by the purely numerical approach, whereas Figure 13b shows the one generated by the proposed hybrid algorithm, obtained by constraining the random joint variables' values to −90□ and 90□. In both cases, the histogram's range is set to 1 - 50 or more iterations (samples over 50 iterations are accumulated over 50 in the histogram).



(a) Numerical approach

(b) Hybrid algorithm

Figure 13. Computation efficiency comparison between (a) a purely numerical approach and (b) the proposed hybrid method, considering randomly-generated attitudes inside the *ideal* workspace

Table 2 also shows some numerical data associated with the evaluation; that is, the percentage of samples that succeeded to converge (within the specified precision) in 1500, 50, and 10 iterations, as well as the average error in position and orientation between the desired attitude and the obtained one (taking into account only the samples that succeeded to converge) and the mean time required per sample in both cases.

Indicator	**Purely numerical**	**Hybrid solution**
Convergence in 1500 it.	97.5%	99.5 %
Convergence in 50 it.	39.8%	92.6 %
Convergence in 10 it.	13.5%	91.4 %
Mean position error	0.0176 cm	0.0011 cm
Mean orientation error	0.0284°	0.0023°
Mean CPU time	71.663 ms	17.057 ms

Table 2. Performance comparison of both algorithms considering randomly-generated attitudes inside the ideal workspace

As can be seen, the proposed algorithm shows an outstanding performance over the purely numerical one used in [19] or [21], as more than 91.4% of the samples are able to converge in 10 or less iterations, in contrast with the latter one which may need 50, 100, 1000 or more iterations to converge. In addition, the proposed algorithm is able to achieve a lower average error (with respect to position or orientation) while being almost 4 times faster.

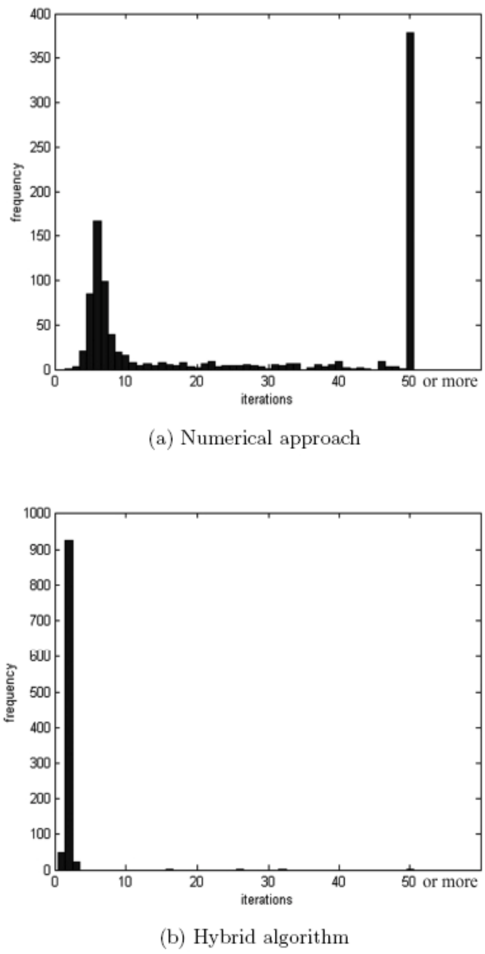
6.2 Random attitudes generated inside the real workspace only

By taking into account the actual joint limits of the robot, both algorithms were compared once again. Figure 14a shows the histogram generated by the purely numerical approach, whereas Figure 14b shows the one generated by the proposed hybrid algorithm. The histogram's range is set again to 1 - 50 or more iterations. Table 2 shows the corresponding numerical data associated with this evaluation.

Surprisingly, the proposed algorithm not only shows an outstanding performance over the purely numerical one, but a very good one feasible for implementation on real-time, as 99.5% of the samples converged in less than 10 iterations (in fact, 97.2% did it within 2 iterations). Notice that the average error is also lower than the previous case, and that the hybrid algorithm performs almost 5 times faster.

Indicator	Purely numerical	Hybrid solution
Convergence in 1500 it.	99.3%	99.9%
Convergence in 50 it.	61.8%	99.8 %
Convergence in 10 it.	45.0%	99.5 %
Mean position error	0.0025 cm	0.0006 cm
Mean orientation error	0.0042°	0.0002°
Mean CPU time	41.046 ms	7.762 ms

Table 3. Performance comparison of both algorithms considering randomly-generated attitudes inside the real workspace



(a) Numerical approach

(b) Hybrid algorithm

Figure 14. Computation efficiency comparison between (a) a purely numerical approach and (b) the proposed hybrid method, considering randomly-generated attitudes inside the *real* workspace

7. Conclusions

This paper proposed a semi-analytical algorithm for computing the IK of humanoid robots whose construction shows a small but non negligible offset at the hip (compared to its actual height) which prevents any purely analytical solution to be developed.

As a case of study, a robot with a non-typical kinematic structure, as it is the SILO2 prototype, is analyzed. However, the algorithm can be applied to any structure that possesses negligible offsets, so that an approximate analytical "solution" can be found after neglecting them, and then be used as the initial condition of a stable numerical method, ensuring in this way that the number of iterations needed will be drastically reduced (compared with the required ones if an arbitrary initial condition is proposed).

Besides that, as the approximate analytical "solution" is in the neighborhood of the desired target in the joint configuration space, it will be more likely that the numerical method reaches the desired solution, as the right minima is very near to the initial condition. So, this method will be as efficient as desired so long as the concerned offsets are small enough. This fact was also evaluated, as this algorithm needed only very few iterations:

Less than 10 in most of the cases, or a maximum of 2 if the desired attitudes belonged to a typical gait; that is, attitudes that do not require big joint ranges to be performed. However, even non-typical motions can be achieved in an affordable computation time.

Concerning to the SILO2 robot, this method effectively solved its IK problem, so that it was possible to calculate precise gaits in real-time, as they were required by the control laws that had been previously developed by the team work at the Institute of Industrial Automation. Furthermore, the kinematic simulator developed during this work represented a useful tool for testing desired attitudes on a virtual model before doing it on the real prototype, avoiding in this way failures that may cause severe damages to the robot if it falls down. The simulator was very useful to test attitudes in this way, as many of them cannot be realized due to mechanical constraints, even when they were feasible during a certain gait. By recognizing these attitudes, the gait could be modified until getting a realizable and useful one, or they may also be used to provide guidelines for modifying the SMART actuators, by redesigning the size of the crank and the connecting road in every four-link mechanism, in order to improve the design of the humanoid.

8. References

[1] Mario Arbulú, José M. Pardos, Luis Cabas, Pavel Staroverov, Dmitry Kaynov, Carlos Pérez, Miguel Rodríguez, and Carlos Balaguer. Rh-0 humanoid full size robot's control strategy based on the Lie logic technique. In *Proceedings of the 5th. IEEE-RAS International Conference on Humanoid Robots*, pages 271–276, Tsukuba, Japan, December 2005.

[2] Gaurav Tevatia and Stefan Schaal. Inverse kinematics for humanoid robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 294– 299, California, USA, April 2000.

[3] D.L. Pieper. *The Kinematics of Manipulators under Computer Control*. PhD thesis, Stanford University, California, USA, 1968.

[4] Mark W. Spong and M. Vidyasagar. *Robot dynamics and control*, chapter 3. John Wiley & Sons, USA, 1989.

[5] Muhammad A. Ali, H. Andy Park, and C.S. George Lee. Closed-form inverse kinematic joint solution for humanoid robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 704– 709, Taipei, Taiwan, October 2010.

[6] Robert L. Williams II. Inverse kinematics and singularities of manipulators with offset wrist. *IASTED International Journal of Robotics and Automation*, 14(1):1–8, 1999.

[7] Juan Manuel Ibarra and Rafael Cisneros. Forward and inverse kinematics for a small-sized humanoid robot. In *Proceedings of the 19th. IEEE International Conference on Electronics, Communications and Computers (CONIELECOMP)*, pages 111 – 118, Cholula, Puebla, Mexico, February 2009.

[8] Mohammed Zeki Hussain Al-Faiz and Mohammed S. Saleh. Inverse kinematics analysis for manipulator robot with wrist offset based on the closed-form algorithm. *International Journal of Robotics and Automation (IJRA)*, 2(4):256–264, September / October 2011.

[9] Giresh K. Singh and Jonathan Claassens. An analytical solution for the inverse kinematics of a redundant 7dof manipulator with link offsets. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2976–2982, Taipei, Taiwan, December 2010.

[10] Michael Mistryl, Iun Nakanishi, Gordon Cheng, and Stefan Schaal. Inverse kinematics with floating base and constraints for full body humanoid robot control. In *Proceedings of the 8th. IEEE-RAS International Conference on Humanoid Robots*, pages 22–27, Daejeon, South Korea, February 2009.

[11] Li-Chun Tommy Wang and Chih Cheng Chen. A combined optimization method for solving the inverse kinematics problem of mechanical manipulators. *IEEE Transactions on Robotics and Automation*, 7(4):489–499, August 1991.

[12] Adrian A. Canutescu and Roland L. Dunbrack Jr. Cyclic coordinate descent: A robotics algorithm for protein loop closure. *Protein Science*, 12(5):963–972, May 2003.

[13] R. Müller Cajar and R. Mukundan. Triangulation: A new algorithm for inverse kinematics. In *Proceedings of the 22th. International Image and Vision Computing New Zealand Conference (IVCNZ)*, pages 181–186, Hamilton, New Zealand, December 2007.

[14] A. Pashkevich. Real-time inverse kinematics for robots with offset and reduced wrist. *Control Engineering Practice*, 5(10):1443–1450, October 1997.

[15] Javier de Lope, Telmo Zarraonandia, Rafaela González Careaga, and Darío Maravall. Solving the inverse kinematics in humanoid robots: A neural approach. In *Proceedings of the 7th. International Work-Conference on Artificial and Natural Neural Networks (IWANN), LNCS 2687*, pages 177–184, Menorca, Spain, June 2003. Springer-Verlag, Berlin Heidelberg.

[16] Z. Bingul, H.M. Ertunc, and C. Oysu. Comparison of inverse kinematics solutions using neural network for 6r robot manipulator with offset. In *Proceedings of the ICSC Congress on Computational Intelligence Methods and Applications*, Istanbul, Turkey, December 2005.

[17] Joachim Pitt, Dietmar Hildebrand, Maximilian Stelzer, and Andreas Koch. Inverse kinematics of a humanoid robot based on conformal geometric algebra using optimized code generation. In *Proceedings of the 8th. IEEE-RAS International Conference of Humanoid Robots*, pages 681–686, Daejeon, South Korea, December 2008.

[18] Dinesh Manocha and John F. Canny. Efficient inverse kinematics for general 6r manipulators. *IEEE Transactions on Robotics and Automation*, 10(5):648–657, October 1994.

[19] Samuel R. Buss. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. Technical report, University of California, California, USA, April 2004. Unpublished (available via the Internet).

[20] Patrick J. Parikh and Sarah S. Y. Lam. A hybrid strategy to solve the forward kinematics problem in parallel manipulators. *IEEE Transactions on Robotics*, 21(1):18–25, February 2005.

[21] Tomomichi Sugihara. Solvability-unconcerned inverse kinematics by the levenberg-marquardt method. *IEEE Transactions on Robotics*, 27(5):984–991, October 2011.

[22] M. Armada, R. Caballero, T. Akinfiev, H. Montes, C. Manzano, L. Pedraza, S. Ríos, and P. González De Santos. Design of SILO2 humanoid robot. In *Proceedings of the 3rd. IARP International Workshop on Humanoid and Human Friendly Robotics*, pages 37–42, Tsukuba, Japan, December 2002.

[23] Héctor Montes Franceschi. *Análisis, Diseño y Evaluación de Estrategias de Control de Fuerza en Robots Caminantes*. PhD thesis, Universidad Complutense de Madrid, Madrid, Spain, 2006.

[24] Richard M. Murray, Zexiang Li, and S. Shankar Sastry. *A Mathematical Introduction to Robotic Manipulation*, chapter 2–3. CRC Press LLC, USA, 1994.

[25] Deepak Tolani, Ambarish Goswami, and Norman I. Badler. Real-time inverse kinematics techniques for anthropomorphic limbs. *Graphical Models*, 62(5):353–388, September 2000.