



# Automatic generation of humanoid's geometric model parameters

Vincent Hugel, Nicolas Jouandeau

## ► To cite this version:

Vincent Hugel, Nicolas Jouandeau. Automatic generation of humanoid's geometric model parameters. 17th annual RoboCup International Symposium, Jul 2013, Eindhoven, Netherlands. 10.1007/978-3-662-44468-9\_36 . hal-02317131

**HAL Id: hal-02317131**

**<https://hal.science/hal-02317131>**

Submitted on 15 Oct 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Automatic generation of humanoid's geometric model parameters

Vincent Hugel and Nicolas Jouandeau

LISV, University of Versailles and LIASD, University of Paris 8

**Abstract.** This paper describes a procedure that automatically generates parameters for the geometric modeling of kinematic chains. The convention of modeling used is the Denavit Hartenberg convention modified by Khalil Kleinfinger, noted DHKK. The procedure proposed here has two advantages. First the user does not need to calculate the geometric parameters by himself. He simply has to give the directions of the successive joint axes, and for each joint axis, a point that belongs to the axis. The second advantage deals with the use of model-generic matrices for the beginning and the end of the kinematic chains, and not only for the joint axes. This prevents the user from doing specific calculation to connect the joint matrices derived from the model with the initial and the final coordinate frames of the chain. Due to its unified formalism, the procedure allows to save time when the kinematics of the robot has to be changed. This paper includes the application of this procedure to the geometric modeling of legs and arms of two versions of the NAO humanoid robot, the one used in the RoboCup 3D Simulation League, and the other one used in the RoboCup Standard Platform League.

## 1 Introduction

Usually the calculation of the geometric model of a kinematic chain starts with the selection of a convention of parameterization to define the successive coordinate frames to go from one extremity to the other extremity of the chain. The user has to master the convention rules and has to manually calculate distances and angles necessary for the parameterization [1]. This is often a tedious task that requires a certain amount of time for the calculation itself, the checking and the validation with simulations. The task is even harder when there are multiple kinematic chains. If we take the example of a humanoid robot, there are five kinematic chains, the head, the arms and the legs. The number of parameters to determine is at least equal to the total number of degrees of freedom of the robot.

This paper proposes an automatization procedure for determining the parameters related to the modeling convention. This procedure simplifies the job of the robotics engineer since he does not have to do any calculation any more. It is straightforward and can be applied to any robot model or kinematic chain. It can be useful at the design stage and at the simulation stage to modify the dimensions of the bodies of the kinematic chain, the layout and the directions

of the joint axes [2] [3], or the order of the joints without having to calculate all parameters manually.

To our knowledge such an automatization of the calculation of geometric model parameters has never been reported.

## 2 Modeling convention

Khalil and Kleininger [4] proposed a modified convention for geometric modeling from the Denavit-Hartenberg convention [5]. The first advantage of this modified convention is its suitability for the kinematic description of both open-loop and closed-loop structures. The second advantage lies in the convenient definition of the  $z_i$  axis as the axis of rotary joint  $i$ . The angle of rotation about axis  $z_i$  is denoted by  $\theta_i$ , with same index  $i$ . This rotation is the fourth transformation<sup>1</sup> among the four successive transformations needed to go from coordinate frame  $i-1$  linked to joint  $i-1$  to coordinate frame  $i$  linked to joint  $i$ . The first three transformations can be used to position the rotation axis  $z_i$ .

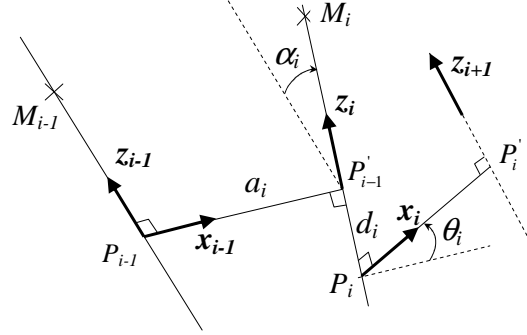


Fig. 1. DHKK convention parameters.

There are four DHKK parameters required to go from  $R_{i-1}$  to  $R_i$ , one for each transformation. Parameters are denoted by  $a_i$ ,  $\alpha_i$ ,  $d_i$  and  $\theta_i$ .  $a_i$  and  $d_i$  are distances,  $\alpha_i$  and  $\theta_i$  are angles. Figure 1 shows the four parameters at stage  $i$ . They involve the three axes  $z_{i-1}$ ,  $z_i$ , and  $z_{i+1}$ .

- $a_i$  is the signed distance  $P_{i-1}P'_{i-1}$  along  $x_{i-1}$ .
- $\alpha_i$  is the rotation angle about  $x_{i-1}$  between  $z_{i-1}$  and  $z_i$ .
- $d_i$  is the signed distance  $P'_{i-1}P_i$  along  $z_i$  axis.
- $\theta_i$  is the rotation angle about  $z_i$  between  $x_{i-1}$  and  $x_i$ .

<sup>1</sup> each notation involves four successive transformations, twice one translation followed by one rotation.

where  $\mathbf{x}_{i-1}$  is along the segment that is orthogonal to  $\mathbf{z}_{i-1}$  and  $\mathbf{z}_i$  axes, from  $\mathbf{z}_{i-1}$  to  $\mathbf{z}_i$ .  $\mathbf{x}_i$  is along the segment that is orthogonal to  $\mathbf{z}_i$  and  $\mathbf{z}_{i+1}$  axes, from  $\mathbf{z}_i$  to  $\mathbf{z}_{i+1}$ .  $P_i$  is the intersection point of  $\mathbf{x}_i$  with  $\mathbf{z}_i$ .  $P'_i$  is the intersection point of  $\mathbf{x}_i$  with  $\mathbf{z}_{i+1}$ .  $M_i$  is a point that belongs to  $\mathbf{z}_i$  axis.

The transformation from  $R_i$  to  $R_{i-1}$  is written as follows, where  $R$  stands for rotation and  $T$  for translation ( $R_i = (P_i, \mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i)$ ):

$${}^i_{i-1}M = T_{\mathbf{x}_{i-1}}(a_i)R_{\mathbf{x}_{i-1}}(\alpha_i)T_{\mathbf{z}_i}(d_i)R_{\mathbf{z}_i}(\theta_i) \quad (1)$$

$${}^i_{i-1}M = \begin{pmatrix} c\theta_i & -s\theta_i & 0 & a_i \\ c\alpha_i s\theta_i & c\alpha_i c\theta_i & -s\alpha_i & -d_i s\alpha_i \\ s\alpha_i s\theta_i & s\alpha_i c\theta_i & c\alpha_i & d_i c\alpha_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2)$$

### 3 Generic procedure for the calculation of DHKK parameters

To determine the DHKK parameters, it is necessary to choose an initial configuration for the kinematic chain. In this configuration the user has to give the coordinates of the joint axes, i.e. the coordinates of  $(\mathbf{z}_i)$  vectors relative to an external reference frame linked to a body of reference like the trunk in a humanoid robot. The user must also give the coordinates of points that belong to the joint axes, one point for each axis. These points are the  $(M_i)$  points. Usually  $(M_i)$  are points located at the joint axis intersections, for example hip center, knee center, or ankle center for a humanoid leg.

The first step of the procedure consists of calculating the  $(\mathbf{x}_i)$  vectors.  $\mathbf{x}_i$  must be orthogonal to  $\mathbf{z}_i$  and  $\mathbf{z}_{i+1}$ . In the general case where  $(M_i, \mathbf{z}_i)$  is not parallel with  $(M_{i+1}, \mathbf{z}_{i+1})$ , we can define  $\mathbf{x}_i$  from the cross-product of  $\mathbf{z}_i$  with  $\mathbf{z}_{i+1}$ :

$$\mathbf{x}_i = \frac{\mathbf{z}_i \times \mathbf{z}_{i+1}}{\|\mathbf{z}_i \times \mathbf{z}_{i+1}\|}$$

There are two particular cases. The first one involves  $(M_i, \mathbf{z}_i) = (M_{i+1}, \mathbf{z}_{i+1})$ . Here we choose to keep the same vector as previously defined, i.e.  $\mathbf{x}_i = \mathbf{x}_{i-1}$ . The second case implies that  $(M_i, \mathbf{z}_i)$  is parallel but not equal to  $(M_{i+1}, \mathbf{z}_{i+1})$ . Here we can use  $\mathbf{M}_i \mathbf{M}_{i+1}$  to obtain the direction orthogonal to both parallel axes inside the plane containing these axes:

$$\frac{\mathbf{z}_i \times \mathbf{M}_i \mathbf{M}_{i+1}}{\|\mathbf{z}_i \times \mathbf{M}_i \mathbf{M}_{i+1}\|} \times \mathbf{z}_i$$

In summary:

$$\begin{aligned} (M_i, \mathbf{z}_i) = (M_{i+1}, \mathbf{z}_{i+1}) &\Rightarrow \mathbf{x}_i = \mathbf{x}_{i-1} \\ (M_i, \mathbf{z}_i) \parallel (M_{i+1}, \mathbf{z}_{i+1}) &\Rightarrow \mathbf{x}_i = \frac{\mathbf{z}_i \times \mathbf{M}_i \mathbf{M}_{i+1}}{\|\mathbf{z}_i \times \mathbf{M}_i \mathbf{M}_{i+1}\|} \times \mathbf{z}_i \\ (M_i, \mathbf{z}_i) \nparallel (M_{i+1}, \mathbf{z}_{i+1}) &\Rightarrow \mathbf{x}_i = \frac{\mathbf{z}_i \times \mathbf{z}_{i+1}}{\|\mathbf{z}_i \times \mathbf{z}_{i+1}\|} \end{aligned}$$

Then the three DHKK parameters are defined<sup>2</sup> as:

$$a_i = \mathbf{M}_{i-1} \mathbf{M}_i \cdot \mathbf{x}_{i-1} \quad (3)$$

$$\alpha_i = \text{atan2}([\mathbf{z}_{i-1}, \mathbf{z}_i, \mathbf{x}_{i-1}], \mathbf{z}_{i-1} \cdot \mathbf{z}_i) \quad (4)$$

$$\theta_i = \text{atan2}([\mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{z}_i], \mathbf{x}_{i-1} \cdot \mathbf{x}_i) + \Delta\theta_i \quad (5)$$

where  $\Delta\theta_i$  represents the command angle of joint  $i$  from the initial configuration.

The last DHKK parameter is expressed as:

$$d_i = \mathbf{P}'_{i-1} \mathbf{P}_i \cdot \mathbf{z}_i \quad (6)$$

with:

$$\begin{aligned} \mathbf{P}_i &= \mathbf{M}_i + r_i \cdot \mathbf{z}_i \\ \mathbf{P}'_i &= \mathbf{M}_{i+1} + s_i \cdot \mathbf{z}_{i+1} \end{aligned}$$

To calculate  $r_i$  and  $s_i$ , we use the following property of the convention:

$$\mathbf{P}_i \mathbf{P}'_i \times (\mathbf{z}_i \times \mathbf{z}_{i+1}) = 0$$

This equation is useful only in the case where  $\mathbf{z}_i \times \mathbf{z}_{i+1} \neq \mathbf{0}$ . Taking the scalar product of this equation with  $\mathbf{z}_i$  for the one part, and with  $\mathbf{z}_{i+1}$  for the other part, gives:

$$\begin{aligned} [\mathbf{P}_i \mathbf{P}'_i, \mathbf{z}_i \times \mathbf{z}_{i+1}, \mathbf{z}_i] &= 0 \\ [\mathbf{P}_i \mathbf{P}'_i, \mathbf{z}_i \times \mathbf{z}_{i+1}, \mathbf{z}_{i+1}] &= 0 \end{aligned}$$

After replacement:

$$\begin{aligned} [\mathbf{M}_i \mathbf{M}_{i+1} + s_i \cdot \mathbf{z}_{i+1}, \mathbf{z}_i \times \mathbf{z}_{i+1}, \mathbf{z}_i] &= 0 \Leftrightarrow \\ s_i \cdot \|\mathbf{z}_i \times \mathbf{z}_{i+1}\|^2 &= [\mathbf{M}_i \mathbf{M}_{i+1}, \mathbf{z}_i, \mathbf{z}_i \times \mathbf{z}_{i+1}] \\ [\mathbf{M}_i \mathbf{M}_{i+1} - r_i \cdot \mathbf{z}_i, \mathbf{z}_i \times \mathbf{z}_{i+1}, \mathbf{z}_{i+1}] &= 0 \Leftrightarrow \\ r_i \cdot \|\mathbf{z}_i \times \mathbf{z}_{i+1}\|^2 &= [\mathbf{M}_i \mathbf{M}_{i+1}, \mathbf{z}_{i+1}, \mathbf{z}_i \times \mathbf{z}_{i+1}] \end{aligned}$$

In the case where  $\mathbf{z}_i \times \mathbf{z}_{i+1} = \mathbf{0}$ , we choose to place  $\mathbf{P}_i$  at  $\mathbf{M}_i$ , which means that  $r_i$  is set to 0. Then it comes  $s_i = \mathbf{M}_{i+1} \mathbf{M}_i \cdot \mathbf{z}_{i+1}$ .

In summary,

$$\begin{aligned} \mathbf{z}_i \parallel \mathbf{z}_{i+1} &\Rightarrow r_i = 0 \\ s_i &= \mathbf{M}_{i+1} \mathbf{M}_i \cdot \mathbf{z}_{i+1} \\ \mathbf{z}_i \not\parallel \mathbf{z}_{i+1} &\Rightarrow r_i = [\mathbf{M}_i \mathbf{M}_{i+1}, \mathbf{z}_{i+1}, \mathbf{z}_i \times \mathbf{z}_{i+1}] / \|\mathbf{z}_i \times \mathbf{z}_{i+1}\|^2 \\ s_i &= [\mathbf{M}_i \mathbf{M}_{i+1}, \mathbf{z}_i, \mathbf{z}_i \times \mathbf{z}_{i+1}] / \|\mathbf{z}_i \times \mathbf{z}_{i+1}\|^2 \end{aligned}$$

---

<sup>2</sup>  $[\mathbf{a}, \mathbf{b}, \mathbf{c}]$  denotes the triple scalar product  $(\mathbf{a} \times \mathbf{b}) \cdot \mathbf{c}$ .

## 4 Application to NAO humanoids

The NAO robot (Fig. 2) manufactured by the French company Aldebaran-Robotics has a coupled yaw-pitch joint with a  $45[deg]$  inclination with respect to the vertical [6]. Table 1 gives the offsets and lengths related to two versions of NAO, i.e. the version used in the 3D Soccer Simulation League (3D-SSL) and the version used in the Standard Platform League (SPL). The trunk center named  $T$  is taken as a reference. The coordinate frame linked to the trunk is  $R_T = (T, \mathbf{i}_T, \mathbf{j}_T, \mathbf{k}_T)$ , with  $\mathbf{i}_T$  being the longitudinal axis,  $\mathbf{j}_T$  the lateral axis, and  $\mathbf{k}_T$  the vertical axis. Figure 3 illustrates the notations used.

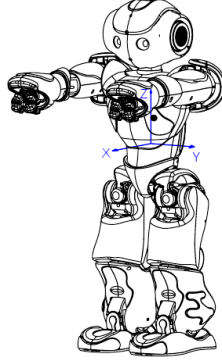
**Table 1.** Leg and arm lengths and offsets of NAO models [m]

| Offset/Length          | 3D-SSL leg | SPL leg | Offset/Length              | 3D-SSL arm | SPL arm |
|------------------------|------------|---------|----------------------------|------------|---------|
| Hip Offset Z: $H_z^o$  | 0.115      | 0.085   | Shoulder Offset Z: $S_z^o$ | 0.075      | 0.1     |
| Hip Offset Y: $H_y^o$  | 0.055      | 0.05    | Shoulder Offset Y: $S_y^o$ | 0.098      | 0.098   |
| Hip Offset X: $H_x^o$  | 0.01       | 0.0     | Elbow Offset Y: $E_y^o$    | 0.0        | 0.015   |
| Knee Offset X: $K_x^o$ | 0.005      | 0.0     | Elbow Offset Z: $E_z^o$    | 0.009      | 0.0     |
| Femur length: $L_f$    | 0.12       | 0.1     | Arm length: $L_a$          | 0.09       | 0.105   |
| Tibia length: $L_t$    | 0.1        | 0.1029  | Forearm length: $L_{fa}$   | 0.105      | 0.13    |
| Foot Height: $F_z^h$   | 0.05       | 0.04519 |                            |            |         |

### 4.1 Leg geometric model

**Table 2.** Input parameters for the NAO leg DHKK kinematics modeling

| Point $M_i$ belonging to joint axis $\mathbf{z}_i$                 | Joint axis $\mathbf{z}_i$                        |
|--|--|
| $M_{s1} = T = [0, 0, 0]^T$   | $\mathbf{z}_{s1} = \mathbf{k}_T = [0, 0, 1]^T$   |
| $M_{s2} = T = [0, 0, 0]^T$   | $\mathbf{z}_{s2} = \mathbf{k}_T = [0, 0, 1]^T$   |
| $M_1 = H = [-H_x^o, -\xi.H_y^o, -H_z^o]^T$                         | $\mathbf{z}_1 = [0, \cos(45), \xi \sin(45)]^T$   |
| $M_2 = H = [-H_x^o, -\xi.H_y^o, -H_z^o]^T$                         | $\mathbf{z}_2 = \mathbf{i}_T = [1, 0, 0]^T$      |
| $M_3 = H = [-H_x^o, -\xi.H_y^o, -H_z^o]^T$                         | $\mathbf{z}_3 = \mathbf{j}_T = [0, 1, 0]^T$      |
| $M_4 = K = [-K_x^o, -\xi.H_y^o, -H_z^o - L_f]^T$                   | $\mathbf{z}_4 = \mathbf{j}_T = [0, 1, 0]^T$      |
| $M_5 = A = [-K_x^o, -\xi.H_y^o, -H_z^o - L_f - L_t]^T$             | $\mathbf{z}_5 = \mathbf{j}_T = [0, 1, 0]^T$      |
| $M_6 = A = [-K_x^o, -\xi.H_y^o, -H_z^o - L_f - L_t]^T$             | $\mathbf{z}_6 = \mathbf{i}_T = [1, 0, 0]^T$      |
| $M_{e1} = Ah = [-K_x^o, -\xi.H_y^o, -H_z^o - L_f - L_t - F_z^h]^T$ | $\mathbf{z}_{e1} = \mathbf{k}_T = [0, 0, 1]^T$   |
| $M_{e2} = Ah = [-K_x^o, -\xi.H_y^o, -H_z^o - L_f - L_t - F_z^h]^T$ | $\mathbf{z}_{e2} = -\mathbf{j}_T = [0, -1, 0]^T$ |



**Fig. 2.** Nao schematics in reference configuration where all joint angles are equal to zero.

Table 2 contains the inputs needed for the DHKK-parameter procedure. It includes the joint axes and points that belong to them. Figure 3 shows the points  $H$  (hip),  $K$  (knee),  $A$  (ankle) and  $Ah$  (ankle projection) of each leg, and the joint axes numbered 1 to 6. All these points and axes are given for legs in the reference position, i.e. completely stretched vertically.  $\xi = 1$  for the right leg and  $-1$  for the left leg.

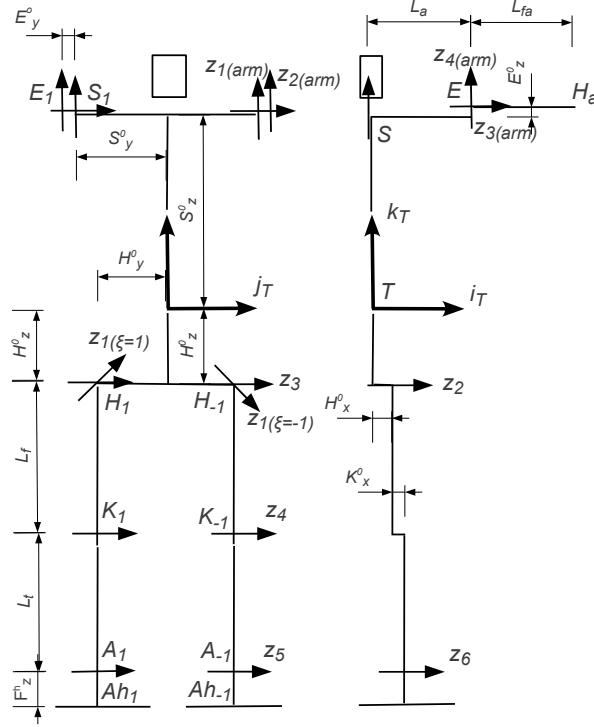
The first two axes  $\mathbf{z}_{s1}$ <sup>3</sup> and  $\mathbf{z}_{s2}$  are used to go from  $R_T = (T, \mathbf{i}_T, \mathbf{j}_T, \mathbf{k}_T)$  to the first coordinate frame centered at the hip, and therefore take into account the translation from  $T$  to  $H$ . Axes  $\mathbf{z}_i$  refer to the joint rotation axes of angle  $\theta_i$ . Joints are ordered from hip to ankle as hip-yaw, hip roll, hip pitch, knee pitch, ankle pitch and ankle roll. The last two axes  $\mathbf{z}_{e1}$ <sup>4</sup> and  $\mathbf{z}_{e2}$  are used to orient the last coordinate frame so that it matches the axes of  $R_T$ . It is centered at  $Ah$  which is the projection of the ankle center on the foot sole.

The introduction of two axes at the beginning and at the end of the kinematic chain prevents the user from making connections with the initial and the final coordinate frames. These additional axes are not *joint* axes but are also considered as z-axes in order to apply the same DHKK formalism. Actually each pair of axes – at the beginning for the one part and at the end for the other part – allows to apply an additional DHKK homogeneous matrix at the beginning, and at the end of the kinematic chain.

Once points and axes are given as inputs, the procedure can be applied to get the DHKK parameters of the kinematic chain. Table 3 gives the different

<sup>3</sup> s stands for start.

<sup>4</sup> e stands for end.



**Fig. 3.** Humanoid skeleton with notations. Frontal view on the left-hand side and sagittal view on the right-hand side.

values for the kinematic parameters that are related to the NAO leg. Angles  $\Delta\theta$  are the angles related to joint moves. They can represent the joint commands or the joint sensor readings. They are zero in the reference position (legs vertically stretched and arms horizontally stretched).

## 4.2 Arm geometric model

Table 4 contains the inputs needed for the DHKK-parameter procedure. It includes the joint axes and the points that belong to them. Figure 3 shows the points  $S$  (shoulder),  $E$  (elbow), and  $H_a$  (hand/wrist) of each arm, and the joint axes numbered 1 to 4. All these points and axes are given for arms in the reference position, i.e. completely stretched horizontally.  $\xi = 1$  for the right arm and  $-1$  for the left arm.

The first two axes  $z_{s1}$ <sup>5</sup> and  $z_{s2}$  are used to go from  $R_T = (T, i_T, j_T, k_T)$  to the first coordinate frame centered at the shoulder, and therefore take into account the translation from  $T$  to  $S$ . Joints are ordered from shoulder to hand

<sup>5</sup> s stands for start.



**Table 3.** DHKK parameters for the NAO Humanoid leg kinematics model

| i | $a_i[m]$                           | $\alpha_i[deg]$                  | $d_i[m]$                          | $\theta_i[deg]$  |
|---|------------------------------------|----------------------------------|-----------------------------------|--|
| s | 0                                  | 0                                | $-(H_z^o - H_y^o)$                | 180  |
| 1 | $H_x^o$                            | $45(1 + \xi)/2 + 135(1 - \xi)/2$ | $-\xi \cdot \sqrt{2} \cdot H_y^o$ | $-90.0 + \Delta\theta_{yaw-pitch}^{hip}$                                 |
| 2 | 0                                  | 90                               | 0                                 | $135\xi + \Delta\theta_{roll}^{hip}$                                     |
| 3 | 0                                  | 90                               | 0                                 | $\pi - \tan^{-1} \frac{H_x^o - K_x^o}{L_f} + \Delta\theta_{pitch}^{hip}$ |
| 4 | $\sqrt{L_f^2 + (H_x^o - K_x^o)^2}$ | 0                                | 0                                 | $\tan^{-1} \frac{H_x^o - K_x^o}{L_f} + \Delta\theta_{pitch}^{knee}$      |
| 5 | $L_t$                              | 0                                | 0                                 | $\Delta\theta_{pitch}^{ankle}$   |
| 6 | 0                                  | 90                               | 0                                 | $-90 + \Delta\theta_{roll}^{ankle}$                                      |
| e | 0                                  | 90                               | $-F_z^h$                          | 90   |

as shoulder pitch, shoulder yaw, elbow roll and elbow yaw. The last two axes  $\mathbf{z}_{e1}$ <sup>6</sup> and  $\mathbf{z}_{e2}$  are used to orient the last coordinate frame so that it matches the axes of  $R_T$ . The last two points are used to set the origin at the extremity of the arm  $H_a$ .

**Table 4.** Input parameters for the NAO arm DHKK kinematics modeling

| Point $M_i$ belonging to joint axis $\mathbf{z}_i$                           | joint axis $\mathbf{z}_i$         |
|--|-----------------------------------|
| $M_{s1} = T = [0, 0, 0]^T$   | $\mathbf{z}_{s1} = \mathbf{k}_T$  |
| $M_{s2} = T = [0, 0, 0]^T$   | $\mathbf{z}_{s2} = \mathbf{k}_T$  |
| $M_1 = S = [0, -\xi \cdot S_y^o, S_z^o]^T$                                   | $\mathbf{z}_1 = \mathbf{j}_T$     |
| $M_2 = S = [0, -\xi \cdot S_y^o, S_z^o]^T$                                   | $\mathbf{z}_2 = \mathbf{k}_T$     |
| $M_3 = E = [L_a, -\xi \cdot (S_y^o + E_y^o), S_z^o + E_z^o]^T$               | $\mathbf{z}_3 = \mathbf{i}_T$     |
| $M_4 = E = [L_a, -\xi \cdot (S_y^o + E_y^o), S_z^o + E_z^o]^T$               | $\mathbf{z}_4 = \mathbf{k}_T$     |
| $M_{e1} = H_a = [L_a + L_{fa}, -\xi \cdot (S_y^o + E_y^o), S_z^o + E_z^o]^T$ | $\mathbf{z}_{e1} = \mathbf{k}_T$  |
| $M_{e2} = H_a = [L_a + L_{fa}, -\xi \cdot (S_y^o + E_y^o), S_z^o + E_z^o]^T$ | $\mathbf{z}_{e2} = -\mathbf{j}_T$ |

Table 5 gives the different values for the kinematic parameters that are related to the NAO arm.

## 5 Inverse Geometric Modeling

The inverse geometric modeling consists of finding analytical solutions for joint angles  $\theta_i$  given the trajectory of the end effector in the Cartesian space. The product of DHKK homogeneous matrices to go from the end effector coordinate

<sup>6</sup> e stands for end.

**Table 5.** DHKK parameters for the NAO Humanoid arm kinematics model

| i | $a_i[m]$     | $\alpha_i[deg]$ | $d_i[m]$    | $\theta_i[deg]$                         |
|---|--------------|-----------------|-------------|---|
| s | 0            | 0               | $S_z^o$     | 180                                     |
| 1 | 0            | 90              | $\xi.S_y^o$ | $180 + \Delta\theta_{pitch}^{shoulder}$ |
| 2 | 0            | 90              | $E_z^o$     | $90 + \Delta\theta_{yaw}^{shoulder}$    |
| 3 | $-\xi E_y^o$ | 90              | $L_a$       | $180 + \Delta\theta_{roll}^{elbow}$     |
| 4 | 0            | 90              | 0           | $90 + \Delta\theta_{yaw}^{elbow}$       |
| e | $L_{fa}$     | 0               | 0           | 0                                       |

frame to the torso coordinate frame can be used to express the relationship between the position/orientation of the end effector with the joint displacements.

Considering the NAO's humanoid leg, we can write the geometric model as:

$$M^0 = M^s \cdot M^{\theta_1} \cdot M^{\theta_2} \cdot M^{\theta_3} \cdot M^{\theta_4} \cdot M^{\theta_5} \cdot M^{\theta_6} \cdot M^e \quad (7)$$

where

- $M^s$  involves the first three  $z$  vectors,
- $M^e$  involves the last three  $z$  vectors,
- $M^{\theta_i}$  involves joint rotation of angle  $\theta_i$ , and vectors  $z_{i-1}$ ,  $z_i$ ,  $z_{i+1}$ ,
- $M^0$  is the homogeneous matrix given by the user, that represents the wanted trajectory (rotation + translation) of the end effector (position of the projection of ankle on the sole and orientation of the sole).

Joint angles  $\theta_i$  are the unknowns and, to invert the direct geometric model, we can use the following notation:

$$M^x = \begin{pmatrix} R_x & T_x \\ 0 & 1 \end{pmatrix}$$

$T_0$  is the position of the end effector in the torso coordinate frame.  $R_0$  is the matrix whose columns contain the coordinates of the end effector frame expressed in torso coordinate frame.  $R_0$  can be interpreted as the rotation matrix of the end effector frame inside torso frame. It can also be interpreted as the transformation matrix to pass from the end effector coordinate frame into the torso coordinate frame.

Equation 7 can be developed as:

$$\begin{aligned}
\begin{pmatrix} R_s & T_s \\ 0 & 1 \end{pmatrix} \begin{pmatrix} R_{123} & T_1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} R_4 & T_4 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} R_{56} & T_5 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} R_e & T_e \\ 0 & 1 \end{pmatrix} &= \begin{pmatrix} R_0 & T_0 \\ 0 & 1 \end{pmatrix} \\
&\Leftrightarrow \begin{pmatrix} R_{123} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} R_{456} & T_4 + R_4 T_5 \\ 0 & 1 \end{pmatrix} = \\
&\begin{pmatrix} {}^t R_s & -{}^t R_s T_s - T_1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} R_0({}^t R_e) & T_0 - R_0({}^t R_e) T_e \\ 0 & 1 \end{pmatrix} \\
&\Leftrightarrow \begin{pmatrix} R_{123} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} R_{456} & T_4 + R_4 T_5 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} R' & M' \\ 0 & 1 \end{pmatrix}
\end{aligned}$$

with

$$\begin{aligned}
R' &= ({}^t R_s) R_0 ({}^t R_e) \\
T' &= -T_1 + ({}^t R_s)(T_0 - T_s - R_0({}^t R_e) T_e)
\end{aligned}$$

which gives the following system to solve:

$$R_{123}(T_4 + R_4 T_5) = T' \quad (8)$$

$$R_{123} R_{456} = R' \quad (9)$$

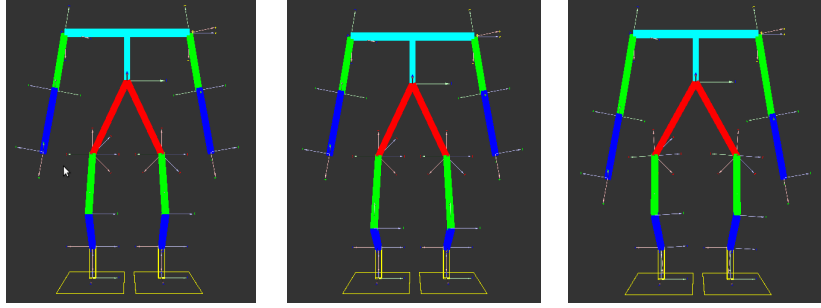
Squaring the first equation gives  $(T_4 + R_4 T_5)^2 = T'^2$  that allows to solve for the knee angle  $\theta_4$  and then  $\Delta\theta_4$ . Then it is possible to solve either for the hip angles then for the ankle angles, or for the ankle angles then the hip angles. The solving for the hip or the ankle angles is traditional.

## 6 Results

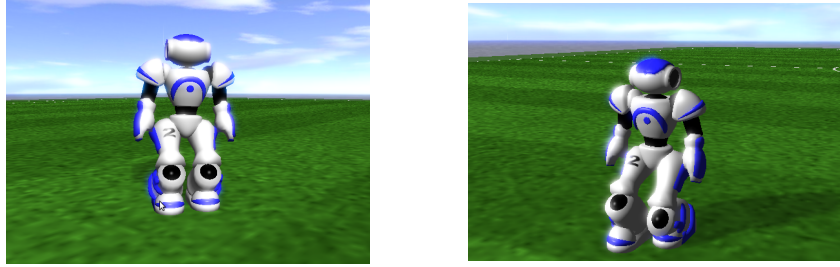
The procedure was implemented with success for the participation in the 3D-SSL of the 2013 RoboCup German Open. The DHKK parameters are calculated at the initialization stage when the C++ instance of the robot's model class is built. The procedure does not modify the computing time required for the generation of joint commands.

DHKK parameters are used when the direct or the inverse geometric model is needed. The direct geometric model is used at the validation stage to display the robot's skeleton (Fig. 4). The validation stage allows checking the movements of the different limbs of the robot.

The inverse geometric model is used in real time in the 3D-SSL software (SimSpark application for the RoboCup 3D-SSL [7] [8], namely *rcssserver3d*) to calculate the joint command angles to be sent to the server. Walking patterns were designed using the model of the 3D linear inverted pendulum and the Zero Moment Point technique [9]. Figure 5 shows two snapshots of the NAO robot in the 3D-SSL environment. The left-hand side picture shows the robot in standing position with flexed knees after initialization of the model's parameters. The right-hand side picture shows the robot making a forward step.



**Fig. 4.** Stick models of the 3D-SSL version of NAO displayed in the qt-opengl environment for the validation stage. The model on the left-hand side is related to the default model with the femur length equal to  $0.12[m]$ . The model in the middle is related to a change of the femur length, that is  $0.14[m]$  instead of  $0.12[m]$ . The model on the right-hand side features a greater distance between hips –  $0.065[m]$  instead of  $0.055[m]$ , and longer arms –  $0.13[m]$  instead of  $0.09[m]$ . DHKK parameters are calculated automatically at the init stage, and are used in the direct geometric model for the display.



**Fig. 5.** Snapshots of the 3D-SSL NAO robot in standing position and making a step forward, achieved with the 3D-SSL official simulation server [7] [8] combined with the RoboViz monitor [10].

## 7 Conclusion

This paper presented a procedure to automatically generate parameters for the geometric modeling of kinematic chains. The convention of modeling used is the Denavit Hartenberg convention modified by Khalil Kleinfinger. This procedure can be used for all kinds of kinematic chains and especially for NAO humanoid models. The user must choose a reference position and give couples  $(M_i, z_i)$  that define the joint rotation axes. Then in the initialization stage, the procedure calculates the DHKK model parameters according to the reference position. These parameters are fixed and only depend on the reference position of the chains. They can be used for direct geometric modeling but also for inverse modeling to calculate the joint command angles in real time. The procedure can deal with the entire kinematic chain to switch from a starting coordinate frame to an end coordinate frame and vice-versa.

## References

1. Graf, C. and Hartl, A. and Röfer, T. and Laue, T. (2009) “A Robust Closed-Loop Gait for the Standard Platform League Humanoid”. *Proc. 4th Workshop on Humanoid Soccer Robots*. pp. 30–37.
2. Zorjan, M. and Hugel, V. and Blazevic, P. (2011) “Influence of hip joint axes change of orientation on power distribution in humanoid motion”. *Proc. IEEE ICARA: The 5th Int. Conf. on Automation, Robots and Applications*. pp. 271–276.
3. Zorjan, M. and Hugel, V. (2013). “Generalized Humanoid Leg Inverse Kinematics to Deal With Singularities”, *Proc. IEEE Int. Conf. on Robotics and Automation*.
4. Khalil, W. and Kleininger, J-F. (1986) “A new geometric notation for open and closed-loop robots,” *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 1174–1180.
5. Denavit, J. and Hartenberg, R.S. (1955) “A kinematic notation for lower-pair mechanisms based on matrices,” *Trans ASME J. Appl. Mech.* 23 pp. 215–221.
6. Gouaillier, D. and Hugel, V. and Blazevic, P. and Kilner, C. and Monceaux, J. and Lafourcade, P. and Marnier, B. and Serre, J. and Maisonnier, B. (2009) “Mechatronic design of NAO humanoid”. *Proc. IEEE Int. Conf. on Robotics and Automation*. pp. 769–774.
7. Obst, O. and Rollmann, M. (2004) “Spark - A Generic Simulator for Physical Multi-Agent Simulations”, *Multiagent System Technologies, Lecture Notes in Computer Science*, vol. 3187, pp. 243–257.
8. SimSpark, a generic physical multiagent simulator system for agents in three-dimensional environments. <http://simspark.sourceforge.net/>.
9. Hugel, V. and Jouandeau, N. (2012) “Walking patterns for real time path planning simulation of humanoids”, *Proc. IEEE Int. Symp. on Robot and Human Interactive Comm. (RO-MAN)*, pp. 424–430.
10. Stoecker, J. and Visser, U. (2012) “RoboViz: Programmable Visualization for Simulated Soccer”, *RoboCup 2011: Robot Soccer World Cup XV*, vol. 7416, *Lecture Notes in Computer Science*. pp. 282–293.