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 Kleinfinger [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 θ_i, with same index i. This rotation is the fourth transformation[1] 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.
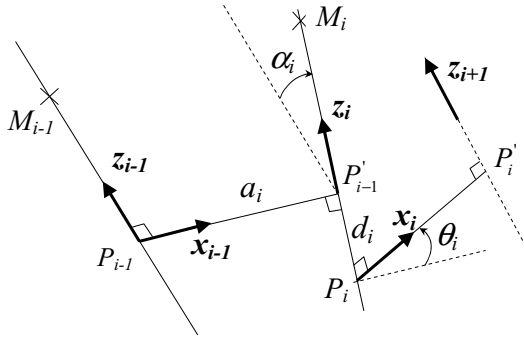

Fig. 1. DHKK convention parameters

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, α_i, d_i and θ_i. a_i and d_i are distances, α_i and θ_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}.
- α_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.
- θ_i is the rotation angle about z_i between x_{i-1} and x_i.

---

[1] Each notation involves four successive transformations, twice one translation followed by one rotation.

where x_{i-1} is along the segment that is orthogonal to z_{i-1} and z_i axes, from z_{i-1} to z_i. x_i is along the segment that is orthogonal to z_i and z_{i+1} axes, from z_i to z_{i+1}. P_i is the intersection point of x_i with z_i. P'_i is the intersection point of x_i with z_{i+1}. M_i is a point that belongs to 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 $(R_i = (P_i, x_i, y_i, z_i))$:

$$\substack{i\\i-1}M = T_{x_{i-1}}(a_i)R_{x_{i-1}}(\alpha_i)T_{z_i}(d_i)R_{z_i}(\theta_i) \tag{1}$$

$$\substack{i\\i-1}M = \begin{pmatrix} c\theta_i & -s\theta_i & 0 & a_i \\ c\alpha_i s\theta_i & c\alpha_i c\theta_i & -s\alpha_i & -d_i s\alpha_i \\ s\alpha_i s\theta_i & s\alpha_i c\theta_i & c\alpha_i & d_i c\alpha_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2}$$

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 (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 (x_i) vectors. x_i must be orthogonal to z_i and z_{i+1}. In the general case where (M_i, z_i) is not parallel with (M_{i+1}, z_{i+1}), we can define x_i from the cross-product of z_i with z_{i+1}:

$$x_i = \frac{z_i \times z_{i+1}}{\|z_i \times z_{i+1}\|}$$

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

$$\frac{z_i \times M_i M_{i+1}}{\|z_i \times M_i M_{i+1}\|} \times z_i$$

In summary:

$$(M_i, z_i) = (M_{i+1}, z_{i+1}) \Rightarrow x_i = x_{i-1}$$
$$(M_i, z_i) \parallel (M_{i+1}, z_{i+1}) \Rightarrow x_i = \frac{z_i \times M_i M_{i+1}}{\|z_i \times M_i M_{i+1}\|} \times z_i$$
$$(M_i, z_i) \nparallel (M_{i+1}, z_{i+1}) \Rightarrow x_i = \frac{z_i \times z_{i+1}}{\|z_i \times z_{i+1}\|}$$

Then the three DHKK parameters are defined[2] as:

$$a_i = \boldsymbol{M_{i-1}M_i}.\boldsymbol{x_{i-1}} \tag{3}$$

$$\alpha_i = atan2\left(\left[\boldsymbol{z_{i-1}}, \boldsymbol{z_i}, \boldsymbol{x_{i-1}}\right], \boldsymbol{z_{i-1}}.\boldsymbol{z_i}\right) \tag{4}$$

$$\theta_i = atan2\left(\left[\boldsymbol{x_{i-1}}, \boldsymbol{x_i}, \boldsymbol{z_i}\right], \boldsymbol{x_{i-1}}.\boldsymbol{x_i}\right) + \Delta\theta_i \tag{5}$$

where $\Delta\theta_i$ represents the command angle of joint i from the initial configuration.
The last DHKK parameter is expressed as:

$$d_i = \boldsymbol{P'_{i-1}P_i}.\boldsymbol{z_i} \tag{6}$$

with:

$$P_i = M_i + r_i.\boldsymbol{z_i}$$
$$P'_i = M_{i+1} + s_i.\boldsymbol{z_{i+1}}$$

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

$$\boldsymbol{P_iP'_i} \times (\boldsymbol{z_i} \times \boldsymbol{z_{i+1}}) = 0$$

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

$$\left[\boldsymbol{P_iP'_i}, \boldsymbol{z_i} \times \boldsymbol{z_{i+1}}, \boldsymbol{z_i}\right] = 0$$
$$\left[\boldsymbol{P_iP'_i}, \boldsymbol{z_i} \times \boldsymbol{z_{i+1}}, \boldsymbol{z_{i+1}}\right] = 0$$

After replacement:

$$\left[\boldsymbol{M_iM_{i+1}} + s_i.\boldsymbol{z_{i+1}}, \boldsymbol{z_i} \times \boldsymbol{z_{i+1}}, \boldsymbol{z_i}\right] = 0 \Leftrightarrow$$
$$s_i.\|\boldsymbol{z_i} \times \boldsymbol{z_{i+1}}\|^2 = \left[\boldsymbol{M_iM_{i+1}}, \boldsymbol{z_i}, \boldsymbol{z_i} \times \boldsymbol{z_{i+1}}\right]$$
$$\left[\boldsymbol{M_iM_{i+1}} - r_i.\boldsymbol{z_i}, \boldsymbol{z_i} \times \boldsymbol{z_{i+1}}, \boldsymbol{z_{i+1}}\right] = 0 \Leftrightarrow$$
$$r_i.\|\boldsymbol{z_i} \times \boldsymbol{z_{i+1}}\|^2 = \left[\boldsymbol{M_iM_{i+1}}, \boldsymbol{z_{i+1}}, \boldsymbol{z_i} \times \boldsymbol{z_{i+1}}\right]$$

In the case where $\boldsymbol{z_i} \times \boldsymbol{z_{i+1}} = \boldsymbol{0}$, we choose to place P_i at M_i, which means that r_i is set to 0. Then it comes $s_i = \boldsymbol{M_{i+1}M_i}.\boldsymbol{z_{i+1}}$.
In summary,

$$\boldsymbol{z_i} \parallel \boldsymbol{z_{i+1}} \Rightarrow r_i = 0$$
$$s_i = \boldsymbol{M_{i+1}M_i}.\boldsymbol{z_{i+1}}$$
$$\boldsymbol{z_i} \nparallel \boldsymbol{z_{i+1}} \Rightarrow r_i = \left[\boldsymbol{M_iM_{i+1}}, \boldsymbol{z_{i+1}}, \boldsymbol{z_i} \times \boldsymbol{z_{i+1}}\right]/\|\boldsymbol{z_i} \times \boldsymbol{z_{i+1}}\|^2$$
$$s_i = \left[\boldsymbol{M_iM_{i+1}}, \boldsymbol{z_i}, \boldsymbol{z_i} \times \boldsymbol{z_{i+1}}\right]/\|\boldsymbol{z_i} \times \boldsymbol{z_{i+1}}\|^2$$

---

[2] $[\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}]$ denotes the triple scalar product $(\boldsymbol{a} \times \boldsymbol{b}).\boldsymbol{c}$.

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 $R_T = (T, i_T, j_T, k_T)$, with i_T being the longitudinal axis, j_T the lateral axis, and k_T the vertical axis. Figure 3 illustrates the notations used.
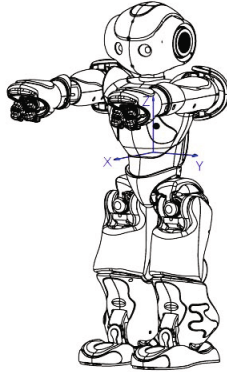

Fig. 2. Nao schematics in reference configuration where all joint angles are equal to zero

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 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

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

Point M_i belonging to joint axis z_i	Joint axis z_i
$M_{s1} = T = [0,0,0]^T$	$z_{s1} = k_T = [0,0,1]^T$
$M_{s2} = T = [0,0,0]^T$	$z_{s2} = k_T = [0,0,1]^T$
$M_1 = H = [-H_x^o, -\xi.H_y^o, -H_z^o]^T$	$z_1 = [0, \cos(45), \xi \sin(45)]^T$
$M_2 = H = [-H_x^o, -\xi.H_y^o, -H_z^o]^T$	$z_2 = i_T = [1,0,0]^T$
$M_3 = H = [-H_x^o, -\xi.H_y^o, -H_z^o]^T$	$z_3 = j_T = [0,1,0]^T$
$M_4 = K = [-K_x^o, -\xi.H_y^o, -H_z^o - L_f]^T$	$z_4 = j_T = [0,1,0]^T$
$M_5 = A = [-K_x^o, -\xi.H_y^o, -H_z^o - L_f - L_t]^T$	$z_5 = j_T = [0,1,0]^T$
$M_6 = A = [-K_x^o, -\xi.H_y^o, -H_z^o - L_f - L_t]^T$	$z_6 = i_T = [1,0,0]^T$
$M_{e1} = Ah = [-K_x^o, -\xi.H_y^o, -H_z^o - L_f - L_t - F_z^h]^T$	$z_{e1} = k_T = [0,0,1]^T$
$M_{e2} = Ah = [-K_x^o, -\xi.H_y^o, -H_z^o - L_f - L_t - F_z^h]^T$	$z_{e2} = -j_T = [0,-1,0]^T$

reference position, i.e. completely stretched vertically. $\xi = 1$ for the right leg and -1 for the left leg.

The first two axes z_{s1}[3] 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 hip, and therefore take into account the translation from T to H. Axes z_i refer to the joint rotation axes of angle θ_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 z_{e1}[4] and 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 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

---

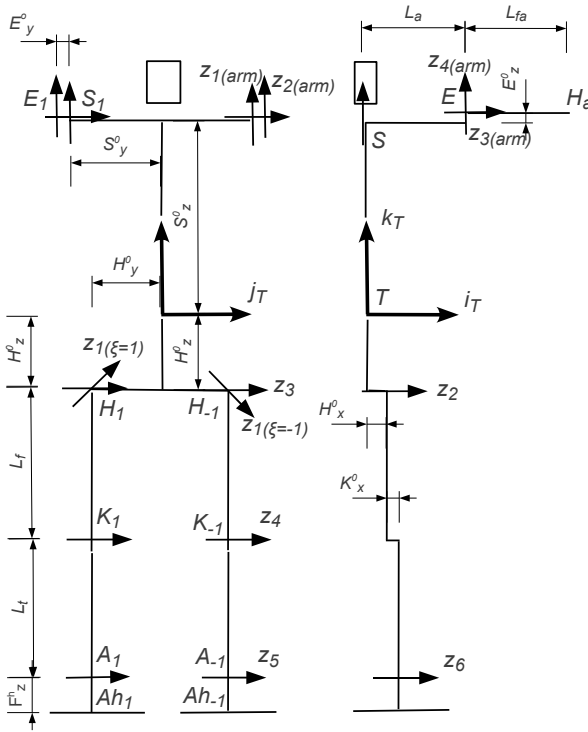[3] s stands for start.
[4] e stands for end.

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

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}[5] 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 as shoulder pitch, shoulder yaw, elbow roll and elbow yaw. The last two axes z_{e1}[6] and 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 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 θ_i given the trajectory of the end effector in the Cartesian space. The

---

[5] s stands for start.
[6] e stands for end.

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	$45(1+\xi)/2 + 135(1-\xi)/2$	$-\xi.\sqrt{2}.H_y^o$	$-90.0 + \Delta\theta_{yaw-pitch}^{hip}$
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

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

Point M_i belonging to joint axis z_i	joint axis z_i
$M_{s1} = T = [0,0,0]^T$	$z_{s1} = k_T$
$M_{s2} = T = [0,0,0]^T$	$z_{s2} = k_T$
$M_1 = S = [0, -\xi.S_y^o, S_z^o]^T$	$z_1 = j_T$
$M_2 = S = [0, -\xi.S_y^o, S_z^o]^T$	$z_2 = k_T$
$M_3 = E = [L_a, -\xi.(S_y^o + E_y^o), S_z^o + E_z^o]^T$	$z_3 = i_T$
$M_4 = E = [L_a, -\xi.(S_y^o + E_y^o), S_z^o + E_z^o]^T$	$z_4 = k_T$
$M_{e1} = H_a = [L_a + L_{fa}, -\xi.(S_y^o + E_y^o), S_z^o + E_z^o]^T$	$z_{e1} = k_T$
$M_{e2} = H_a = [L_a + L_{fa}, -\xi.(S_y^o + E_y^o), S_z^o + E_z^o]^T$	$z_{e2} = -j_T$

product of DHKK homogeneous matrices to go from the end effector coordinate 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:

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

where

- M^s involves the first three z vectors,
- M^e involves the last three z vectors,
- M^{θ_i} involves joint rotation of angle θ_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 θ_i are the unknowns and, to invert the direct geometric model, we can use the following notation:

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

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

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:

$$\begin{pmatrix} R_s & T_s \\ 0 & 1 \end{pmatrix}\begin{pmatrix} R_{123} & T_1 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} R_4 & T_4 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} R_{56} & T_5 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} R_e & T_e \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} R_0 & T_0 \\ 0 & 1 \end{pmatrix}$$

$$\Leftrightarrow \begin{pmatrix} R_{123} & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} R_{456} & T_4 + R_4T_5 \\ 0 & 1 \end{pmatrix} =$$

$$\begin{pmatrix} {}^tR_s & -{}^tR_sT_s - T_1 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} R_0({}^tR_e) & T_0 - R_0({}^tR_e)T_e \\ 0 & 1 \end{pmatrix}$$

$$\Leftrightarrow \begin{pmatrix} R_{123} & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} R_{456} & T_4 + R_4T_5 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} R' & M' \\ 0 & 1 \end{pmatrix}$$

with

$$R' = ({}^tR_s)R_0({}^tR_e)$$
$$T' = -T_1 + ({}^tR_s)(T_0 - Ts - R_0({}^tR_e)T_e)$$

which gives the following system to solve:

$$R_{123}(T_4 + R_4T_5) = T' \tag{8}$$
$$R_{123}R_{456} = R' \tag{9}$$

Squaring the first equation gives $(T_4 + R_4T_5)^2 = T'^2$ that allows to solve for the knee angle θ_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.
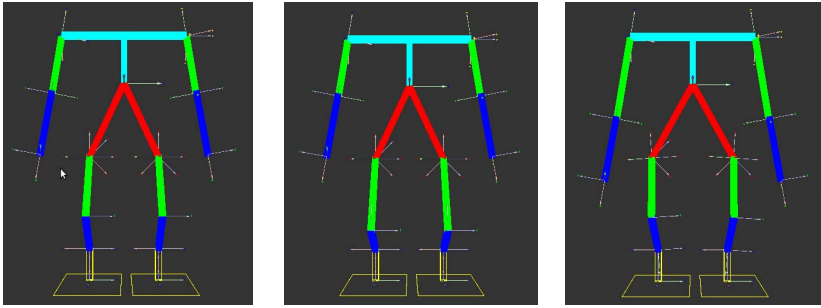
Fig. 4. Stick models of the 3D-SSL version of NAO displayed in the qt-opengl environment for the validation stage. The model on the left-hand side is related to the default model with the femur length equal to $0.12[m]$. The model in the middle is related to a change of the femur length, that is $0.14[m]$ instead of $012[m]$. 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.
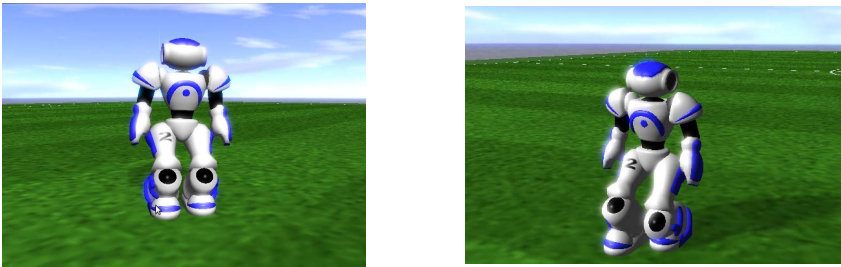

Fig. 5. Snapshots of the 3D-SSL NAO robot in standing position and making a step forward, achieved with the 3D-SSL official simulation server [7] [8] combined with the RoboViz monitor [10]

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'smodel 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.

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 $(M_i, \boldsymbol{z}_i)$ 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., Hartl, A., Röfer, T., Laue, T.: A Robust Closed-Loop Gait for the Standard Platform League Humanoid. In: Proc. 4th Workshop on Humanoid Soccer Robots, pp. 30–37 (2009)
2. Zorjan, M., Hugel, V., Blazevic, P.: Influence of hip joint axes change of orientation on power distribution in humanoid motion. In: Proc. IEEE ICARA: The 5th Int. Conf. on Automation, Robots and Applications, pp. 271–276 (2011)
3. Zorjan, M., Hugel, V.: Generalized Humanoid Leg Inverse Kinematics to Deal With Singularities. In: Proc. IEEE Int. Conf. on Robotics and Automation (2013)
4. Khalil, W., Kleinfinger, J.-F.: A new geometric notation for open and closed-loop robots. In: Proc. IEEE Int. Conf. on Robotics and Automation, pp. 1174–1180 (1986)
5. Denavit, J., Hartenberg, R.S.: A kinematic notation for lower-pair mechanisms based on matrices. Trans. ASME J. Appl. Mech. 23, 215–221 (1955)
6. Gouaillier, D., Hugel, V., Blazevic, P., Kilner, C., Monceaux, J., Lafourcade, P., Marnier, B., Serre, J., Maisonnier, B.: Mechatronic design of NAO humanoid. In: Proc. IEEE Int. Conf. on Robotics and Automation, pp. 769–774 (2009)
7. Obst, O., Rollmann, M.: Spark - A Generic Simulator for Physical Multi-Agent Simulations. In: Lindemann, G., Denzinger, J., Timm, I.J., Unland, R. (eds.) MATES 2004. LNCS (LNAI), vol. 3187, pp. 243–257. Springer, Heidelberg (2004)

8. SimSpark, a generic physical multiagent simulator system for agents in three-dimensional environments, `http://simspark.sourceforge.net/`
9. Hugel, V., Jouandeau, N.: Walking patterns for real time path planning simulation of humanoids. In: Proc. IEEE Int. Symp. on Robot and Human Interactive Comm (RO-MAN), pp. 424–430 (2012)
10. Stoecker, J., Visser, U.: RoboViz: Programmable Visualization for Simulated Soccer. In: Röfer, T., Mayer, N.M., Savage, J., Saranlı, U. (eds.) RoboCup 2011. LNCS, vol. 7416, pp. 282–293. Springer, Heidelberg (2012)