Georgia Institute of Technology, Technical Report, GT-GOLEM-2014-001, 2014

Krang Kinematics: A Denavit-Hartenberg Parameterization

Can Erdogan, Munzir Zafar, Mike Stilman

1 Introduction

Krang is a 19 degree of freedom robot with actuators along three different major axes with a unique physical structure. Moreover, in comparison to most robots on which kinematics is studied, Krang is neither a stationary robot with a fixed base nor is it a legged humanoid for which somewhat a standard has been established. Still, we would like the kinematics for Krang to be easy to understand, regenerate and implement. To this end, we choose the well-established Denavit-Hartenberg (DH) parameterization.

In the following, we will describe the DH parameterization for Krang. To do so, we will first present a refresher on DH and the specific rules we follow. With the advantage of DH being adopted for the last 60 years comes the disadvantage that different interpretations arose within the standard over the years. We follow one of the most popular and well-documented interpretations, and also describe it here.

After the tutorial, we will focus on the specific parameterization of Krang in three sections: wheels to torso, torso to left arm and torso to right arm. We separate the kinematics in this way because (1) the lower body of Krang, from wheels to the torso module, is significantly different than the upper body and (2) once lower body is established, left and right arms are quite similar to each other.

Lastly, we will discuss the inverse kinematics for the arms. The discussion will be mostly based on [2] where we will delve into the details of the technical work and relate it to the Krang's parameterization.

2 Denavit-Hartenberg Parameterization

In general, DH parameterization has two major steps: assigning frames to the actuators and deciding on the transformations between the frames. Remember that the goal of this exercise is to create a kinematic map for Krang to relate the joint positions and velocities we control with the behavior of the end-effectors. We first begin with assigning frames.

2.1 Assigning the frames

There are 3 cases in assigning coordinate frames to the modules. In all cases, the z-axis is aligned with the axis the module moves on and the direction can be chosen arbitrarily. The three cases are:

1. **z_{i-1} and z_i are not coplanar:** Figure 1 delineates this case where the module axes are not coplanar. In this case, there exists a unique line segment perpendicular to both z_{i-1} and z_i that connects them and is the shortest available segment. We define x_i colinear to this line and the origin o_i is placed in the intersection of x_i and z_i.

2. **z_{i-1} is parallel to z_i:** In this case, there is not a unique line segment and we can choose o_i anywhere along z_i. Once o_i is fixed, x_i is along the common normal of z_{i-1} and z_i, either pointing towards z_{i-1} or the opposite direction.

3. **z_{i-1} intersects z_i:** x_i is perpendicular to the plane defined by z_{i-1} and z_i, and the origin o_i can be placed anywhere on z_i although mostly placed at the intersection of z_i and x_i.

In addition to the module frames, we also need to add a tool frame which usually uses the same z-axis as the last module frame. After choosing the z-axis, the same rules can be followed for the x-axis.

Another important point to keep in mind is the indexing of the frames. Modules are indexed from 1 and frames are indexed from 0. Frame 0 is the world coordinates that do not change with the module angles. Moreover, frame 0 is placed at the center of module 1, frame 1 is at module 2 and etc. This convention ensures that the frames share the same indices with the modules that move them.

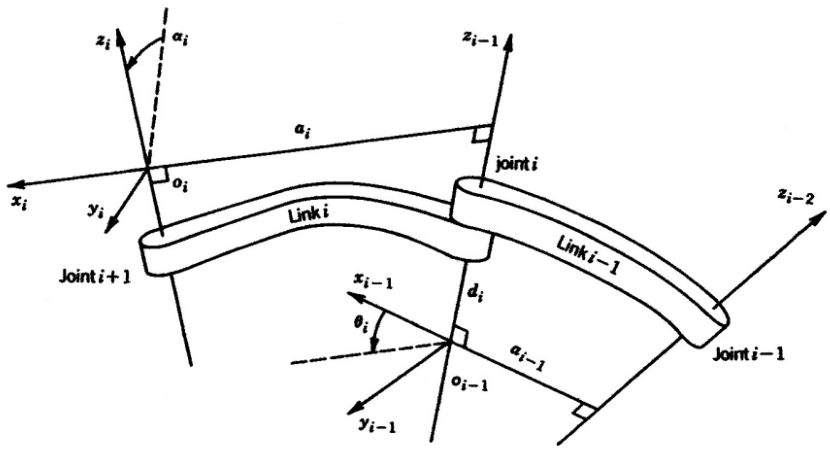

Figure 1: Frames assigned to actuators and the parameters $\{\alpha, \theta, a, d\}$ (From [1])

Lastly, the explanation in this section is mostly based on the excellent treatment by Spong et al. [1] and we suggest studying that reference. Also, the three cases described above indeed arise in Krang and we will describe their details further in the document. Now, having assigned frames to each module (and the end-effector), we go on to parameterize the transformations between the frames.

2.2 Parameterizing the transformations

Let $o_i x_i y_i z_i$ and $o_j x_j y_j z_j$ be the origin and axes of the i^{th} and j^{th} frames, f_i and f_j respectively. Let T_j^i be the 4×4 homogeneous transformation matrix that represent f_j with respect to f_i. For instance, the coordinates P^j of a point P in frame f_j can be transformed to the coordinates in frame f_i by: $P^i = T_j^i P^j$.

Our goal is now to define the transformation matrix T_j^i between the frames defined in the previous section. We use four main parameters as described in the table below:

Symbol	Definition
a_i	The distance along x_i from o_i to the intersection of the x_i and z_{i-1} axes
d_i	The distance along z_{i-1} from o_{i-1} to the intersection of the x_i and z_{i-1} axes.
α_i	The angle between z_{i-1} and z_i measured about x_i.
θ_i	The angle between x_{i-1} and x_i measured about z_{i-1}.

Let's define T_i^{i-1} given the parameters a_i, d_i, α_i and θ_i:

$$T_i^{i-1} = Rot_{z_i,\theta_i} Trans_{z_i,d_i} Trans_{x_{i-1},a_i} Rot_{x_{i-1},\alpha_i} \quad (1)$$

the matrix $Rot_{z,\theta}$ is a rotation around an axis z about θ radians and the matrix $Trans_{z,d}$ is a translation along an axis z for d units. We can look at Figure 1 and imagine a point P^i in the f_i coordinate frame, and move it towards the $(i-1)^{th} frame$ step by step. Now, the equation to get P^{i-1} is:

$$P^{i-1} = Rot_{z_{i-1},\theta_i} Trans_{z_{i-1},d_i} Trans_{x_i,a_i} Rot_{x_i,\alpha_i} P^i \quad (2)$$

where we do the following steps by matrix multiplication from right to left to get P^{i-1}:

1. Rotate the point around x_i for α_i degrees so that z_i and z_{i-1} are parallel.
2. Move the point a_i displacement along x_i (right). Now, if we imagine o_i moving with the point, it should coincide z_{i-1}, and z_i and z_{i-1} are colinear now.
3. Move the point d_i displacement along z_{i-1} (down). Now, again, if we imagine o_i is moving, it should councide o_{i-1}. Now, the only difference between the two frames is the rotation around z_{i-1}.
4. Rotate the point around z_{i-1} for θ_i degrees so that x_i and x_{i-1} are parallel.

The advantage of the DH parameterization and the factorization of the transformations into 4 basic operations is the simple closed-form expression for T_i^{i-1} (note: c and s stand for *cos* and *sin*):

$$T_i^{i-1} = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} c_{\alpha_i} & s_{\theta_i} s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i} c_{\alpha_i} & -c_{\theta_i} s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

Next, we discuss Krang's coordinate frames and determine the DH parameters based on its structure.

3 Krang Parameterization

The purpose of DH parameterization is to attain the transformations between successive modules' coordinate frames. However, inherently, DH requires an ordering of the modules: which module do you start with? For Krang, we start with the wheels and fix a frame f_0 with the y_0 frame aligned with the vertical of the room and x_0 and z_0 moving with the wheel axis. This allows us to add an additional transformation from the room origin to the robot position and orientation easily. Next, we explain the parameterization for the lower body of Krang.

3.1 Lower body

Figure 2 shows the frames assigned to the modules. There are a few points to note here. First, the two wheels are treated as a single module, as well as the two waist modules. Second, the notation is that θ values control the rotations about the z axes between the successive modules. The θ_1 and θ_2 values are based on the imu reading, q_{imu}, and the waist modules' angle, q_w, respectively. Next, we justify the placement of frames 1, 2 and the arm base frames $3l$ and $3r$.

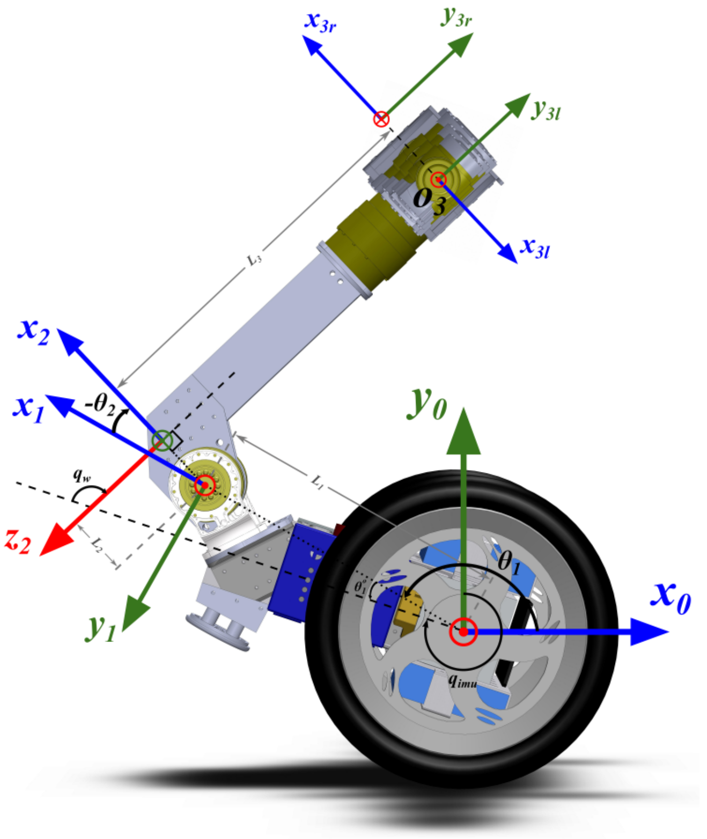

Figure 2: The base frames and the joint angles from the side view

The z-axis of the first modules, z_1, is parallel to z_0 so this frame falls into the second case where we place the x-axis of the i^{th} frame along the common normal z_{i-1} and z_i, just as x_1 is placed. Secondly, the frame 2 is an interesting case because the z-axes of the torso module and the waist modules are non-planar. In this case, we take the shortest unique line that is perpendicular to both z_1 and z_2 and place x_2 along that line. Lastly, we discuss the arm modules. An important note is that we assume:

> The z-axis of a horizontal arm module is **away** from the end-effector and the z-axis of a vertical arm module is **towards** the rounded covers with the "Schunk" labels.

Given this choice, the z-axes of both arms intersect with the z-axis of the torso module and we have the third case where we place the x-axes perpendicular to the plane created by the intersecting z-axes. Note that we choose the y-axes to be vertical because this convention helps with the arm frames later on.

Having fixed the frames, now we discuss the parameterization. Frame by frame, we will create the DH table in Table 1 where we have noted the controlling parameters values. We start with the transformation between frames 0 and 1, which is induced by the wheel motion. First, note that there is a mechanically induced offset θ_1^o between the base of the robot and the x_1 axis. Secondly, the only way to determine the angle of the base with respect to the room vertical, y_0, is the imu reading, denoted as q_{imu}. Using the two values, the θ_1 variable is computed as: $\pi/2 - q_{imu} - \theta_1^o$. We also have L_1 as the distance between the wheel axes and the waist modules, and d_1 is zero because both frame origins are through the middle of the robot.

Table 1: DH Parameters for the Lower Body

Frames	a_i	α_i	d_i	θ_i
0-1	L_1	0	0	$\pi/2 - q_{imu} - \theta_1^o$
1-2	L_2	$-\pi/2$	0	$\pi/2 - q_w + \theta_1^o$
2-3l	0	$-\pi/2$	-L_3	$\pi + q_{tor}$
2-3r	0	$-\pi/2$	-L_3	q_{tor}

Next, we look at the transformation between frames 1 and 2. Here, q_w, the waist modules' angle controls the θ_2 variable, with a constant offset of $-(\pi/2 + \theta_1^o)$. Note that while q_w in a clockwise rotation, θ_2 decreases, and we zero the waist modules when the spine and the base are parallel. The distance between the two z-axes is L_2 as shown in Figure 2 and the z_1 axis has to rotate $-\pi/2$ radians about x_2 to be paralel to z_2.

The last two transformations in the lower body are between the torso, f_2, and the first modules in the arms, f_{3l} and f_{3r}. There is a few details here. First, because the x_3s and z_2 intersect, the variables a_3s are zero and the distance of the spine L_3 expresses itself in the d_i column. Second, for both arms with the opposite x-z axes, the rotation about x_3s from z_3s to z_2 is $-\pi/2$. Lastly, although the rotation difference between x_2 and x_3r is only the torso's movement, q_{tor}, there is an offset of π for the left arm.

3.2 Upper Body

Figure 3 shows the frames for the left arm with the accompanying link lengths. In this section, we will describe two transformations: (1) from a horizontal module to a vertical one and (2) from a vertical module to horizontal one. Once these two types are established, the rest of the transformations are merely repetitions. Moreover, note that, the transformations for the two arms are also the same because after the difference in the base frames, f_{3l} and f_{3r}, the arms are identical and we can use the same parameterization with different variables. Note that all the DH parameters for the left arm is in Table 2 and we will only describe the first two rows.

Now, we describe the transformation from f_{3l} to f_{4l} - this is the horizontal to vertical case. Although the origin of f_{3l} is not denoted in Figure 3, it is at the intersection of y_3 and z_{3l}. Given that the z-axes intersect, this transformation falls into the third case where we automatically have a_{4l} to be zero. The distance between the x-axes is negative $L4$ since the translation is along z_{3l}, looking away from the end-effector. Note that there is a π radians rotation (plus or minus) around x_i for all the rotations in the arms since the z-axes change from horizontal to vertical and vice versa. Lastly, the rotation of f_{4l} is controlled by the joint rotation q_0^L.

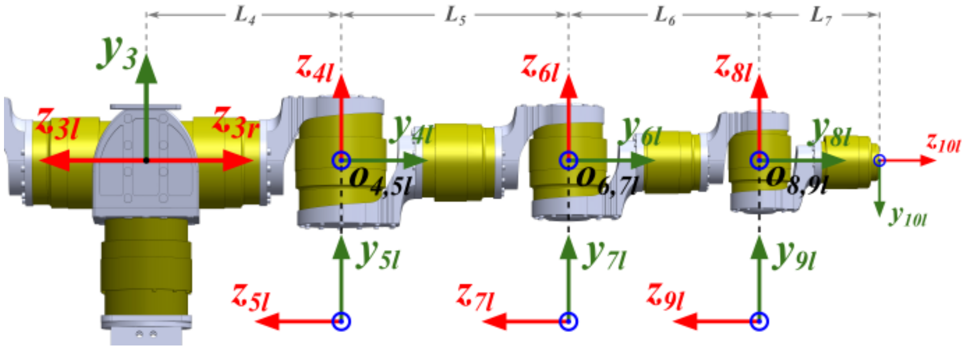

Figure 3: The left arm frames, the torso and the base of the right arm

The second important transformation is from f_{4l} to f_{5l} and this one is relatively simpler because the two frames share the same origin. Since they share an origin, there are no translations and we have already covered the rotation around the x-axes. The z-axis rotation is also similar in that it is the joints movement: q_1^L. One important detail that is specific to this joint only is the θ_{bend} variable which is used to take care of a *bend* in the bracket between the first and second modules. The ability to include such deformations in the kinematic model easily is one of the advantages of the DH parameterization.

Table 2: DH Parameters for the Left Arm

Frames	a_i	α_i	d_i	θ_i
3l-4l	0	$-\pi/2$	$-L_4$	q_0^L
4l-5l	0	$\pi/2 - \theta_{bend}$	0	q_1^L
5l-6l	0	$-\pi/2$	$-L_5$	q_2^L
6l-7l	0	$\pi/2$	0	q_3^L
7l-8l	0	$-\pi/2$	$-L_6$	q_4^L
8l-9l	0	$\pi/2$	0	q_5^L
9l-10l	0	π	$-L_7$	q_6^L

3.3 Kinect

In this section, we describe the two transformations that relate the shoulder bracket frame to the Kinect frame. Clearly, this relationship is vital to use visual input for manipulation, interaction and etc. Figures 4 and 5 show the two Solidworks images with the annotated distances. In the following we define the two transformations: (1) from bracket to Kinect hinge and (2) from hinge to Kinect 3D sensor.

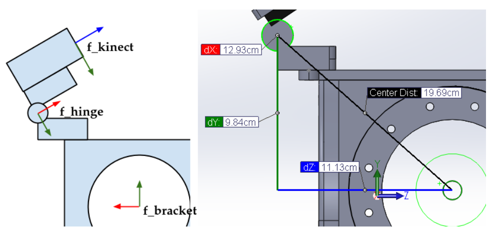

Figure 4: The three frames in play and the transformation from the bracket to the Kinect hinge

The transformation T_b^h is governed by the three variables $\delta_x^b = 11.13$, $\delta_y^b = 9.84$ and θ, the hinge angle. Given these three variables, we can write the transformation as:

$$T_b^h = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 11.13 \\ \sin(\theta) & \cos(\theta) & 0 & 9.84 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

Note that one can check the validity of this transform by evaluating the transformation of a point P^b at the origin at the frame b and observing that the translation $[\delta_x^b; \delta_y^b; 0]$ returns a point with position x and y axis values in frame h. A similar test can be done for the rotation. See the *kinematics/tests* folder for more.

The transformation T_h^k is simpler in the sense that we only need to take into account a change in axes and translation but not a rotation. Note that here we also have a translation along the arms, in the x direction of the Kinect frame since the 3D sensor is off the center of the robot. The variables δ_x^h, δ_y^h and δ_z^h can be seen in Figure 5 (Note that the axes in this frame belong to hinge frame although we will use the switched ones in Kinect frame):

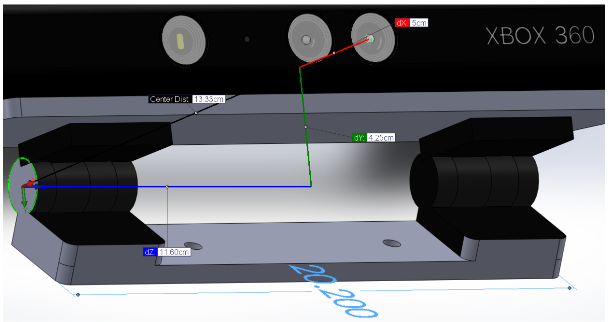

Figure 5: The transformation between the Kinect hinge and the 3D sensor lens

With these measurements, the transformation T_h^k can be written as where we get the 1.5 cm for the z-axis by subtracting 20.2/2 from the given 11.60 to use the center of the hinges as f_h:

$$T_h^k = \begin{bmatrix} 0 & 0 & 1 & 5 \\ 0 & 1 & 0 & -4.25 \\ -1 & 0 & 0 & 1.5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

We can apply a similar test by thinking of a point P^h at the origin of f_h and observing that (1) the x_k value is positive as expected, (2) the y_k value is position, and (3) the z_k is negative since the hinge is located “behind” the sensor.

References

[1] M.W. Spong, S. Hutchinson, and M. Vidyasagar. *Robot modeling and control*. John Wiley & Sons Hoboken ˆ eNJ NJ, 2006.

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