Inverse Kinematics and Basic Bipedal locomotion of an Autonomous Robot

William Clayton, University of South Carolina Upstate

2015

Abstract:

This paper describes the methods used to set up bipedal locomotion of a hobby level robot, specifically, the Lynxmotion Biped Pete. First, the direct kinematics are set up using the Denavit–Hartenberg principles to define all motion of the joints. The convenience of the Denavit–Hartenberg parameters allows for each joint's frame of reference to be described using a single matrix based on four parameters. The direct kinematics is used, when given the joint angles, to find the Cartesian coordinates of the final end effector, or the foot, respect to the base frame as well as its rotation in three dimensional space. Inverse kinematics is the opposite, given a final position of the foot and rotation, the resulting joint angles are found. Finally, once the inverse kinematics equations are calculated, the robot can be programmed with walking trajectories implemented manually or by any algorithm.

Introduction:

Direct kinematics and inverse kinematics have been a problem for robotics for a long time now. The main problem is in the amount of variation from system to system. However, for direct kinematics, there are many solutions for one system that are viable, and depend upon the method of implementation. While the main difficulty surrounding inverse kinematics is the large, possibly infinite number, of solutions for the joint angles given one end effector position. This means, that many times inverse kinematics is non-linear and often finding a solution involves geometric intuition to solve. Thus, much of the kinematics can only be generated on a case by case, unless there are similarities that can be transferred from robot to robot.

Methods:

- Direct Kinematics:

The direct kinematics set up for the robot is specific to the type of robot and the physical characteristic unique to that robot. The reference frame mapping used in this paper consists of having a base frame for the robot (A_b), and a base frame for each leg (A_r and A_l). From A_b to the base frames of the legs is a constant transformation matrix, determined when mapping reference frames. Dealing with just the right leg for now, however the same applies to the left, the remaining reference frames are determined using Denavit–Hartenberg (DH) convention. The DH convention is a set of steps taken that set up reference frames for each joint, whose motion can be describe with the same homogenous transformation matrix Eq. 1. The first step is to establish a base frame. Here, three base frame are used, one for the robot, and one for each legs. This allows for the end positions of each foot to be relatable to one another, while making the inverse kinematics simpler. Defining each frame uses the same methods for each, align the current frame's z axis with the next frame's joint rotation, point the x axis in the direction of the next link, and define y from the right hand rule using the x and z axis. Eq. 1 takes four parameters: a, distance along the current frame's x axis from the previous frame. α, the rotation around the current frames x axis to align the z axis with the next link's joint rotation. d, the height along the previous frames z axis where the current frame's origin is location. And v, the rotation around the z axis. Since the z axis is aligned with the joint's rotation, v is the joint's angle. The DH parameters for the robot exampled here is: Table 1, while the reference frame mapping is: Fig 1.

Frames	*a*	*α*	*d*	*v*
1	.5	$\frac{\pi}{2}$	0	$v1 + \frac{\pi}{2}$
2	1	0	0	v2
3	.5	0	0	v3
4	.5	$-\frac{\pi}{2}$	0	v4
5	0	$\frac{\pi}{2}$	0	$v5 - \frac{\pi}{2}$

Table 1

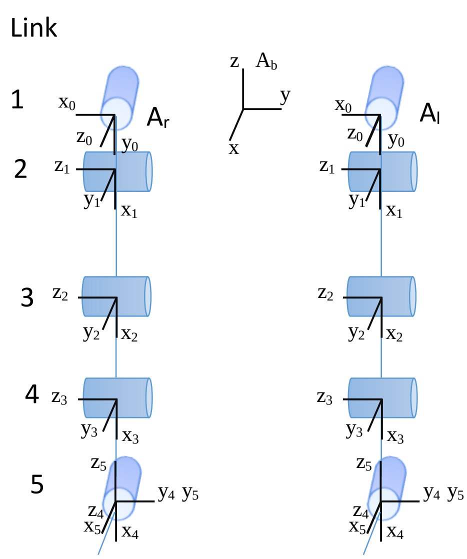

Figure 1

$$\begin{bmatrix} R & P \\ 0^T & 1 \end{bmatrix} \quad \text{Eq. 1}$$

In Eq. 1, R is the 3x3 rotation matrix, and P is the 1x3 position matrix of the reference frame. For the DH convention, which only rotates and transforms about the x and z axis, simply multiplying a rotation α about the x, a transformation a along the x, a rotation v about the z, and a transformation d along the z equals Eq. 2. Eq. 2 allows for each joint's movement to be describe using only four parameters. The types of joints used in this robot have constant values for every variable except v, making the direct kinematics depend on just one variable.

$$\begin{bmatrix} cosv & -sinvcos\alpha & sinvsin\alpha & acosv \\ sinv & cosvcos\alpha & -cosvsin\alpha & asinv \\ 0 & sin\alpha & cos\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq. 2}$$

The resulting equations calculated from the homogenous transformation matrices are then multiplied together to change reference frame from one to another. To find the final position of the end effector respective of the base frame, simply multiply through each frame's transformation matrix.

$$T_5^b = A_0^b A_1^0 A_2^1 A_3^2 A_4^3 A_5^4 \quad \text{Eq. 3}$$

Where T_5^b denotes the transformation frame from the base frame of the robot, A_b, to the end effector's frame of reference. This notation is used to describe the initial frame of reference on the top, and the reference frame being translated to on the bottom. Also, note A_0^b is the constant reference frame from the base frame of the robot, to the base frame of the specific leg, which is A_1^0. Eq. 3 will be very useful when solving inverse kinematics. Another useful way to describe T_5^b which will be useful for inverse kinematics is to break each column into its components.

$$T_5^b = \begin{bmatrix} n_5^b & s_5^b & a_5^b & p_5^b \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq. 4}$$

These components can then be used to determine the rotation of the x, n_5^b, rotation of the y, s_5^b, rotation of the z, a_5^b, and position offset, p_5^b. These components will allow the final position and rotation to be used to find the v variables in inverse kinematics.

- Inverse Kinematics

The inverse kinematic equations are equations that when given the T_5^b of Eq. 3, the v values for the joint angles are calculated. For this system, the first variable solved for is v_1. The first link of the robot lies entirely in the xz-plane, relative to A_b reference frame (which will be the reference frame assume unless otherwise specified). Also, seeing how the only other rotation about the y axis is the last link, which doesn't adjust the position of the end effector, the projection of the leg onto the xz-plane to find the angle of v_1, by using the p_5^b, from Eq. 4.

$$v_1 = a\tan 2(p.x - a_r, -p.z)^1 \qquad \text{Eq. 5}$$

Where a_r is the x offset from the A_b to the A_r (a_l for A_l). Since the origin of A_b is at the waist of the robot, for this system, the z values of the end effector are negative, while the angle of v_1 treats –z as the base axis. This particular robot cannot rotate v_1 more than 90° away from the robot (ether the left for the left leg, or right for the right), as such, one particular problem with this equation is when the final position of the foot is located above the hip, which would cause the projection to fall in the positive z axis instead of the negative. To adjust for this, subtract v_1 from π to find the actual angle needed, only when v_1 is greater than 90°.

Then using v_1, and the DH parameters of link 1, to generate the transformation matrix A_1^0. Using that and Eq. 3 gives us:

$$A_1^{0^{-1}} A_0^{b^{-1}} T_5^b = A_2^1 A_3^2 A_4^3 A_5^4 \qquad \text{Eq. 6}$$

$$T_5^1 = A_2^1 A_3^2 A_4^3 A_5^4 \qquad \text{Eq. 7}$$

In which T_5^1 is the transformation matrix from the end effector to the second link. The final link has two qualities that will allow us to generalize $A_2^1 A_3^2 A_4^3$, making solving the rest of the equation much

[1] Atan2 takes into account the signs of the two inputs to accurately calculate the quadrant the value is in.

simpler. First being the final transformation matrix does not change the position of the end effector, thus $p_5^1 = p_4^1$. And the second being that A_5^4 does not effect s_5^1, because A_5^4 rotates about s_5^1, thus $s_5^1 = s_4^1$. This will effectively allows for T_4^1 to be calculated regardless of A_5^4, due to the structure of link 2-4, T_4^1 can be treated as a planar arm, as shown in Fig. 2.

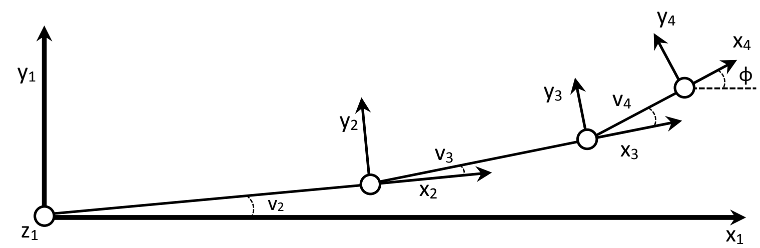

Figure 2

The angle φ is the final rotation about z (relative to the T_5^1 reference frame) of which the end effector is orientated. By using the generic rotation matrix about z, Eq. 8, we can find φ by using s_5^1, from T_5^1, in Eq. 9.

$$R_z(v) = \begin{bmatrix} cosv & -sinv & 0 \\ sinv & cosv & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{Eq. 8}$$

$$\varphi = atan2(-\,s_5^1.x,\ s_5^1.y) \quad \text{Eq. 9}$$

Using Eq. 9 to find φ, φ can then be used to find the position origin of A_3^2, or reference frame three in Fig. 2. p_3^2 is the final position subtract the components of link 4.

$$p_3^2 = p_5^1 - a_4 \begin{bmatrix} cos\varphi \\ sin\varphi \\ 1 \end{bmatrix} \quad \text{Eq. 10}$$

Then using the fact that the components of the each link has to add together to get the final position, then p_3^2 can also be represented as:

$$p_3^2 = a_2 \begin{bmatrix} cosv_2 \\ sinv_2 \\ 1 \end{bmatrix} + a_3 \begin{bmatrix} cos(v_2 + v_3) \\ \sin(v_2 + v_3) \\ 1 \end{bmatrix} \quad \text{Eq. 11}$$

Eq. 10 calculates p_3^2 by using φ, while Eq. 11 expresses p_3^2 in terms of v_2 and v_3. By taking Eq. 11 apart, into its x and y components, and squaring then adding the two equations yields:

$$p_3^2.x^2 + p_3^2.y^2 = {a_2}^2 + {a_3}^2 + 2a_2a_3\cos v_3 \qquad \text{Eq. 12}$$

Which, solving for v_3 gives:

$$\cos v_3 = \frac{p_3^2.x^2 + p_3^2.y^2 - {a_2}^2 - {a_3}^2}{2a_2a_3} \qquad \text{Eq. 13}$$

Cosv$_3$ has a range of 0 to π, thus needs to be converted to get the full range:

$$sinv_3 = \pm\sqrt{1 - cosv_3} \qquad \text{Eq. 14}$$

$$v_3 = atan2(sinv_3, cosv_3) \qquad \text{Eq. 15}$$

Where, in Eq. 14, the positive value corresponds to the elbow-down type position, and the negative value corresponds to the elbow-up type position. Then, by taking v_3 and substituting it into Eq. 11, Eq. 11 becomes two equations depended on two values, cosv$_2$ and sinv$_2$. Then by solving the system of equations for cosv$_2$ and sinv$_2$ yields the two equations:

$$sinv_2 = \frac{(a_2 + a_3cosv_3)p_3^2.y - a_3sinv_3p_3^2.x}{p_3^2.x^2 + p_3^2.y^2} \qquad \text{Eq. 16}$$

$$cosv_2 = \frac{(a_2 + a_3cosv_3)p_3^2.x - a_3sinv_3p_3^2.y}{p_3^2.x^2 + p_3^2.y^2} \qquad \text{Eq. 17}$$

Then, repeating the same process as Eq. 15:

$$v_2 = atan2(sinv_2, cosv_2) \qquad \text{Eq. 18}$$

Using the fact that φ is the final rotation about z (relative to the T_4^1 reference frame), which is simply the angles of each joint added together, then v_4 can be found from φ, v_2, and v_3.

$$\varphi = v_2 + v_3 + v_4 \qquad \text{Eq. 19}$$

Solving for v_4:

$$v_4 = \varphi - v_2 - v_3 \qquad \text{Eq. 20}$$

Now the only variable left to find is v_5. By taking v_2, v_3, and v_4 and generating the transformation matrices using their respective DH parameters, along with Eq. 6, or Eq. 7, A_5^4 can be isolated.

$$A_4^{3^{-1}} A_3^{2^{-1}} A_2^{1^{-1}} A_1^{0^{-1}} A_0^{b^{-1}} T_5^b = A_5^4 \quad \text{Eq. 21}$$

By knowing the layout of the homogenous transformation matrices, shown in Eq. 2, it's obvious that v_5 is already isolated in the first column, n_5^4, thus the final equations needed for inverse kinematics:

$$v_5 = \operatorname{atan} 2(n_5^4.y, n_5^4.x) \quad \text{Eq. 22}$$

After having implemented the equations above to find the five joint angle variables, a simple way to test the accuracy of the implementation is to generate various joint angles, run them through the direct kinematics, then take the final position and rotation calculated and find the necessary joint angles using the inverse kinematics a compare between the two.

- Basic Walking Gait

During walking, the legs go through two stages, a support stage and a swing stage. The robot shifts support to one of the leg, while swinging the other. Then a double support stage takes place before the legs switch which one supports and which one swings. For a robot's gait to be statically stable, at any point during the motion, the robot's center of mass has to remain above the robots support polygon. The support polygon, is the area covered by the supporting legs. During a one leg support phase, the support polygon is simply the individual foot. While during a two leg support phase, the support polygon is the area covered by the two feet and the area between the two feet. By mapping out a walking gait that keeps the robots center of mass above the support polygon, the robot will not fall over, while on level ground.

Results:

The goal of this project was to establish kinematics for a specific robot with the goal of being able to implement any walking algorithm design for the robot, and to create a simple static walking gait for the robot. The direct kinematics defined for the robot accurately define its motion, and help establish the inverse kinematics perfectly. The direct kinematics are capable of being adjusted to fit any coordinate system, it simply requires adjusting the DH parameter a. The inverse kinematics described here are by no means the only solution for this system, and may not be the best solution, however, it functioned well for its task. The equations are by no means perfect and may need tweaking for different algorithms to us, but they are workable in their current state.

Conclusion:

These two tools created, the direct and inverse kinematics, were created with the purpose of being able to implementing various other walking techniques, for example, ZMP (zero moment point) was a specific example thought about when designing the purpose for the kinematics of this robot. This project hopes to be the ground work for this robot and future projects to go on to more advance walking techniques and implementations.