Kinematics of an elbow manipulator with forearm rotation: the Excalibur

Rosanna Heise and Bruce A. MacDonald

Knowledge Sciences Institute
The University of Calgary
Calgary, Alberta, Canada, T2N 1N4
e-mail: heise@cpsc.UCalgary.CA, bruce@cpsc.UCalgary.CA

Abstract

General methods and typical specific solutions for robot arm geometry are well-known. This paper presents a detailed solution for a six joint manipulator which has a rotation at mid-forearm rather than a third wrist axis. Details are given indicating how real joint angles relate to those modeled by the more abstract kinematics. All degeneracies are considered and methods for handling them are given. The paper provides a complete tutorial for kinematic modeling with a specific arm.

Introduction

General methods for solving the geometry of robot arms are well-known [Paul 1981]. In addition examples of typical arm configurations — such as the elbow and Stanford arms — have been given, providing the basic insight required for understanding robot kinematics. The paper complements this by providing a complete kinematic solution for another manipulator, one with a joint mid-way along the forearm. Included are the details of how real joint axes correspond to the joints used in the geometric model, and methods for handling all degeneracies. This complete example is a tutorial for handling the geometry and reality of robot kinematics. The paper was motivated by difficulties experienced while tuning the kinematics code supplied with the arm. In fact, the supplied inverse kinematics incorrectly calculated several of the joint values. Small problems in the complex geometric models can result in considerable frustration, both for the experienced and inexperienced. A full development of forward and inverse kinematics is included here to obviate this. A brief summary of the algebra of linkages starts the discussion, followed by the forward and inverse kinematics of the Excalibur robot[1].

[1]Product of RSI Robotic Systems International, Sidney, B.C. See [Excalibur 1986] for complete specifications of the robot.

1 Overview of the Method

A transformation between bases or coordinate frames defined at each manipulator link, is easily accomplished by matrices.[2] [Paul 1981] states a simple method for formulating the matrices. Any manipulator consists of links connected at joints. Movement of each link is either a revolution about (revolute joint) or translation along (prismatic joint) the joint axis. Figure 1 shows how each link can be characterized by a length, twist, distance, angle, and basis.[3] With these parameters, transformation from link $n-1$ to link n is comprised of four steps:

1. rotate about z_{n-1} by an angle of θ_n
2. translate along z_{n-1} by distance d_n
3. translate along x_n (was x_{n-1}, now rotated) by length a_n
4. rotate by the twist α_n about x_n.[4]

Expressing this relationship as a product of matrices yields the change of basis matrix, A_n, which takes coordinates from the frame of link n to that of link $n-1$

$$\begin{aligned} A_n &= Rot(z,\theta_n)Trans(0,0,d_n)Trans(a_n,0,0)Rot(x,\alpha_n) \\ &= \begin{pmatrix} \cos\theta_n & -\sin\theta_n\cos\alpha_n & \sin\theta_n\sin\alpha_n & a_n\cos\theta_n \\ \sin\theta_n & \cos\theta_n\cos\alpha_n & -\cos\theta_n\sin\alpha_n & a_n\sin\theta_n \\ 0 & \sin\alpha_n & \cos\alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{aligned}$$

and the overall transform for an n link arm is $T_n = A_1\, A_2\, A_3\, \cdots\, A_n$. ZT_nE is the matrix which transforms end effector coordinates into world coordinates, provided Z is the change of basis from link one to world coordinates and E is the change of basis from the end of the gripper to link n coordinates. ZT_nE can be expressed directly in world coordinates as a location p and an orientation, illustrated in figure 2. The orientation is a normal vector n, an orientation vector o, and an approach vector a, yielding the *noap* matrix:[5]

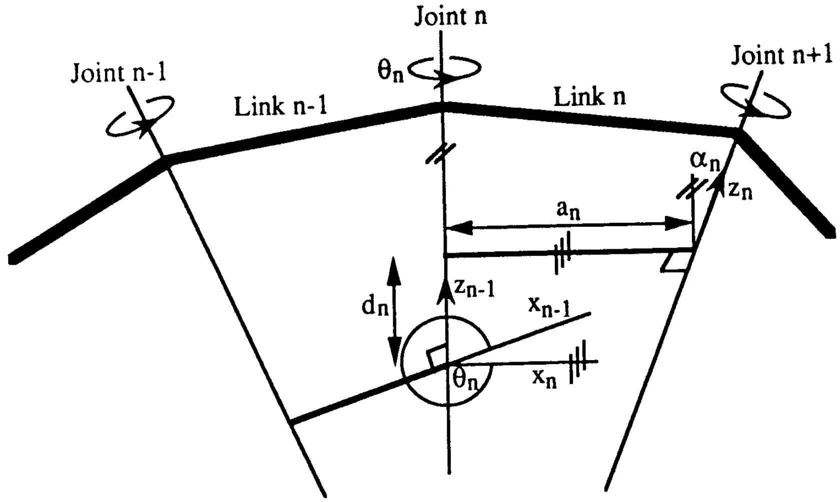
$$noap = \begin{pmatrix} | & | & | & | \\ n & o & a & p \\ | & | & | & | \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

[2]Change of basis matrices can be found in most elementary algebra books, for example, [Grossman 1987].

[3]All of the Excalibur's joints are revolute. For a discussion of bases at prismatic joints see [Paul 1981].

[4]Angles and distances are expressed in terms of the current basis. For example, the sign of α_n is determined by the direction of x_n.

[5]n, o, and a are easily converted to roll, pitch, and yaw. See [Paul 1981].



a_n length of the common normal connecting the two joint axes

α_n twist between the two joint axes. measured in a plane perpendicular to the common normal.

d_n distance separating the two normals along the axis of joint n.

θ_n angle between two links, measured between the normals in a plane perpendicular to them.

orig_n intersection of the common normal between joint axes n and $n+1$ and joint axis $n+1$.[a]

z_n Aligned with the axis of joint $n+1$.

x_n Aligned with the common normal between joint axis n and joint axis $n+1$.[b]

y_n $x_n \times z_n$.

[a]Should these axes intersect, the point of intersection is the origin. If the axes are parallel, the origin is so chosen that the joint distance, d_i, is zero to the next defined coordinate frame origin.
[b]If these axes intersect, the x-axis is aligned with $z_{n-1} \times z_n$.

Figure 1: Setting up coordinate frames for an arbitrary manipulator (adapted from [Paul 1981])

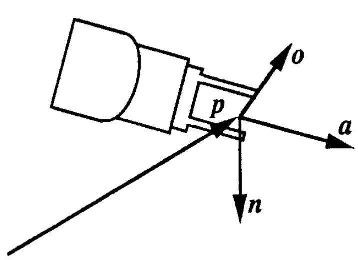


Figure 2: Normal, orientation, approach, and position

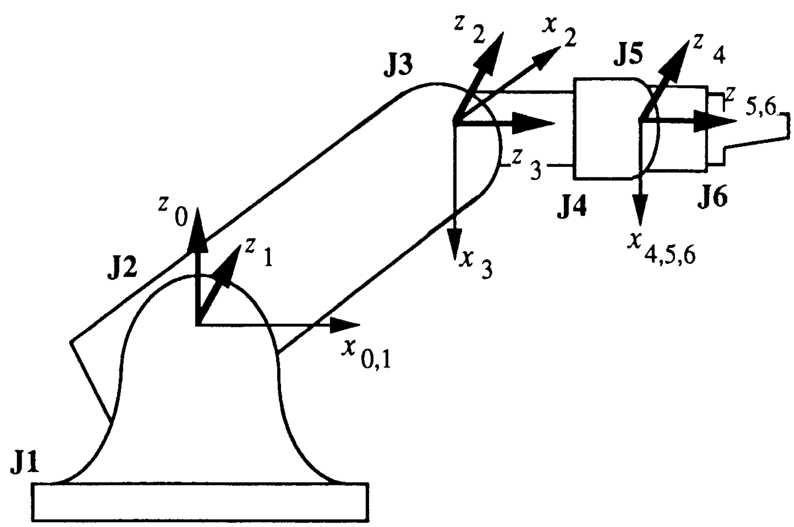


Figure 3: Coordinate frames for the Excalibur

link	θ_n	α_n	a_n	d_n	$\cos\alpha_n$	$\sin\alpha_n$
1	θ_1	$-90°$	0	0	0	-1
2	θ_2	$0°$	L_2	0	1	0
3	θ_3	$90°$	0	0	0	1
4	θ_4	$-90°$	0	L_{34}	0	-1
5	θ_5	$90°$	0	0	0	1
6	θ_6	$0°$	0	0	1	0

Table 1: The Excalibur link parameters. $L_{n_1 n_2 \cdots n_k}$ is the combined length of links n_1, n_2, $\cdots$ n_k

2 Excalibur Forward Kinematics

Using the method outlined in the previous section, conversion from Excalibur joint coordinates to world coordinates is easily accomplished. Coordinate frames[6] are set up for each link as shown in figure 3, resulting in the link parameters of table 1. The sign of α_n is determined using the right-hand rule with the thumb pointing along the direction of z_n. Substituting these values into A matrices and writing $\cos\theta_n$ as c_n and $\sin\theta_n$ as s_n yields the six matrices of figure 4.

Two translations are necessary to finish the conversion: a translation by L_1 along the z-axis to move the world origin to the base of the manipulator and a translation by L_{56} along

[6]There are, of course, many ways of defining these frames, depending on the direction chosen for each of the basis vectors. The frames given here allow expression of length as positive. If the z-axes were pointing down, then length would be negative.

$$A_1 = \begin{pmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad A_2 = \begin{pmatrix} c_2 & -s_2 & 0 & L_2c_2 \\ s_2 & c_2 & 0 & L_2s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_3 = \begin{pmatrix} c_3 & 0 & s_3 & 0 \\ s_3 & 0 & -c_3 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad A_4 = \begin{pmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & L_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_5 = \begin{pmatrix} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad A_6 = \begin{pmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Figure 4: A matrices for the Excalibur.

$$Z = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad E = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_{56} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Figure 5: World origin and tool translation matrices

the z-axis since the point p lies at the end of the gripper (figure 5), forming the *noap* matrix with components shown in figure 6, where $c_{23} = \cos(\theta_2 + \theta_3)$ and $s_{23} = \sin(\theta_2 + \theta_3)$.

It is important to notice that the Excalibur robot does not measure its joint angles in exactly the manner suggested above, as shown in figure 7. As angles are read, the following modifications must be made to the joint values before sending them through the kinematics above. If ρ_n is the angle read from the robot, then

$$\begin{array}{lll} \theta_1 = \rho_1 & \theta_2 = -\rho_2 - 30^\circ & \theta_3 = -\rho_3 + 150^\circ \\ \theta_4 = \rho_4 & \theta_5 = -\rho_5 & \theta_6 = \rho_6. \end{array} \tag{1}$$

3 Excalibur Inverse Kinematics

Inverse kinematics must solve the kinematic equations for all θ_n, thus converting world coordinates to joint angles. This task is more difficult since sine and cosine are not invertible. Finding a solution involves both algebraic and geometric techniques. Many of the angles are determined from properties of triangles, while others are easier isolated using the matrix equations, premultiplying both sides of $noap = ZA_1A_2 \cdots A_nE$ by inverses of the right hand side matrices.

$$
\begin{aligned}
n_x &= c_1[c_{23}(c_4c_5c_6 - s_4s_6) - s_{23}s_5c_6] - s_1[s_4c_5c_6 + c_4s_6] \\
n_y &= s_1[c_{23}(c_4c_5c_6 - s_4s_6) - s_{23}s_5c_6] + c_1[s_4c_5c_6 + c_4s_6] \\
n_z &= s_{23}[s_4s_6 - c_4c_5c_6] - c_{23}s_5c_6 \\
\\
o_x &= c_1[s_{23}s_5s_6 - c_{23}(c_4c_5s_6 + s_4c_6)] + s_1[s_4c_5s_6 - c_4c_6] \\
o_y &= s_1[s_{23}s_5s_6 - c_{23}(c_4c_5s_6 + s_4c_6)] - c_1[s_4c_5s_6 - c_4c_6] \\
o_z &= s_{23}[c_4c_5s_6 + s_4c_6] + c_{23}s_5s_6 \\
\\
a_x &= c_1(c_{23}c_4s_5 + s_{23}c_5) - s_1s_4s_5 \\
a_y &= s_1(c_{23}c_4s_5 + s_{23}c_5) + c_1s_4s_5 \\
a_z &= c_{23}c_5 - s_{23}c_4s_5 \\
\\
p_x &= c_1[c_{23}c_4s_5L_{56} + s_{23}(c_5L_{56} + L_{34}) + c_2L_2] - s_1s_4s_5L_{56} \\
p_y &= s_1[c_{23}c_4s_5L_{56} + s_{23}(c_5L_{56} + L_{34}) + c_2L_2] + c_1s_4s_5L_{56} \\
p_z &= -s_{23}c_4s_5L_{56} + c_{23}(c_5L_{56} + L_{34}) - s_2L_2 + L_1
\end{aligned}
$$

Figure 6: *noap* matrix components

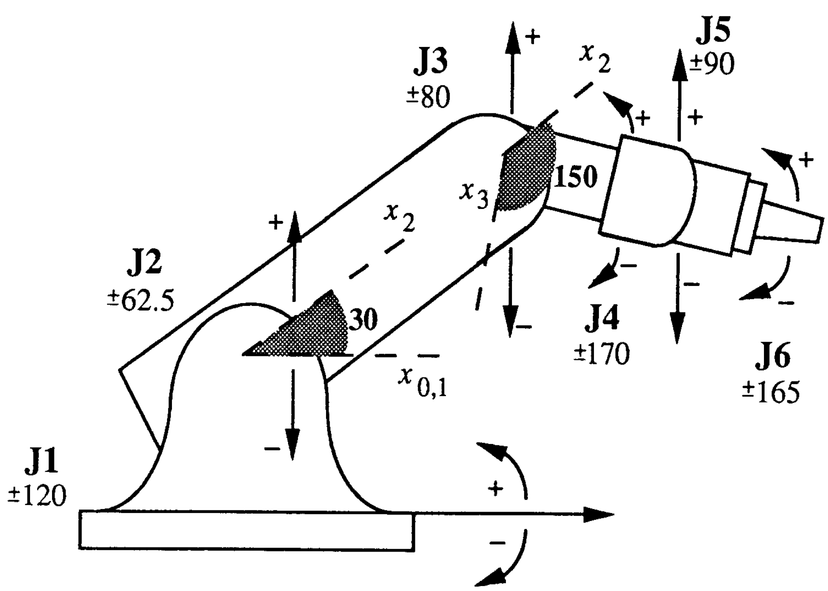


Figure 7: Excalibur rest position (all angles read as 0) and joint limits

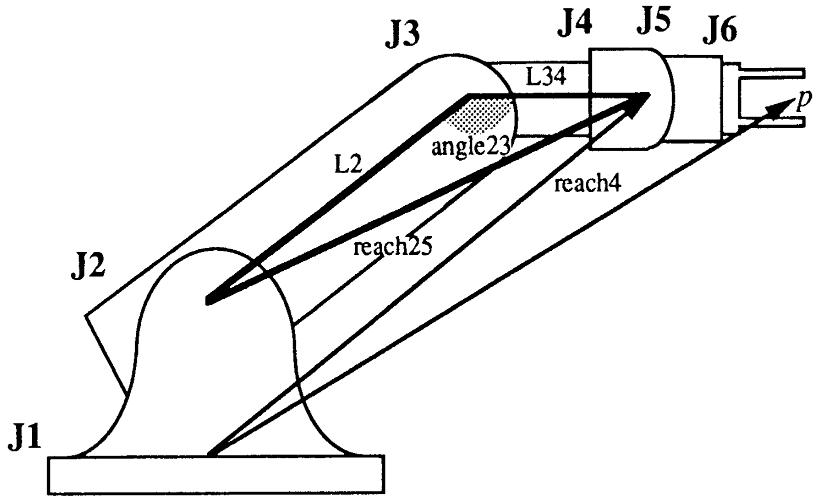


Figure 8: Useful *reach* vectors for determining joint angles

One solution for the joint angles, $\theta_1, \theta_2, \ldots \theta_6$, is now presented. Suppose that the vectors n, o, a, and p are known.

1. Define *reach*4 to be the vector from the base origin to the end of link four, as shown in figure 8.

$$reach4 = (reach4_x, reach4_y, reach4_z) = p - L_{56}a.$$

   This is necessary, since joint one and joint five both *potentially* cause a displacement about the same axis. The vector eliminates the contribution of joint five, enabling the calculation of θ_1. Let $reach4_{xy}$ be the projection of *reach*4 into the x-y plane.

2. Two cases arise when considering the value for θ_1:

   (a) If the arm is to be placed vertical (*i.e.* $reach4_{xy} = 0$) then it is unnecessary to move the first joint since joints four and six can account for any change required in orientation.

   (b) Otherwise, $\theta_1 = \text{atan2}(reach4_y,\ reach4_x)$.[7] If, however, the elbow of the manipulator (joint three) is to be positioned downwards then the solution for joint one is not this easy. Instead, when the above calculation for θ_1 exceeds the manipulator bounds, assume that the robot is bent backwards over the base, as in figure 9. Now θ_1 is changed by π. Alternatively, the bounds for joint one may not be exceeded but the solution is still off by π. This condition will only by noticed after solving for the orientation joints (four, five, and six) is impossible.

3. Let *reach*25 be the length joining joint two to joint five.

$$reach25 = \sqrt{(reach4_{xy})^2 + (reach4_z - L_1)^2}.$$

   Concentrate on the triangle formed with sides L2, L34, and reach25, visible previously in figure 8.

[7] atan2(y,x) = $\tan^{-1}\frac{y}{x}$ where the sign on both components uniquely specifies the quadrant.

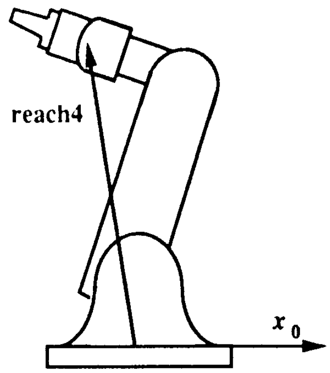
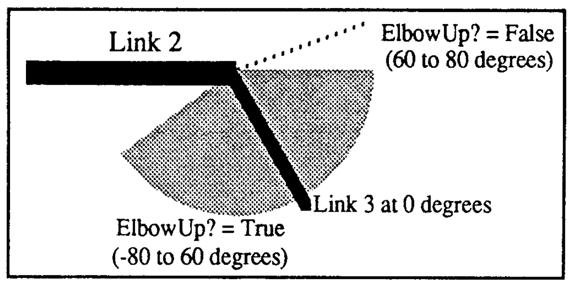
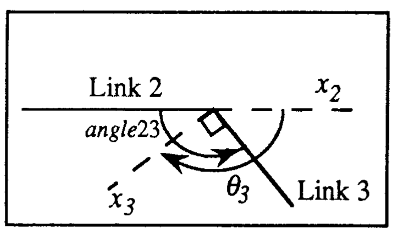


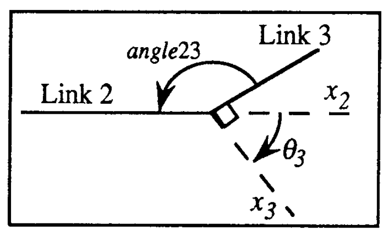
Figure 9: The excalibur bent over itself. Elbow is down.



(a) Range of link 3



(b) ElbowUp? = True



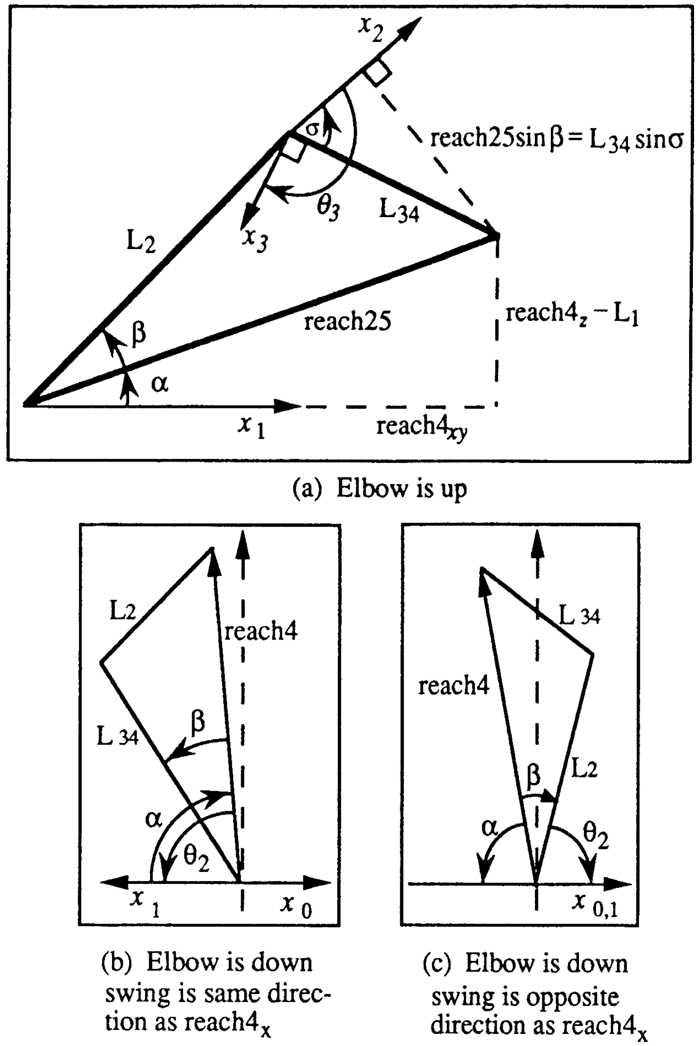
(c) ElbowUp? = False

Figure 10: Finding the angle at link 3

4. Call the angle between links two and three, *angle*23. The cosine of this angle can be determined using the cosine law.[8] This cosine calculation may yield an absolute value greater than one, indicating that the position the robot is required to reach is not within its bounds.

5. Uniquely determining θ_2 and θ_3 is geometrically impossible since their axes are parallel. However, by setting one parameter, *elbowUp?*, this problem is alleviated. Figure 10

[8] $\cos\theta = \frac{a^2+b^2-c^2}{2ab}$ where a, b, and c form a triangle and θ is the angle between a and b.



(a) Elbow is up

(b) Elbow is down swing is same direction as reach4$_x$

(c) Elbow is down swing is opposite direction as reach4$_x$

Figure 11: Summing α and β to determine θ_2

illustrates the significance of *elbowUp?*.[9] The solution for θ_3 is then

$$\theta_3 = \begin{cases} 270^\circ - angle23 & \text{if } elbowUp? = \text{true}^{10} \\ angle23 - 90^\circ & \text{otherwise.}^{11} \end{cases}$$

6. The next task is to solve for θ_2 by summing two angles: α and β, as shown in figure 11.

   - $\alpha = \text{atan2}\,(reach4_z - L_1,\ reach4_{xy})$
   - $\beta = \text{atan2}\,(\sin\beta,\ \cos\beta)$ where $\cos\beta$ is determined using the cosine law on the triangle $reach25$-L_2-L_{34} and $\sin\beta$ is determined using similar right triangles. As illustrated

[9]The value of *elbowUp?* may at times determine whether a solution can be computed. *elbowUp?* is usually set to true, thus leaving link three less restricted.

[10]This formula is easier to visualize as $(180^\circ - angle23) + 90^\circ$.

[11]Determined using complementary angles about the lines formed by link 2 and link 3.

in the figure,

$$\begin{aligned}\sin\beta &= \frac{L_{34}\sin\sigma}{reach25}\\ &= \frac{L_{34}\sin(\theta_3-90°)}{reach25}.\end{aligned}$$

When the elbow is up, $\theta_2 = -(\alpha+\beta)$. Otherwise the elbow is down and the solution depends on whether the swing (joint one) axis is in the same direction as $(reach4_x,\ reach4_y)$. If so, *i.e.* the cosine of θ_1 has the same sign as $reach4_x$, then once again $\theta_2 = -(\alpha+\beta)$, arising in part (b) of the figure. In the remaining case, (c) of the figure, $\theta_2 = \alpha - \beta - \pi$.

7. Adding θ_2 and θ_3 gives θ_{23}.

   Solving for the remaining three joints is difficult and resorting to algebra is helpful. These joints are primarily concerned with the orientation of the end effector, whereas the first three were concerned with attaining the desired position.

8. The cosine of angle five is isolated in the equation $A_3^{-1}A_2^{-1}A_1^{-1}Z^{-1}(noap) = A_4A_5A_6E$:

$$\begin{pmatrix} c_1c_{23} & s_1c_{23} & -s_{23} & L_1s_{23}-L_2c_3\\ -s_1 & c_1 & 0 & 0\\ c_1s_{23} & s_1s_{23} & c_{23} & L_1c_{23}-L_2s_3\\ 0 & 0 & 0 & 1\end{pmatrix}\begin{pmatrix} | & | & | & |\\ n & o & a & p\\ | & | & | & |\\ 0 & 0 & 0 & 1\end{pmatrix}$$
$$= \begin{pmatrix} f_{31}(n) & f_{31}(o) & f_{31}(a) & f_{31}(p)+L_1s_{23}-L_2c_3\\ f_{32}(n) & f_{32}(o) & f_{32}(a) & f_{32}(p)\\ f_{33}(n) & f_{33}(o) & f_{33}(a) & f_{33}(p)+L_1c_{23}-L_2s_3\\ 0 & 0 & 0 & 1\end{pmatrix}$$
$$= \begin{pmatrix} c_4c_5c_6-s_4s_6 & -c_4c_5s_6-s_4c_6 & c_4s_5 & L_{56}c_4s_5\\ s_4c_5c_6+c_4s_6 & -s_4c_5s_6+c_4c_6 & s_4s_5 & L_{56}s_4s_5\\ -s_5c_6 & s_5s_6 & c_5 & L_{56}c_5+L_{34}\\ 0 & 0 & 0 & 1\end{pmatrix}$$

   where[12]

$$\begin{aligned} f_{31}(v) &= (c_1c_{23})v_1 + (s_1c_{23})v_2 - s_{23}v_3\\ f_{32}(v) &= -s_1v_1 + c_1v_2\\ f_{33}(v) &= (c_1s_{23})v_1 + (s_1s_{23})v_2 + c_{23}v_3.\end{aligned}$$

9. Comparing the value in position 3-3 of both sides of this equation gives

$$\cos\theta_5 = f_{33}(a).$$

[12] f_{ij} refers to the function resulting from the matrix equation $A_i^{-1}A_{i-1}^{-1}\cdots A_1^{-1}T_n = A_{i+1}A_{i+2}\cdots A_n$ in the jth row [Paul 1981].

10. In the trivial case ($c_5 = 1$), $\theta_5 = 0$ since $-90° \leq \theta_5 \leq 90°$. This implies that the axes of rotation for joints four and six are parallel, hence the rotation should be split between the two. The previous matrix equation provides the necessary value:

$$\begin{aligned}\frac{f_{32}(n)}{f_{31}(n)} &= \frac{s_4c_5c_6 + c_4s_6}{c_4c_5c_6 - s_4s_6} \\ &= \frac{s_4c_6 + c_4s_6}{c_4c_6 - s_4s_6} \quad \text{since } c_5 = 1 \\ &= \frac{\sin(\theta_4 + \theta_6)}{\cos(\theta_4 + \theta_6)}.\end{aligned}$$

Thus $\theta_4 + \theta_6 = \operatorname{atan}(f_{32}(n),\ f_{31}(n))$. Assigning actual values to θ_4 and θ_6 can be done in many ways. Whenever possible it is a good idea to ensure that the amount of motion occurring in both joints is approximately the same. This is accomplished by finding the difference between the new required value for $\theta_4 + \theta_6$ and the old value, adding half to each joint.

11. If $\theta_5 \neq 0$ it is generally impossible to find a unique solution for the last three joints. This interaction is different from that between joints two and three, where a parameter was allocated to aid the solution. There each solution resulted in a configuration of the manipulator which *looked* different. In this case the different solutions result in the manipulator looking as if it is in the same position. Hence the actual solution found is not important in the function of the manipulator.[13] The strategy is to assume that θ_5 is a positive angle, then calculate θ_4 and θ_6 and ensure that they fall within the limits of the manipulator. If they do not, then assume that θ_5 is negative. This means that θ_4 and θ_6 must be moved to the opposite quadrant by adding or subtracting 180°.

    - Let $0 < \theta_5 \leq 90°$, where θ_5 is determined by an inverse cosine. Calculate θ_4 and θ_6, from the matrix equation, checking that they fall within the manipulator bounds.

    $$\begin{aligned}\theta_4 &= \operatorname{atan2}\left(f_{32}(a), f_{31}(a)\right) \\ \theta_6 &= \operatorname{atan2}\left(f_{33}(o), -f_{33}(n)\right)\end{aligned}$$

    - Otherwise θ_5 is negative, so θ_4 and θ_6 are in the opposite quadrant from that calculated in the previous step.

12. Make sure that all joints calculated fall within the mechanical limitations of the robot.

4 Conclusion

This paper develops a complete tutorial example of a kinematic solution for a specific robot arm, one comprising six joints with one mid-way along the forearm. Degeneracies and the correspondence between real and modeled joint axes are handled in a realistic manner.

[13]That is, provided that joint limitations do not come into play.

Acknowledgements

This work is supported by the Natural Sciences and Engineering Research Council of Canada, and by a Province of Alberta scholarship. Thanks to David Pauli for commenting on a draft of this paper.

References

[Excalibur 1986] Excalibur user's manual. RSI Robotic Systems International, 1986.

[Grossman 1987] S. I. Grossman. *Elementary linear algebra.* Wadsworth Publishing Company, 1987.

[Paul 1981] R. P. Paul. *Robot manipulators: mathematics, programming, and control.* The MIT Press, 1981.