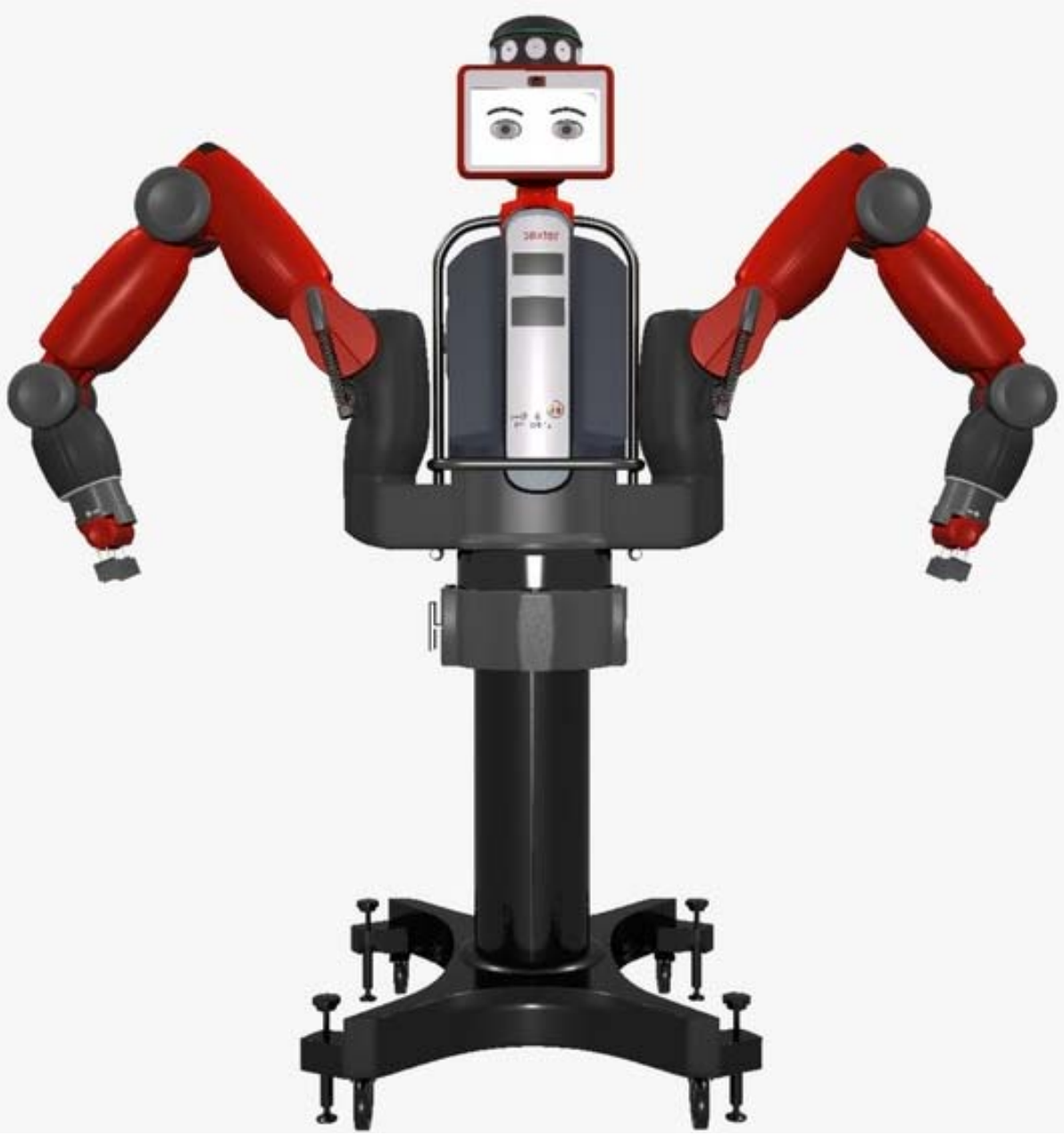
Baxter Humanoid Robot Kinematics

Robert L. Williams II, Ph.D., williar4@ohio.edu

Mechanical Engineering, Ohio University, April 2017



For referencing this document, please use:

R.L. Williams II, "Baxter Humanoid Robot Kinematics", Internet Publication, https://www.ohio.edu/mechanical-faculty/williams/html/pdf/BaxterKinematics.pdf, April 2017.

Baxter Humanoid Robot Kinematics

Robert L. Williams II, Ph.D., williar4@ohio.edu
Mechanical Engineering, Ohio University, Athens, Ohio, USA

Table of Contents

1. INTRODUCTION 3

2. BAXTER ROBOT SYSTEM DESCRIPTION 4

3. BAXTER ROBOT SYSTEM DENAVIT-HARTENBERG (DH) PARAMETERS 7

4. BAXTER FORWARD POSE KINEMATICS 13

4.1 Seven-dof Left Arm FPK Expressions 13

4.2 Six-dof Left Arm FPK Expressions 19

5. BAXTER ROBOT SYSTEM WORKSPACE 23

6. BAXTER INVERSE POSE KINEMATICS 28

6.1 Analytical IPK Solution for the 6-dof Baxter Left Arm 28

6.2 Numerical IPK Solution for the General 7-dof Baxter Left Arm 33

7. BAXTER VELOCITY KINEMATICS AND RESOLVED-RATE CONTROL 37

7.1 Baxter Robot Jacobian Matrix 37

7.2 Resolved-Rate Control Method for Kinematically-Redundant Arms 48

8. BAXTER ROBOT ARM KINEMATICS EXAMPLES 51

9. CONCLUSION 68

REFERENCES 68

1. Introduction

The Baxter Robot System is a human-sized humanoid robot with dual 7-degree-of-freedom (dof) arms with stationary pedestal, torso, and 2-dof head, a vision system, a robot control system, a safety system, and an optional gravity-offload controller and collision detection routine. This document presents kinematics and dynamics equations for control of this robot.

The Baxter arm design is rather traditional, similar to many previous 7-dof robot arms. 7-dof robot arms are classified as kinematically-redundant, i.e. possessing more joint freedoms than necessary to operate fully in the desired Cartesian space. Specifically, Baxter has $n = 7$ single-dof revolute (**R**) joints, which is one greater than the $m = 6$ Cartesian dof (3 translations and 3 rotations) for general trajectories. When $n > m$, the robot qualifies as kinematically-redundant, which means that in addition to reaching general desired 6-dof Cartesian trajectories, a 7-dof robot arm can also be used for optimizing robot arm performance at the same time.

The Baxter arm is rather unique considering its actuation by Serial Elastic Actuators (SEAs). All seven DC servomotor actuators for each arm include a flexible torsional spring in each drive shaft. This allows a naturally-compliant arm which will give when interacting with objects and human beings in its workspace. Baxter is a coBot which is designed specifically to interact safely with humans, unlike traditional robot manipulators. This safety is provided both by the mechanical design (SEAs) and also software safeguards. A side benefit of using the SEAs is that torque sensing is easily available for each of the active joints, which is great for statics and dynamics control.

Unlike some previous 7-dof robot arms, Baxter suffers with three unfortunate offsets in the kinematic structure – this greatly complicates the analytical kinematics and dynamics equations. I suppose this means that strictly numerical control has taken over in the current generation, which is too bad considering physical insights are lost this way. The Baxter 7-dof arms neither have a spherical shoulder nor a spherical wrist (i.e. nowhere are there 3 consecutive coordinate frames meeting at the same origin). Actually, the Baxter designers consider a 2-dof shoulder, a 2-dof elbow, and a 3-dof wrist. The elbow also has a kinematic offset. According to Pieper's principle, if a 6-dof serial robot has 3 consecutive coordinate frames meeting at the same origin, then an analytical solution is guaranteed to exist for the coupled nonlinear inverse pose kinematics problem. This does not occur for Baxter, so I suspect no analytical solutions exist for IPK (even for a simplified 6-dof Baxter with the first elbow joint locked to zero) – even if they did they would be significantly complicated due to the three offsets mentioned previously. Why do the designers include these pesky offsets? Why is the wrist of the terrible design wherein the singularity is in the middle of the useful rotational workspace (such as in the conventional Puma industrial robot)? Do the offsets make for a better workspace? Do they move the singularities around effectively (they cannot eliminate such singularities)? Or are the offsets just nice for each electromechanical design? This paper intends to answer some of these questions.

Presented is a description of the Baxter Robot System, followed by kinematics analysis and equations including Forward Pose Kinematics (FPK) and Inverse Pose Kinematics (IPK) expressions and solutions. Numerical examples are given for both FPK and IPK with both snapshots and trajectories. The velocity equations are also derived and used in a resolved-rate control scheme which has many advantages over IPK-based control. Singularity analysis is also presented. Then dynamics . . .

2. Baxter Robot System Description

The Baxter Humanoid Robot System is shown in Figure 1, in the standard pose with all zero joint angles (except for both left/right arm θ_1 S_0 shoulder joint angles – this is discussed later). Actually, the left and right wrist pitch joints W_1 are shown in two configurations, the horizontal ones with $\theta_6 = 0$ and the vertical down ones with $\theta_6 = +90^\circ$. In the hardware shown in Figure 1, each arm has seven single-dof revolute (**R**) joints, and the pan/tilt head has two single-dof **R** joints. The head panning is continuous, but the head nodding is on/off (up/down). There is an animated face on a flat screen. Though this is a humanoid robot system, there are no legs for locomotion and the torso is stationary. Therefore, the overall robot has 16-dof, not counting any gripper freedoms.

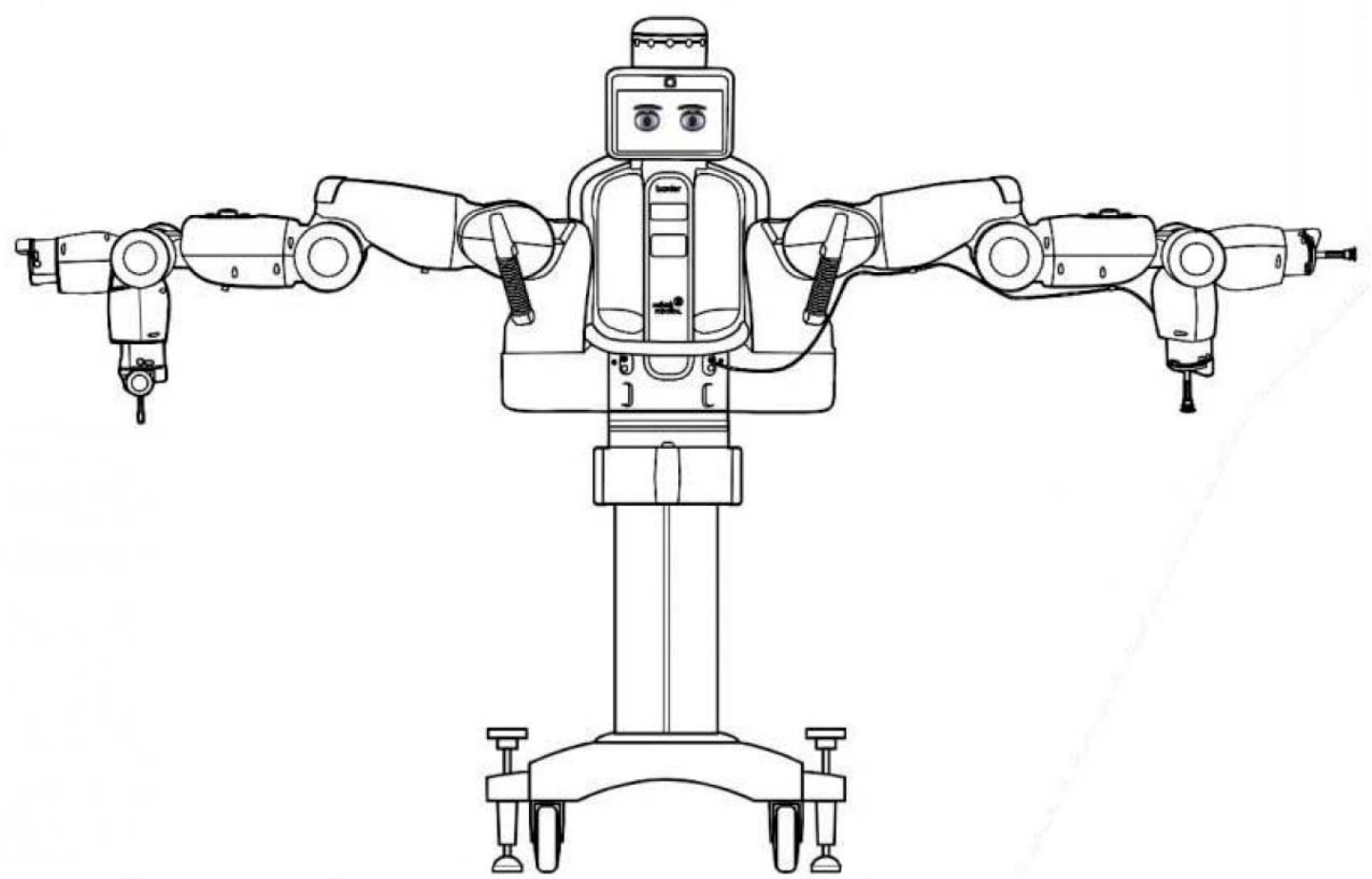


Figure 1. Baxter Humanoid Robot, Zero Pose
(Rethink Robotics, 2016)

Each 7-dof arm has a 2-dof (offset-**U**-joint) shoulder joint, a 2-dof (offset-**U**-joint) elbow joint, and a 3-dof (offset-**S**-joint) wrist joint. There are no parallel **R** joint axes anywhere on each arm, nor a series of three consecutive **R**-joints sharing a common origin (either/both of which would significantly simplify the kinematics and dynamics equations). The 2-dof head (**U**-joint with continuous pan and discrete up/down nod) enables pan and tilt for the camera/face.

Figure 2. Baxter 7-dof Left Arm R Joints (Rethink Robotics, 2016)

Table 1. Seven-dof Left Arm R Joints Naming Convention

Joint Name	Joint Motion
S_0	shoulder roll
S_1	shoulder pitch
E_0	elbow roll
E_1	elbow pitch
W_0	wrist roll
W_1	wrist pitch
W_2	wrist roll

Baxter Robot System Technical Specifications (Rethink Robotics, 2016)

Baxter is about 3’ tall (around 6’ tall with stationary pedestal) and weighs 165 lbs. (306 lbs. including the pedestal). Baxter has a 103” ‘wingspan’ and a 32” x 36” pedestal base.

Both 7-dof arms include angle position and joint torque sensing. For Cartesian sensors, there are three integrated cameras, plus sonar, accelerometers and range-finding sensors. Each Baxter arm has a temperature sensor, allowing human fingers to be detected for lead-through programming and other applications. The Baxter Robot System (research version) allows programming via a standard, open-source ROS API interface.

The first four active joints (the shoulder and elbow) of each 7-dof arm have a peak torque of 50 Nm, while the three wrist joints have a peak torque of 15 Nm. The whole-workspace accuracy is published to be ± 5 mm (which can be improved to ± 0.5 mm to certain unspecified limited portions of the workspace. The maximum payload, including the end-effector in the safety-enabled mode, is 2.3 kg. This increases to about 25 kg with safety disabled.

The joint sensor resolution for each of the 7-dof arm joints, right and left, is 14 bits for 360°, which works out to 0.02197° per encoder count.

The onboard computer consists of a third-generation Intel Core i7-3770 8MB 3.4 GHz processor with HD4000 Graphics, 4GB 1600 MHz DDR3 memory, and 128 GB solid state hard drive. The camera has a maximum resolution of 1280 x 800 pixels (640 x 400 pixels effective resolution), with a 30 fps frame rate and 1.2 mm focal length. The animated face flat screen has a resolution of 1024 x 600 pixels.

3. Baxter Robot System Denavit-Hartenberg (DH) Parameters

The modified Denavit-Hartenberg (DH, Denavit and Hartenberg, 1955) parameters are presented in this section, for each of the serial chains (two 7-dof arms, 2-dof head) for the Baxter Robot System. The Denavit-Hartenberg (DH) Parameters (1955) are used to describe the links/joints geometry of a serial-chain robot. DH parameters have been adopted for standard kinematics analysis in serial-chain robots (Craig, 2005). The community has come to call Craig-style DH Parameters as 'modified', with the original DH Parameters interpretation by Paul as 'standard'. The modified DH parameters have certain advantages over the standard (the main one being that a Craig coordinate frame rotates right at it joint, rather than distal from the joint as in Paul).

Seven-dof Left Arm

The Cartesian reference frame definitions for Baxter's 7-dof left arm are shown in Figure 3. Table 2 gives the associated DH parameters (Craig convention, known as 'modified DH parameters') for the 7-dof left arm.

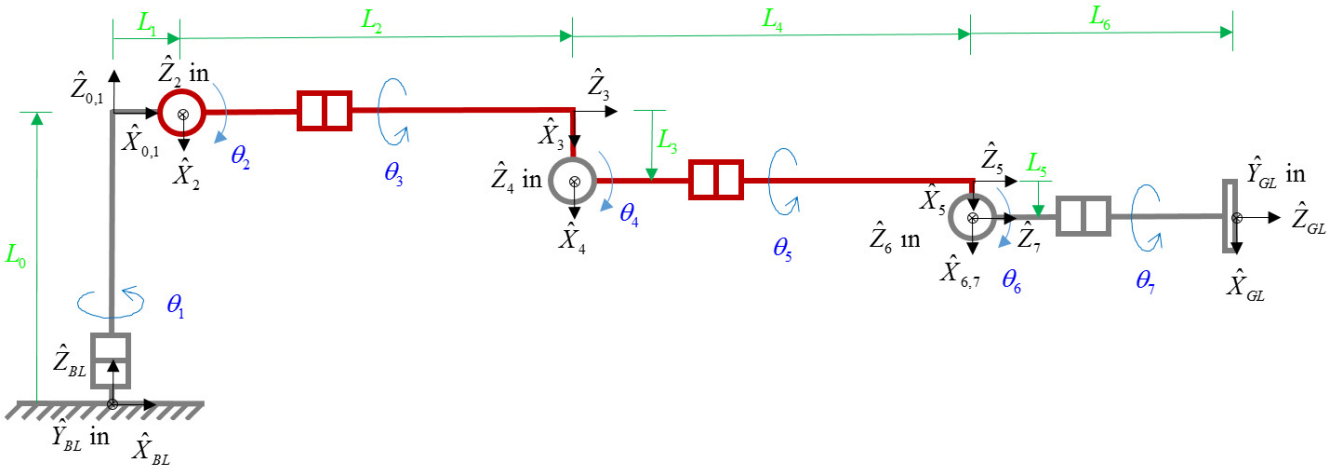


Figure 3. Seven-dof Left Arm Kinematic Diagram with Coordinate Frames

Table 2. Seven-dof Left Arm DH Parameters

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	-90°	L_1	0	$\theta_2 + 90^\circ$
3	90°	0	L_2	θ_3
4	-90°	L_3	0	θ_4
5	90°	0	L_4	θ_5
6	-90°	L_5	0	θ_6
7	90°	0	0	θ_7

Seven-dof Right Arm

The Cartesian reference frame definitions for Baxter's 7-dof right arm are shown in Figure 4. Table 3 gives the associated DH parameters (Craig convention, known as 'modified DH parameters') for the 7-dof right arm.

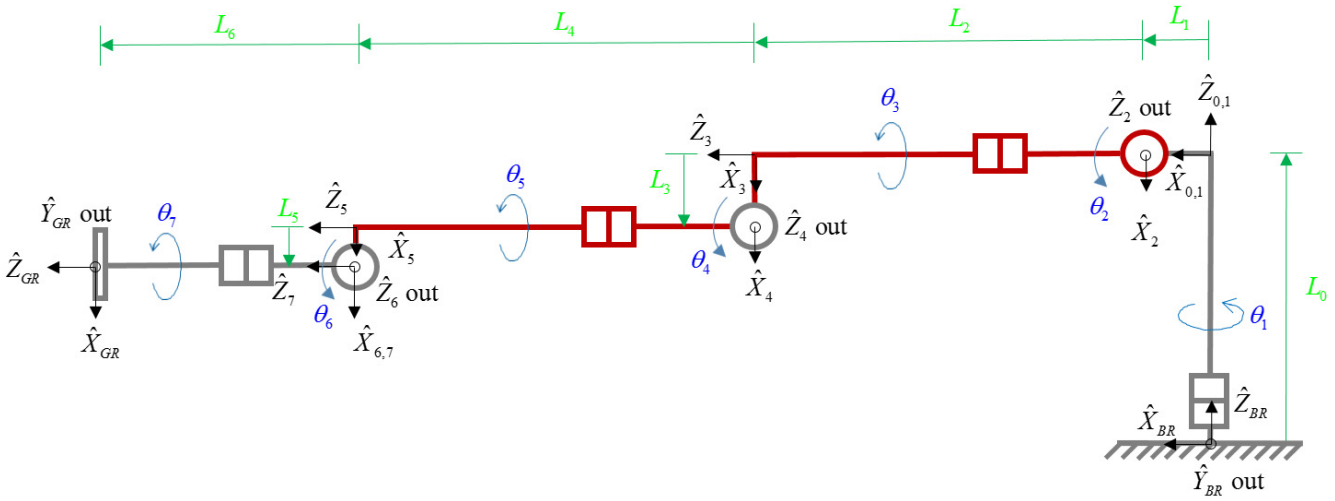


Figure 4. Seven-dof Right Arm Kinematic Diagram with Coordinate Frames

Table 3. Seven-dof Right Arm DH Parameters

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	-90°	L_1	0	$\theta_2 + 90^\circ$
3	90°	0	L_2	θ_3
4	-90°	L_3	0	θ_4
5	90°	0	L_4	θ_5
6	-90°	L_5	0	θ_6
7	90°	0	0	θ_7

Note that the Baxter robot system was designed so that the DH Parameters are identical for the left and right arms (compare Tables 2 and 3). This is a great aspect, so that all kinematics and dynamics derivations for the left arm apply equally to the right arm! Of course, this is with regard to the left and right base coordinate frames $\{BL\}$ and $\{BR\}$; poses in the world coordinate frame $\{WO\}$ are easily handled via homogeneous transformation matrices, presented later.

Note that the kinematic models and DH Parameters presented above agree with the as-delivered Baxter left- and right-arms with regard to zero location for all angles and also actual $\pm$ joint motion conventions. Also note that no calibration procedure is necessary for Baxter upon power-up; instead, all 14 joint encoders have known absolute zero locations.

The joint angle limits for the Baxter Robot System 7-dof arms are given in Table 4 below. Refer to the joint numbering given in Figures 3 and 4. Note all units in Table 4 are degrees.

Table 4. Seven-dof Left- and Right-Arm Joint Limits

Joint Name	Joint Variable	θ_i min	θ_i max	θ_i range
S_0	θ_1	$+51^\circ$	-141°	192°
S_1	θ_2	$+60^\circ$	-123°	183°
E_0	θ_3	$+173^\circ$	-173°	346°
E_1	θ_4	$+150^\circ$	-3°	153°
W_0	θ_5	$+175^\circ$	-175°	350°
W_1	θ_6	$+120^\circ$	-90°	210°
W_2	θ_7	$+175^\circ$	-175°	350°

Table 5. Seven-dof Left- and Right-Arm Link- and Offset-Lengths

Length	**Value (mm)**
L_0	270.35
L_1	69.00
L_2	364.35
L_3	69.00
L_4	374.29
L_5	10.00
L_6	368.30

Note: L_6 is the length from the wrist pitch center to the center of the parallel-jaw gripper fingers – this dimension is different to the gripper plate (229.53 mm) and to the tip of the grippers (387.35 mm).

Further note: in MATLAB simulation and real-time Baxter Robot programming we will use m, not mm, for the length units. This will better avoid numerical imbalances between the translational and rotational terms.

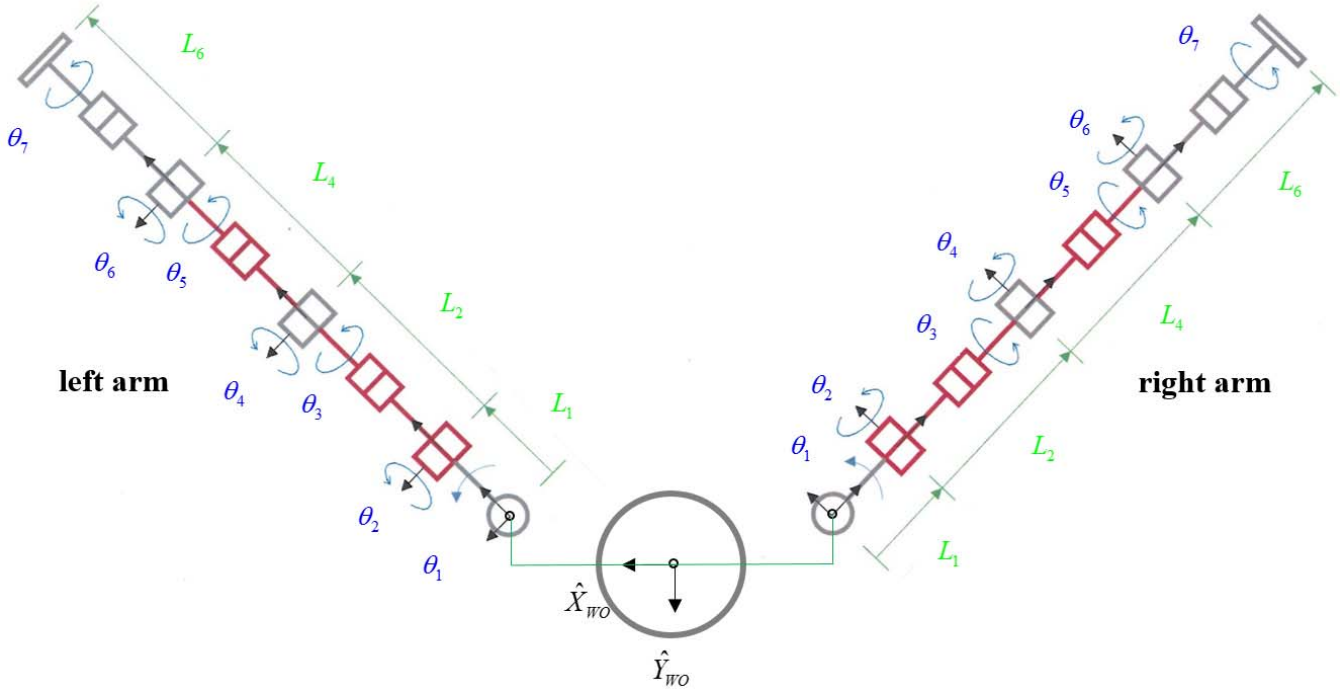


Figure 5a. Baxter Left- and Right-Arm Kinematic Diagrams, Top View
Zero Joint Angles Shown
(not to scale)

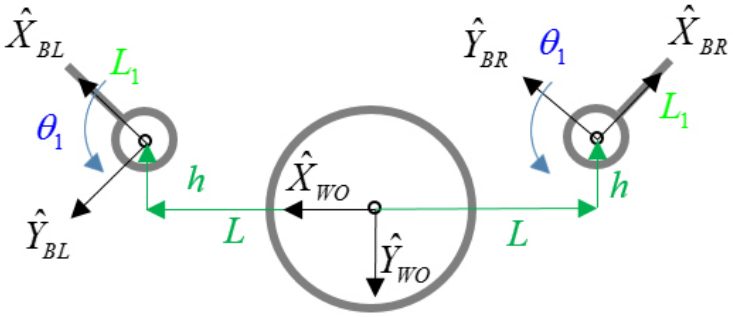


Figure 5b. Top View Details

As shown in Figure 5a, the zero locations for joints left/right arm θ_1 S_0 shoulder joint angles are not straight to the sides as indicated in Figure 1, but rather at the 45° angles seen above. Again, the kinematics and dynamics equations analytical terms will be identical for the left and right arms with respect to their respective base frames. However, the absolute terms must be transformed into a common frame such as $\{WO\}$. The required homogeneous transformation matrices are given below, by inspection.

Figure 5b shows the two length offsets L and h, for the joint center of each S_0 shoulder joint angles with respect to the origin of $\{Wo\}$ in the top view, which is located at the center of the torso circle, on the floor. The height from the floor to the level of the $\{B\}$ frames for the left and right arms is H (see Figure 6). The fixed homogeneous transformations giving the pose of each left/right arm $\{B\}$ frame with respect to the world frame $\{Wo\}$ are given below.

$$\left[{}^{Wo}_{BL}T\right]=\begin{bmatrix}\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & L\\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & -h\\ 0 & 0 & 1 & H\\ 0 & 0 & 0 & 1\end{bmatrix} \qquad \left[{}^{Wo}_{BR}T\right]=\begin{bmatrix}-\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & -L\\ -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 & -h\\ 0 & 0 & 1 & H\\ 0 & 0 & 0 & 1\end{bmatrix}$$

These required left/right arm lengths for the {*B*} frames to {*Wo*} frame transforms are given in Table 6.

Table 6. {*B*} to {*Wo*} Lengths

Length	Value (mm)
L	278
h	64
H	1104

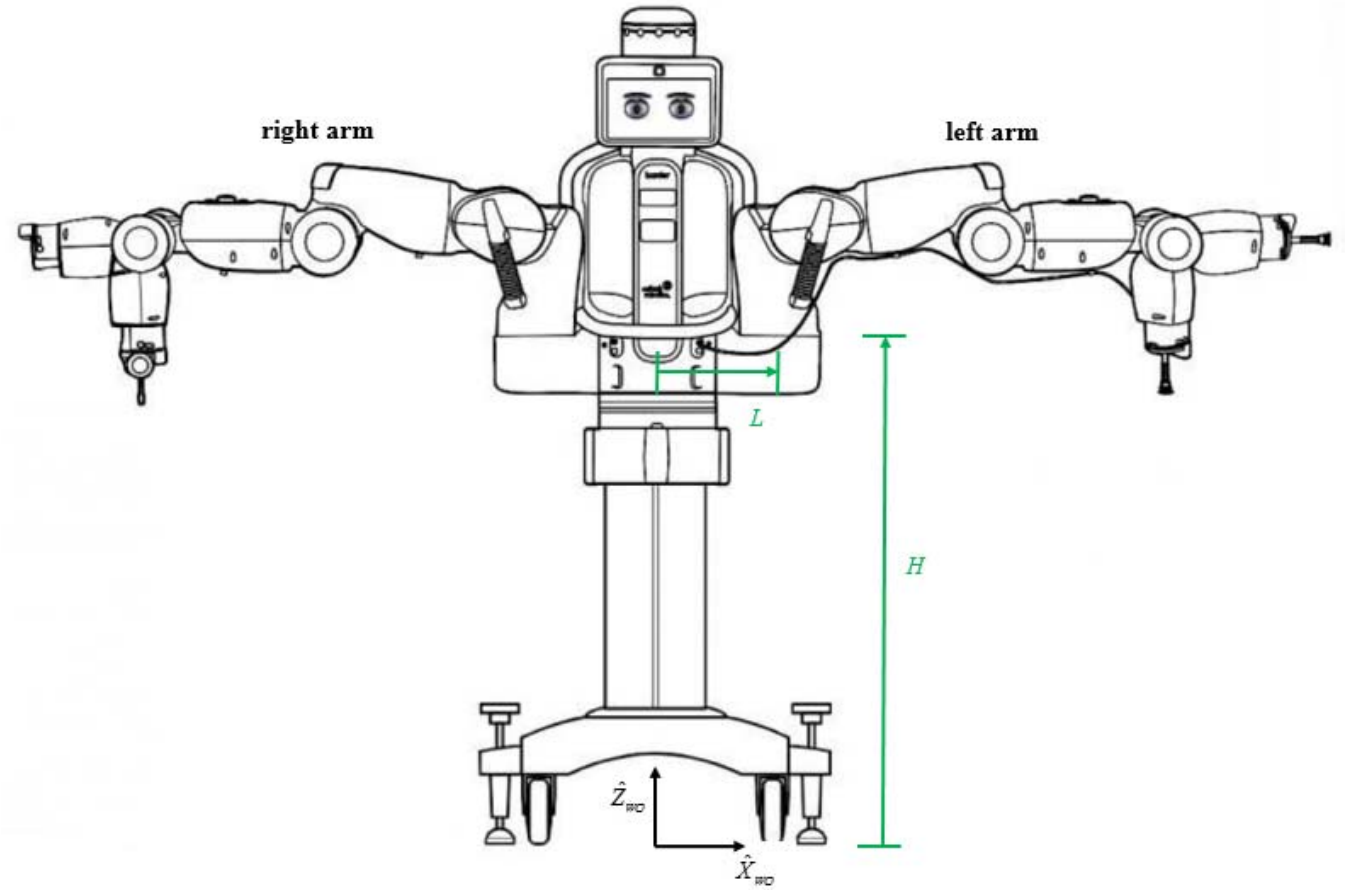


Figure 6. Baxter Front View

The robot arm configurations shown in Figures 3, 4, and 5 are for all zero joint angles. Baxter's 'neutral' location joint angles are given in Table 7 (these angles are identical for the left and right arms). A MATLAB rendition of the neutral joint angles pose is given in the Examples later.

Table 7. Baxter 'Neutral' Joint Angles

Joint Name	**Joint Variable**	θ_i
S_0	θ_1	0°
S_1	θ_2	-31°
E_0	θ_3	0°
E_1	θ_4	43°
W_0	θ_5	0°
W_1	θ_6	72°
W_2	θ_7	0°

4. Baxter Forward Pose Kinematics

In general, the Forward Pose Kinematics (FPK) problem for a serial-chain robot is stated: Given the joint values, calculate the pose (position and orientation) of the end-effector frame of interest. For serial-chain robots, the FPK problem set up and solution is straight-forward. It is based on substituting each line of the Denavit-Hartenberg Parameters Table 2 into the equation below (Craig, 2005), giving the pose of frame $\{i\}$ with respect to its nearest neighbor frame $\{i–1\}$ back along the serial chain:

$$\left[{}^{i-1}_{i}T\right] = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -d_i s\alpha_{i-1} \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & d_i c\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \left[{}^{i-1}_{i}R\right] & \left\{{}^{i-1}P_i\right\} \\ 0 \quad 0 & 0 \quad 1 \end{bmatrix}$$

Where the following abbreviations were used: $c\theta_i = \cos\theta_i$, $s\theta_i = \sin\theta_i$, $c\alpha_i = \cos\alpha_i$, and $s\alpha_i = \sin\alpha_i$.

The equation above represents pose (position and orientation) of frame $\{i\}$ with respect to frame $\{i–1\}$ by using a 4x4 homogeneous transformation matrix. The upper left 3x3 matrix is the rotation matrix $\left[{}^{i-1}_{i}R\right]$ giving the orientation of frame $\{i\}$ with respect to frame $\{i–1\}$, expressed in $\{i–1\}$ coordinates. The upper right 3x1 vector $\left\{{}^{i-1}P_i\right\}$ is the position vector from the origin of $\{i–1\}$ to the origin of $\{i\}$, expressed in $\{i–1\}$ coordinates.

Then homogeneous transformation equations are used to find the pose of the overall end-effector frame of interest with respect to the base reference frame, to complete the FPK solution for each serial chain.

4.1 Seven-dof Left Arm FPK Expressions

The statement of the FPK problem for the seven-dof left arm serial chain of the Baxter humanoid robot is:

$$\text{Given } (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7), \text{ calculate } \left[{}^{0}_{7}T\right] \text{ and } \left[{}^{W}_{G}T\right].$$

where $\{G\}$ is the left-arm end-effector (gripper) frame and $\{W\}$ is the World fixed reference frame on the floor. For notational simplicity in Cartesian coordinate frame definition, frame numbers $\{0\}$, $\{1\}$, … $\{7\}$, etc. will be recycled for both arm serial chains. Therefore, in this paper some repeated frame numbers are context-dependent, which must be sorted out in programming the Baxter humanoid robot.

Substitute each row of the DH parameters in Table 2 into the equation for $\left[{}^{i-1}_{i}T\right]$ to obtain the seven neighboring homogeneous transformation matrices as a function of the joint angles for the 7-dof left arm.

$$\left[{}^{0}_{1}T\right]=\begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \left[{}^{1}_{2}T\right]=\begin{bmatrix} -s_2 & -c_2 & 0 & L_1 \\ 0 & 0 & 1 & 0 \\ -c_2 & s_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \left[{}^{2}_{3}T\right]=\begin{bmatrix} c_3 & -s_3 & 0 & 0 \\ 0 & 0 & -1 & -L_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\left[{}^{3}_{4}T\right]=\begin{bmatrix} c_4 & -s_4 & 0 & L_3 \\ 0 & 0 & 1 & 0 \\ -s_4 & -c_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \left[{}^{4}_{5}T\right]=\begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & -1 & -L_4 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \left[{}^{5}_{6}T\right]=\begin{bmatrix} c_6 & -s_6 & 0 & L_5 \\ 0 & 0 & 1 & 0 \\ -s_6 & -c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\left[{}^{6}_{7}T\right]=\begin{bmatrix} c_7 & -s_7 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_7 & c_7 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where the following abbreviations were used: $c_i = \cos\theta_i$, $s_i = \sin\theta_i$, for $i = 1, 2, \cdots, 7$.

Now substitute these seven neighboring homogeneous transformation matrices into the following homogeneous transform equation to derive the active-joints FPK result.

$$\left[{}^{0}_{7}T\right]=\left[{}^{0}_{1}T(\theta_1)\right]\left[{}^{1}_{2}T(\theta_2)\right]\left[{}^{2}_{3}T(\theta_3)\right]\left[{}^{3}_{4}T(\theta_4)\right]\left[{}^{4}_{5}T(\theta_5)\right]\left[{}^{5}_{6}T(\theta_6)\right]\left[{}^{6}_{7}T(\theta_7)\right]$$

Note this left-arm active-joint FPK solution may be grouped as follows, by the 2-dof shoulder, 2-dof elbow, and 3-dof wrist joints.

$$\left[{}^{0}_{7}T(\theta_1,\theta_2,\theta_3,\theta_4,\theta_5,\theta_6,\theta_7)\right]=\left[{}^{0}_{2}T(\theta_1,\theta_2)\right]\left[{}^{2}_{4}T(\theta_3,\theta_4)\right]\left[{}^{4}_{7}T(\theta_5,\theta_6,\theta_7)\right]$$

where:

$$\left[T_{\text{shoulder}}\right]=\left[{}^{0}_{2}T(\theta_1,\theta_2)\right]=\begin{bmatrix} -c_1s_2 & -c_1c_2 & -s_1 & L_1c_1 \\ -s_1s_2 & -s_1c_2 & c_1 & L_1s_1 \\ -c_2 & s_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\left[T_{\text{elbow}}\right]=\left[{}^{2}_{4}T(\theta_3,\theta_4)\right]=\begin{bmatrix} c_3c_4 & -c_3s_4 & -s_3 & L_3c_3 \\ s_4 & c_4 & 0 & -L_2 \\ s_3c_4 & -s_3s_4 & c_3 & L_3s_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\left[T_{\text{wrist}}\right]=\left[{}^{4}_{7}T(\theta_5,\theta_6,\theta_7)\right]=\begin{bmatrix} -s_5s_7+c_5c_6c_7 & -s_5c_7-c_5c_6s_7 & c_5s_6 & L_5c_5 \\ s_6c_7 & -s_6s_7 & -c_6 & -L_4 \\ c_5s_7+s_5c_6c_7 & c_5c_7-s_5c_6s_7 & s_5s_6 & L_5s_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Further, combining the 2-dof shoulder and 2-dof elbow homogeneous transformation matrices via matrix multiplication:

$$\left[{}^{0}_{7}T(\theta_1,\theta_2,\theta_3,\theta_4,\theta_5,\theta_6,\theta_7)\right]=\left[{}^{0}_{4}T(\theta_1,\theta_2,\theta_3,\theta_4)\right]\left[{}^{4}_{7}T(\theta_5,\theta_6,\theta_7)\right]$$

where:

$$\left[T_{\text{shoulder/elbow}}\right]=\left[{}^{0}_{2}T(\theta_1,\theta_2)\right]\left[{}^{2}_{4}T(\theta_3,\theta_4)\right]=\left[{}^{0}_{4}T(\theta_1,\theta_2,\theta_3,\theta_4)\right]$$

$$=\begin{bmatrix} -(s_1s_3+c_1s_2c_3)c_4-c_1c_2s_4 & -c_1c_2c_4+(s_1s_3+c_1s_2c_3)s_4 & -s_1c_3+c_1s_2s_3 & (L_1+L_2c_2)c_1-L_3(s_1s_3+c_1s_2c_3) \\ (c_1s_3-s_1s_2c_3)c_4-s_1c_2s_4 & -s_1c_2c_4-(c_1s_3-s_1s_2c_3)s_4 & c_1c_3+s_1s_2s_3 & (L_1+L_2c_2)s_1+L_3(c_1s_3-s_1s_2c_3) \\ s_2s_4-c_2c_3c_4 & s_2c_4+c_2c_3s_4 & c_2s_3 & -L_2s_2-L_3c_2c_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The overall analytical FPK expressions for the active Baxter joints are now given:

$$\left[{}^{0}_{7}T(\theta_1,\cdots,\theta_7)\right]=\begin{bmatrix} r_{11} & r_{12} & r_{13} & {}^{0}x_7 \\ r_{21} & r_{22} & r_{23} & {}^{0}y_7 \\ r_{31} & r_{32} & r_{33} & {}^{0}z_7 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The orthonormal rotation matrix elements for this result are:

$$r_{11}=((as_4-c_1c_2c_4)s_6-(bs_5+(ac_4+c_1c_2s_4)c_5)c_6)c_7+((ac_4+c_1c_2s_4)s_5-bc_5)s_7$$
$$r_{12}=((ac_4+c_1c_2s_4)s_5-bc_5)c_7+((as_4-c_1c_2c_4)s_6-(bs_5+(ac_4+c_1c_2s_4)c_5)c_6)s_7$$
$$r_{13}=-(as_4-c_1c_2c_4)c_6-(bs_5+(ac_4+c_1c_2s_4)c_5)s_6$$

$$r_{21}=-((ds_4+s_1c_2c_4)s_6-(fs_5+(dc_4-s_1c_2s_4)c_5)c_6)c_7-((dc_4-s_1c_2s_4)s_5-fc_5)s_7$$
$$r_{22}=-((dc_4-s_1c_2s_4)s_5-fc_5)c_7-((ds_4+s_1c_2c_4)s_6-(fs_5+(dc_4-s_1c_2s_4)c_5)c_6)s_7$$
$$r_{23}=(ds_4+s_1c_2c_4)c_6+(fs_5+(dc_4-s_1c_2s_4)c_5)s_6$$

$$r_{31} = (hs_6 + (gc_5 + c_2 s_3 s_5)c_6)c_7 - (gs_5 - c_2 s_3 c_5)s_7$$
$$r_{32} = -(gs_5 - c_2 s_3 c_5)c_7 - (hs_6 + (gc_5 + c_2 s_3 s_5)c_6)s_7$$
$$r_{33} = -hc_6 + (gc_5 + c_2 s_3 s_5)s_6$$

where:

$$a = s_1 s_3 + c_1 s_2 c_3$$
$$b = s_1 c_3 - c_1 s_2 s_3$$
$$d = c_1 s_3 - s_1 s_2 c_3$$
$$f = c_1 c_3 + s_1 s_2 s_3$$
$$g = s_2 s_4 - c_2 c_3 c_4$$
$$h = s_2 c_4 + c_2 c_3 s_4$$

Applying another level of substitutions:

$$A = as_4 - c_1 c_2 c_4$$
$$B = ac_4 + c_1 c_2 s_4$$
$$D = ds_4 + s_1 c_2 c_4$$
$$F = dc_4 - s_1 c_2 s_4$$
$$G = gs_5 - c_2 s_3 c_5$$
$$H = gc_5 + c_2 s_3 s_5$$

the same orthonormal rotation matrix elements are:

$$r_{11} = (As_6 - (bs_5 + Bc_5)c_6)c_7 + (Bs_5 - bc_5)s_7$$
$$r_{12} = (Bs_5 - bc_5)c_7 + (As_6 - (bs_5 + Bc_5)c_6)s_7$$
$$r_{13} = -Ac_6 - (bs_5 + Bc_5)s_6$$

$$r_{21} = -(Ds_6 - (fs_5 + Fc_5)c_6)c_7 - (Fs_5 - fc_5)s_7$$
$$r_{22} = -(Fs_5 - fc_5)c_7 - (Ds_6 - (fs_5 + Fc_5)c_6)s_7$$
$$r_{23} = Dc_6 + (fs_5 + Fc_5)s_6$$

$$r_{31} = (hs_6 + Hc_6)c_7 - Gs_7$$
$$r_{32} = -Gc_7 - (hs_6 + Hc_6)s_7$$
$$r_{33} = -hc_6 + Hs_6$$

The translational terms for this result are:

$$
\begin{aligned}
{}^0x_7 = {} & L_1c_1 + L_2c_1c_2 - L_3(s_1s_3 + c_1s_2c_3) - L_4((s_1s_3 + c_1s_2c_3)s_4 - c_1c_2c_4) \\
& - L_5((s_1c_3 - c_1s_2s_3)s_5 + ((s_1s_3 + c_1s_2c_3)c_4 + c_1c_2s_4)c_5)
\end{aligned}
$$

$$
\begin{aligned}
{}^0y_7 = {} & L_1s_1 + L_2s_1c_2 + L_3(c_1s_3 - s_1s_2c_3) + L_4((c_1s_3 - s_1s_2c_3)s_4 + s_1c_2c_4) \\
& + L_5((c_1c_3 + s_1s_2s_3)s_5 + ((c_1s_3 - s_1s_2c_3)c_4 - s_1c_2s_4)c_5)
\end{aligned}
$$

$$
{}^0z_7 = -L_2s_2 - L_3c_2c_3 - L_4(s_2c_4 + c_2c_3s_4) + L_5((s_2s_4 - c_2c_3c_4)c_5 + c_2s_3s_5)
$$

substituting the $a - h$ terms defined above:

$$
{}^0x_7 = L_1c_1 + L_2c_1c_2 - L_3a - L_4(as_4 - c_1c_2c_4) - L_5(bs_5 + (ac_4 + c_1c_2s_4)c_5)
$$

$$
{}^0y_7 = L_1s_1 + L_2s_1c_2 + L_3d + L_4(ds_4 + s_1c_2c_4) + L_5(fs_5 + (dc_4 - s_1c_2s_4)c_5)
$$

$$
{}^0z_7 = -L_2s_2 - L_3c_2c_3 - L_4h + L_5(gc_5 + c_2s_3s_5)
$$

and substituting the $A - H$ terms defined above:

$$
{}^0x_7 = L_1c_1 + L_2c_1c_2 - L_3a - L_4A - L_5(bs_5 + Bc_5)
$$

$$
{}^0y_7 = L_1s_1 + L_2s_1c_2 + L_3d + L_4D + L_5(fs_5 + Fc_5)
$$

$$
{}^0z_7 = -L_2s_2 - L_3c_2c_3 - L_4h + L_5H
$$

Note that, since the origins of frames {6} and {7} are coincident at the wrist point, the translational terms above are only functions of the first five joint angles:

$$
\left\{ {}^0P_7 \right\} = \left\{ {}^0P_7(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) \right\} = \begin{Bmatrix} {}^0x_7 \\ {}^0y_7 \\ {}^0z_7 \end{Bmatrix}
$$

Of course, the translational terms involving the tool-plate (or gripper) frame $\{G\}$ with respect to the base, $\left\{ {}^0P_G \right\}$, are functions of all seven active joint angles. This is covered below.

Additional, Fixed Transforms – Gripper, Base, and World Frames

In addition, the following three homogeneous transformation matrices are required in the overall FPK solution. Note that these three matrices are not determined from any are not evaluated by any row in the DH parameter table (those were all used above), since there is no variable associated with these fixed homogeneous transformation matrices based on constant lengths and orientation DH parameters. Instead, they are determined by inspection, using the rotation matrix and position vector components of the homogeneous transformation matrix definition.

$$\left[{}^{BL}_{0}T \right] = \left[{}^{BR}_{0}T \right] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \left[{}^{7}_{GL}T \right] = \left[{}^{7}_{GR}T \right] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The overall 7-dof Baxter left-arm FPK solution involving both active joints and constant homogeneous transformations is:

$$\left[{}^{W}_{GL}T \right] = \left[{}^{W}_{BL}T \right] \left[{}^{BL}_{0}T \right] \left[{}^{0}_{7}T(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7) \right] \left[{}^{7}_{GL}T \right]$$

Matrices $\left[{}^{Wo}_{BL}T \right]$ and $\left[{}^{Wo}_{BR}T \right]$ were given earlier; these are different for the left and right arms. Note that the other two constant homogeneous transformation matrices are identical for the left and right arms.

The FPK solutions can be evaluated numerically or symbolically, or using a combination of these two methods.

4.2 Six-dof Left Arm FPK Expressions

In case one wishes to control a 6-dof (rather than the kinematically-redundant 7-dof) Baxter robot arm, it is logical to lock joint 3 to $\theta_3 = 0$. This case, pictured in the kinematic diagram of Figure 7 below, has the advantage that now-consecutive active joint angles θ_2 and θ_4 rotate about now-always-parallel axes $\hat{Z}_2$ and $\hat{Z}_4$. This will lead to simpler kinematics equations since we can take advantage of sum-of-angle formulas (cosines and sines of $(\theta_2 + \theta_4)$), not to mention that kinematics equations are no longer functions of θ_3. Further, this represents much simpler inverse kinematics problems (pose and/or velocity) since this is the $m = n = 6$ constrained case (which is far simpler than the $m = 6 < n = 7$ underconstrained kinematically-redundant case). m is the dimension of the Cartesian space and n is the dimension of the joint space (i.e. the number of 1-dof joints in the robot).

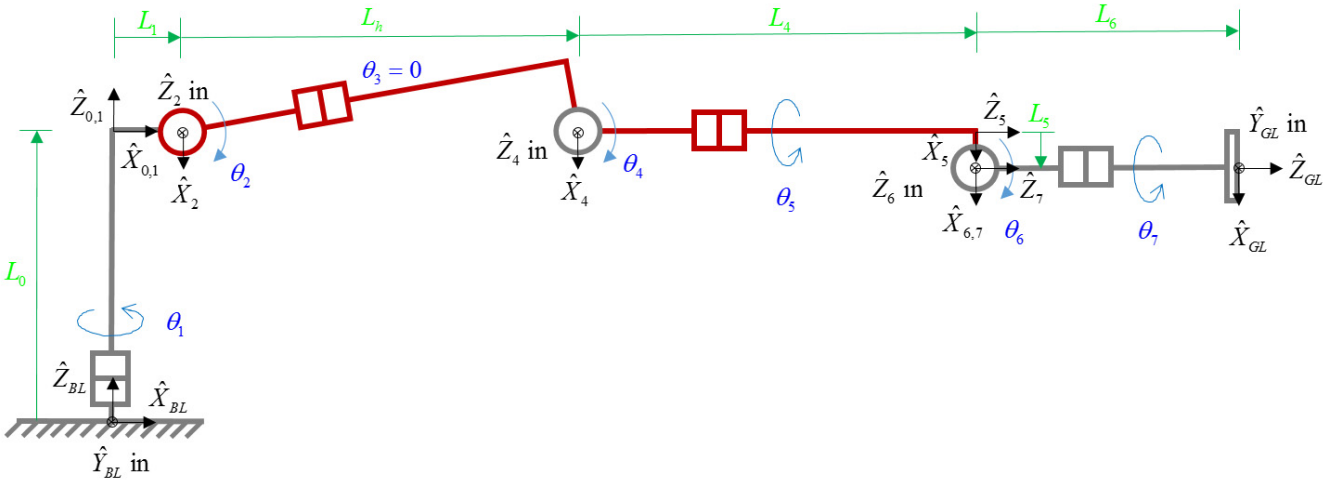


Figure 7. Six-dof Left Arm Kinematic Diagram with $\theta_3 = 0$

The Denavit-Hartenberg Parameters for this 6-dof case are given in Table 8 below. Please note that frame {3} is not included, so the numbering may be confusing at first. That is, $i = 3,4,5,6$ now actually correspond to frames 4,5,6,7. Also note that the introduced kinematic link length is $L_h = \sqrt{L_2^2 + L_3^2}$.

Table 8. Six-dof Left Arm DH Parameters

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	-90°	L_1	0	θ_2
3	0	L_h	0	$\theta_4 + 90^\circ$
4	90°	0	L_4	θ_5
5	-90°	L_5	0	θ_6
6	90°	0	0	θ_7

Note the link length L_2 in Figure 7 is no longer horizontal as it was in Figure 3. Instead it is inclined by the following θ_2 angular offset:

$$\theta_{2\text{off}} = \tan^{-1}\left[\frac{L_3}{L_2}\right] = 10.73^\circ$$

The statement of the FPK problem for the six-dof left arm serial chain of the Baxter humanoid robot is:

$$\text{Given } (\theta_1, \theta_2, \theta_4, \theta_5, \theta_6, \theta_7) \text{ and } \theta_3 = 0 \text{, calculate } \left[{}^0_6T\right] \text{ and } \left[{}^W_GT\right].$$

Again, note that the FPK results to follow drop frame {3} such that *i* = 3,4,5,6 corresponds to frames 4,5,6,7.

Substitute each row of the DH parameters in Table 8 into the equation for $\left[{}^{i-1}_iT\right]$ to obtain the six neighboring homogeneous transformation matrices as a function of the joint angles for the 6-dof left arm.

$$\left[{}^0_1T\right] = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \left[{}^1_2T\right] = \begin{bmatrix} c_2 & -s_2 & 0 & L_1 \\ 0 & 0 & 1 & 0 \\ -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \left[{}^2_3T\right] = \begin{bmatrix} -s_4 & -c_4 & 0 & L_h \\ c_4 & -s_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\left[{}^3_4T\right] = \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & -1 & -L_4 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \left[{}^4_5T\right] = \begin{bmatrix} c_6 & -s_6 & 0 & L_5 \\ 0 & 0 & 1 & 0 \\ -s_6 & -c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \left[{}^5_6T\right] = \begin{bmatrix} c_7 & -s_7 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_7 & c_7 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where the following abbreviations were used: $c_i = \cos\theta_i$, $s_i = \sin\theta_i$, for *i* = 1,2,4,5,6,7. Now substitute these six neighboring homogeneous transformation matrices into the following homogeneous transform equation to derive the active-joints FPK result.

$$\left[{}^0_6T\right] = \left[{}^0_1T(\theta_1)\right]\left[{}^1_2T(\theta_2)\right]\left[{}^2_3T(\theta_4)\right]\left[{}^3_4T(\theta_5)\right]\left[{}^4_5T(\theta_6)\right]\left[{}^5_6T(\theta_7)\right]$$

Note this left-arm active-joint FPK solution may be grouped as follows, by the 3-dof shoulder/elbow and 3-dof wrist joints.

$$\left[{}^0_6T(\theta_1, \theta_2, \theta_4, \theta_5, \theta_6, \theta_7)\right] = \left[{}^0_3T(\theta_1, \theta_2, \theta_4)\right]\left[{}^3_6T(\theta_5, \theta_6, \theta_7)\right]$$

Since joint 3 is locked to $\theta_3 = 0$, joint axes 2 and 4 rotate about parallel *Z* axes, which means sum-of-angles formulas may be applied to simplify the expressions significantly:

$$\left[T_{\text{shoulder/elbow}}\right] = \left[{}^{0}_{3}T(\theta_1,\theta_2,\theta_4)\right] = \begin{bmatrix} -c_1 s_{24} & -c_1 c_{24} & -s_1 & c_1(L_1 + L_h c_2) \\ -s_1 s_{24} & -s_1 c_{24} & c_1 & s_1(L_1 + L_h c_2) \\ -c_{24} & s_{24} & 0 & -L_h s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\left[T_{\text{wrist}}\right] = \left[{}^{3}_{6}T(\theta_5,\theta_6,\theta_7)\right] = \begin{bmatrix} -s_5 s_7 + c_5 c_6 c_7 & -s_5 c_7 - c_5 c_6 s_7 & c_5 s_6 & L_5 c_5 \\ s_6 c_7 & -s_6 s_7 & -c_6 & -L_4 \\ c_5 s_7 + s_5 c_6 c_7 & c_5 c_7 - s_5 c_6 s_7 & s_5 s_6 & L_5 s_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The following abbreviations were used: $c_i = \cos\theta_i$, $s_i = \sin\theta_i$, for *i* = 1,2,4,5,6,7, and $c_{24} = \cos(\theta_2 + \theta_4)$, $s_{24} = \sin(\theta_2 + \theta_4)$.

The overall 6-dof analytical FPK expressions for the active Baxter joints are now given:

$$\left[{}^{0}_{6}T(\theta_1,\theta_2,\theta_4,\theta_5,\theta_6,\theta_7)\right] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & {}^{0}x_6 \\ r_{21} & r_{22} & r_{23} & {}^{0}y_6 \\ r_{31} & r_{32} & r_{33} & {}^{0}z_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The orthonormal rotation matrix elements for this result are significantly simpler than the 7-dof case:

$$\begin{aligned} r_{11} &= -c_1(c_{24}s_6c_7 - s_{24}(s_5s_7 - c_5c_6c_7)) - s_1(c_5s_7 + s_5c_6c_7) \\ r_{12} &= \;\; c_1(c_{24}s_6s_7 + s_{24}(s_5c_7 + c_5c_6s_7)) - s_1(c_5c_7 - s_5c_6s_7) \\ r_{13} &= \;\; c_1(c_{24}c_6 - s_{24}c_5s_6) - s_1s_5s_6 \end{aligned}$$

$$\begin{aligned} r_{21} &= c_1(c_5s_7 + s_5c_6c_7) + s_1(-c_{24}s_6c_7 + s_{24}(s_5s_7 - c_5c_6c_7)) \\ r_{22} &= c_1(c_5c_7 - s_5c_6s_7) + s_1(\;\; c_{24}s_6s_7 + s_{24}(s_5c_7 + c_5c_6s_7)) \\ r_{23} &= c_1s_5s_6 + s_1(c_{24}c_6 - s_{24}c_5s_6) \end{aligned}$$

$$\begin{aligned} r_{31} &= c_{24}(s_5s_7 - c_5c_6c_7) + s_{24}s_6c_7 \\ r_{32} &= c_{24}(s_5c_7 + c_5c_6s_7) - s_{24}s_6s_7 \\ r_{33} &= -c_{24}c_5s_6 - s_{24}c_6 \end{aligned}$$

The translational terms for this result are also significantly simpler than the 7-dof equivalents:

$$^0x_6 = L_1c_1 + L_hc_1c_2 + L_4c_1c_{24} - L_5(s_1s_5 + c_1s_{24}c_5)$$

$$^0y_6 = L_1s_1 + L_hs_1c_2 + L_4s_1c_{24} + L_5(c_1s_5 - s_1s_{24}c_5)$$

$$^0z_6 = -L_hs_2 - L_4s_{24} - L_5c_{24}c_5$$

Note that, since the origins of frames {6} and {7} are coincident at the wrist point, the translational terms above are only functions of the first four joint angles (since now joint 3 is locked to $\theta_3 = 0$):

$$\left\{ {}^0P_6 \right\} = \left\{ {}^0P_6(\theta_1, \theta_2, \theta_4, \theta_5) \right\} = \begin{Bmatrix} {}^0x_6 \\ {}^0y_6 \\ {}^0z_6 \end{Bmatrix}$$

5. Baxter Robot System Workspace

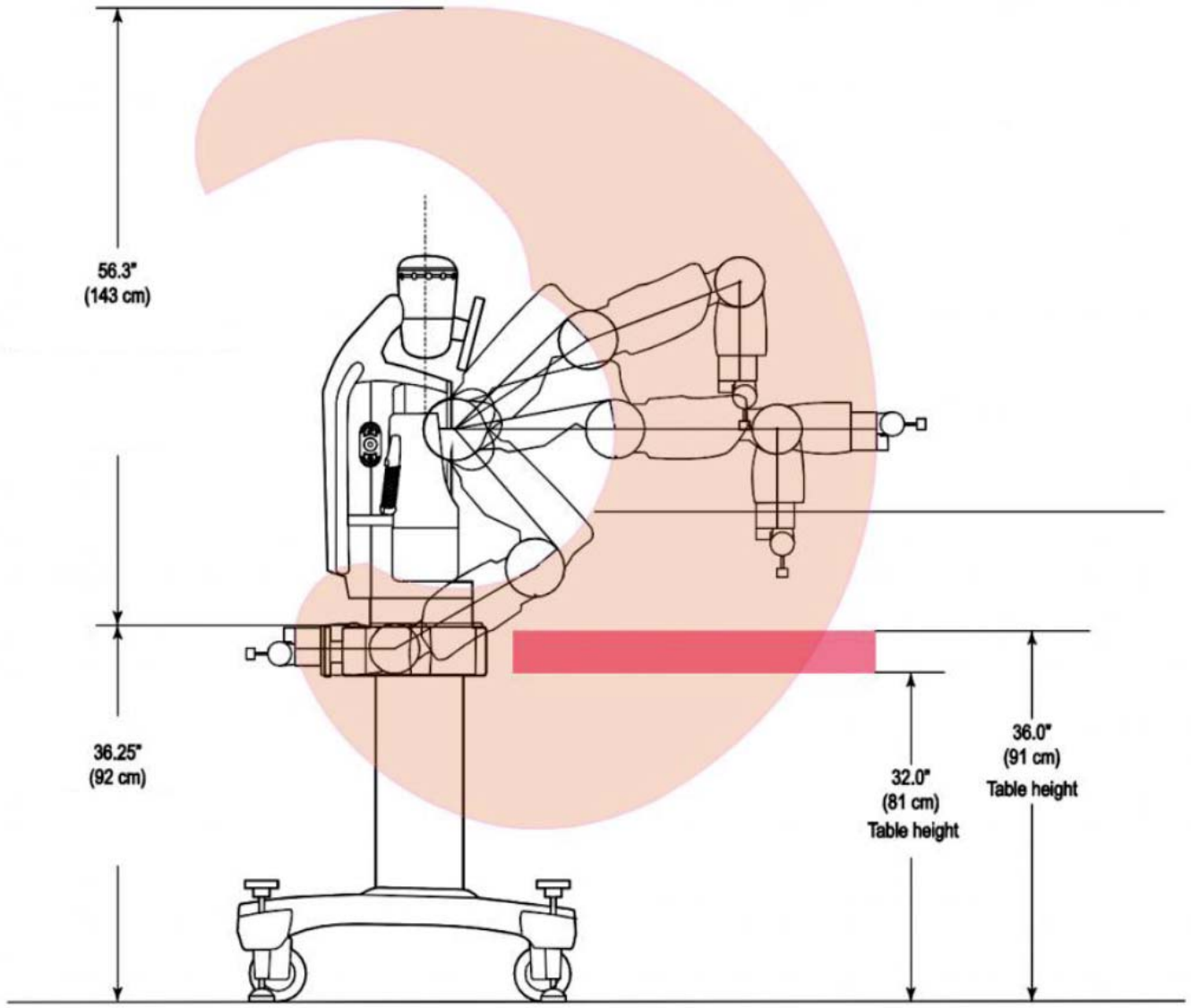


Figure 8a. Baxter Robot Arms Workspace, Side View

(Rethink Robotics, 2016)

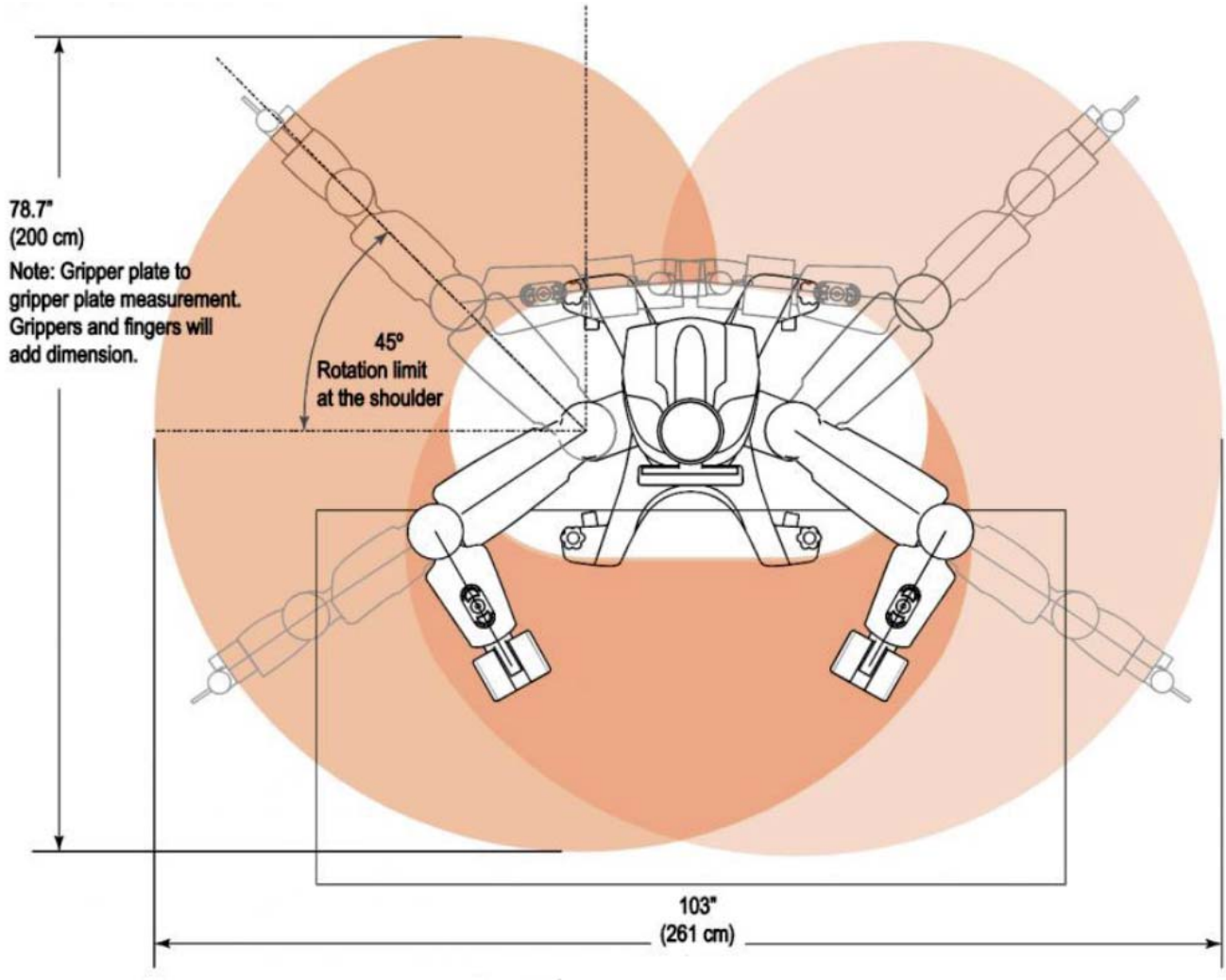


Figure 8b. Baxter Robot Arms Workspace, Top View

(Rethink Robotics, 2016)

The figure below shows the 3D reachable workspace for the Baxter Left/Right Robot Arms, with length units of m. This was generated by nested FPK numerical iteration in MATLAB, over all joint angle limits (excepting θ_7, i.e. joint W_2, whose wrist roll does not affect the *XYZ* coordinates of the end-effectors). This figure was drawn intentionally sparse in order to see the Baxter Left/Right Arms shown at all zero angles. We see that the workspace corresponds to that given by Rethink Robotics in the figures above. There is significant overlap between the Left- and Right-Arm reachable workspaces. The following figure does not show the internal workspace voids due to the link lengths and the joint limits, since the external points occlude them – but they are similar to those pictured above. The ensuing figure shows the Baxter left-arm workspace in the *XZ* plane of {BL}. We see that the Rethink Robotics figure is essentially correct, but not to scale (the MATLAB figure is to scale).

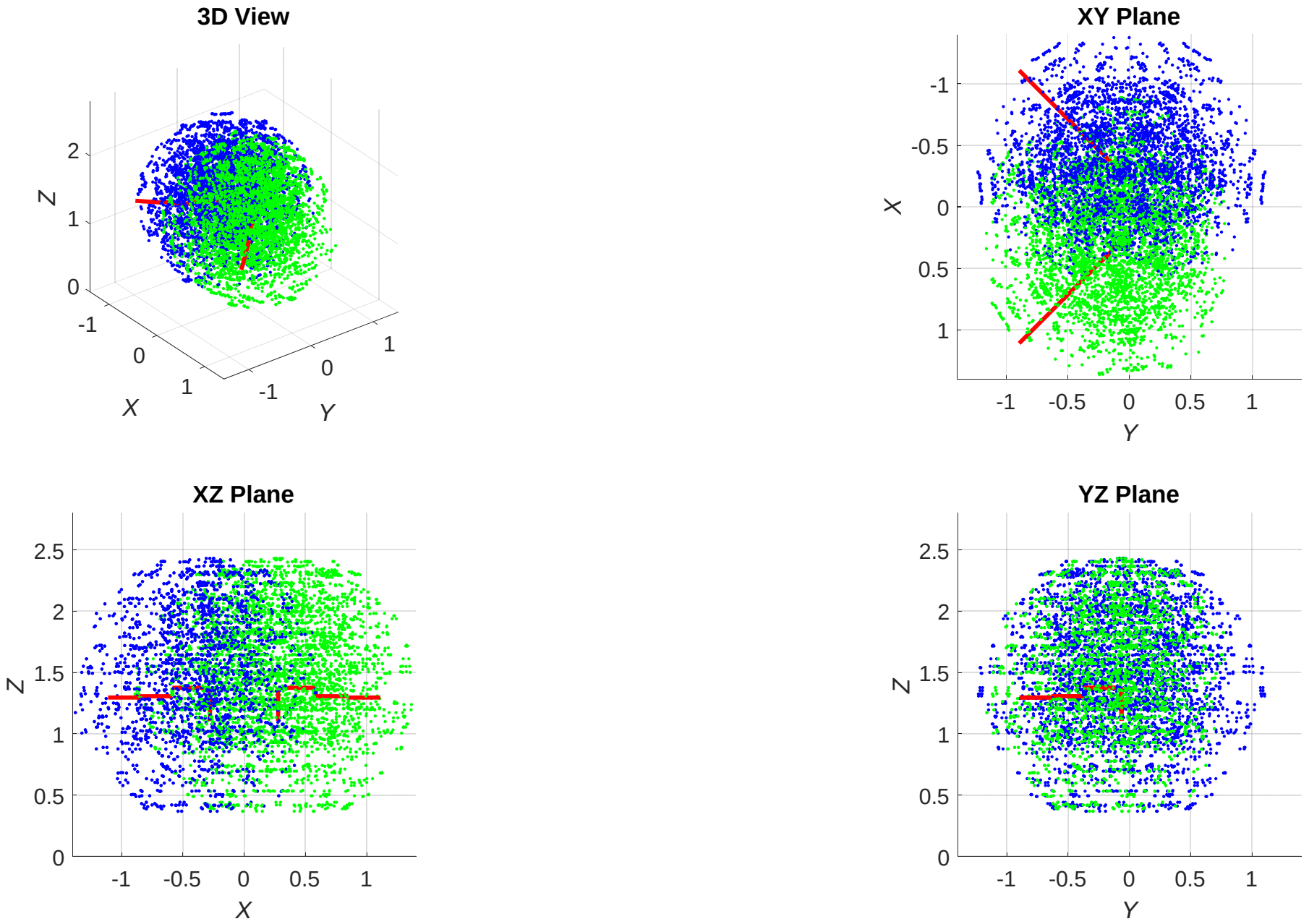


Figure 9. Baxter Left/Right Robot Arms Workspace, Numerically Generated by MATLAB (m units)

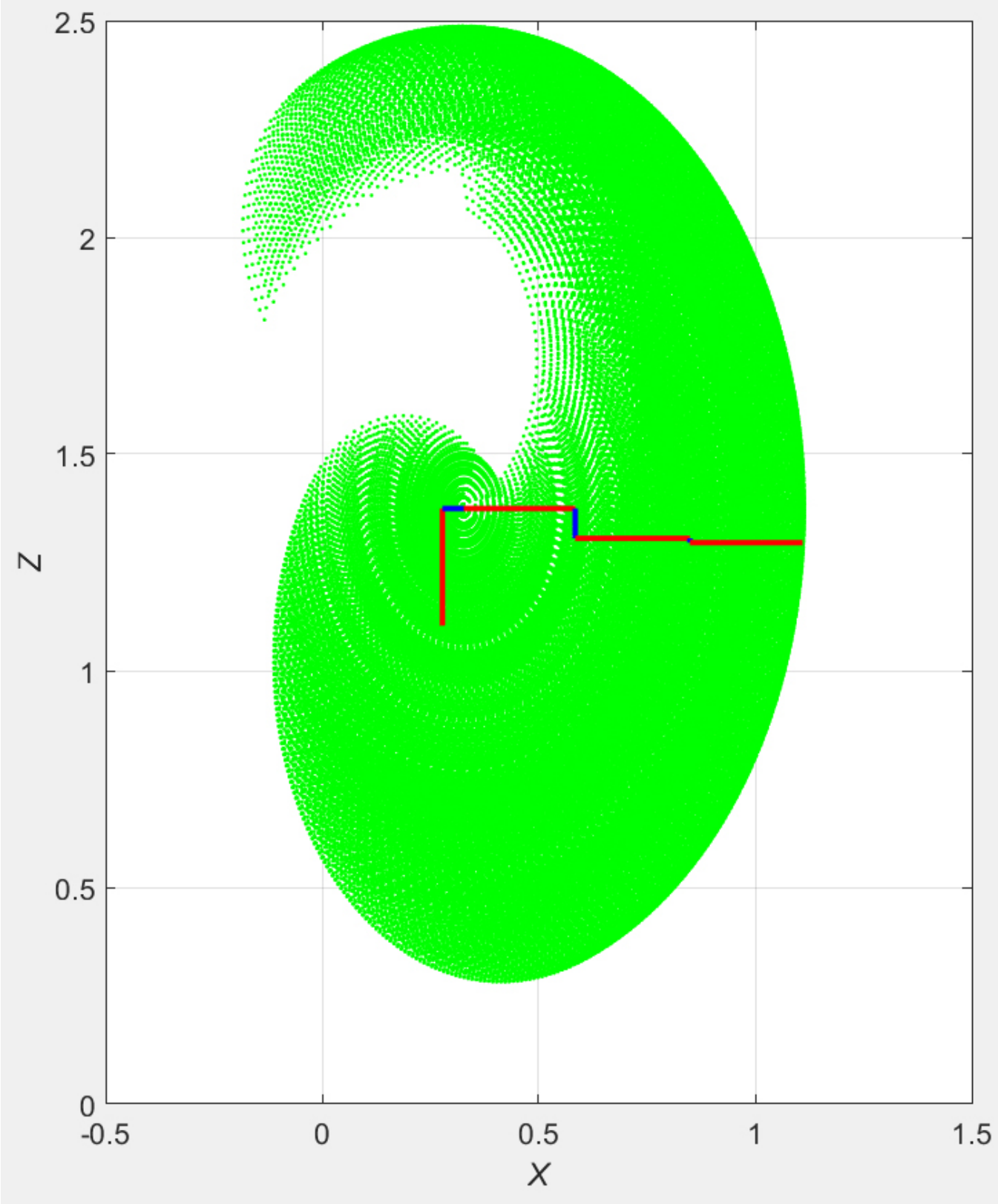


Figure 10a. Baxter Left-Arm {*BL*} *XZ* Plane Workspace

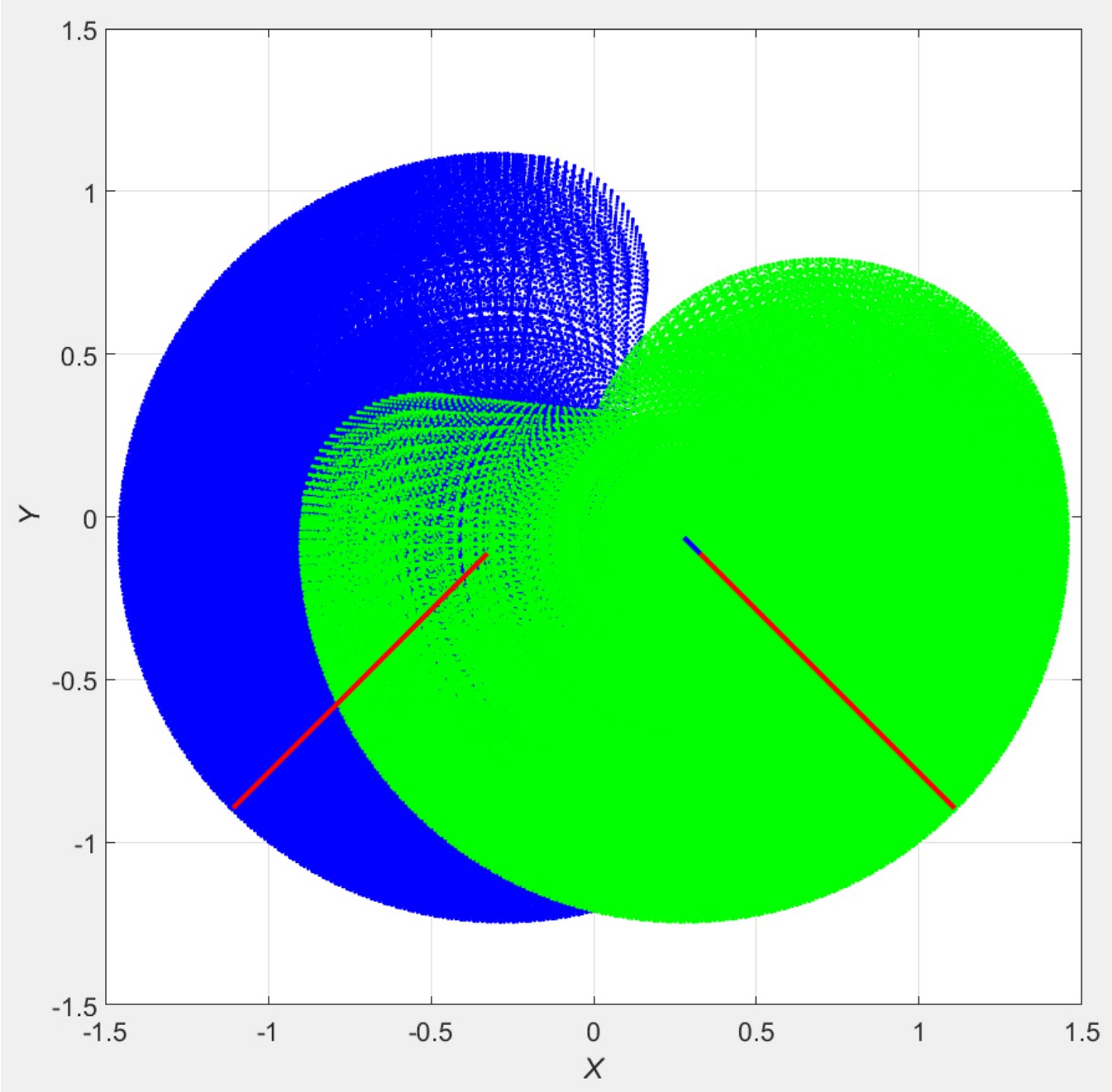


Figure 10b. Baxter Left-Arm {0} *XY* Plane Workspace

6. Baxter Inverse Pose Kinematics

In general, the Inverse Pose Kinematics (IPK) problem for a serial-chain robot is stated: Given the pose (position and orientation) of the end frame of interest, calculate the joint values to obtain that pose. For serial-chain robots, the IPK solution starts with the FPK equations. The solution of coupled nonlinear algebraic equations is required and multiple solution sets generally result.

The kinematically-redundant 7-dof Baxter arms provide a big IPK challenge compared to that of a similar 6-dof robot arm. Also, for the Baxter Robot arm, the two link offsets L_3 and L_5 provide significant complications to the IPK problem. According to Pieper's solution (Pieper, 1968), if three consecutive coordinate frames of your serial robot arm intersect in one point, an analytical solution is guaranteed to exist. Unfortunately the offset $L_5 \neq 0$ prevents a spherical wrist (with three coincident coordinate frame origins); further, the Baxter robot arm does not have any more than two consecutive coordinate frames that share a common origin (only frames {0},{1} and {6},{7}; see Figure 3). There is **no** analytical IPK solution for the general 7-dof Baxter robot arm with non-zero offsets (however, note Pieper's criterion only speaks to the existence of an analytical solution, not the absence of an analytical IPK solution).

Therefore, to start this chapter, we will simplify the Baxter arm as follows, to derive an analytical solution to the IPK problem: let us lock the third joint angle to $\theta_3 = 0$ and also assume $L_5 = 0$. The former limitation means this first solution will not exploit the kinematic redundancy of the Baxter robot arms. The latter assumption will lead to some error, but the size of $L_5 = 0$ is very small relative to all other Baxter robot arm lengths, as seen in Table 5. The kinematic diagram and FPK solution was already presented for the $\theta_3 = 0$ case in Figure 7 and ensuing derivations.

6.1 Analytical IPK Solution for the 6-dof Baxter Left Arm

The statement of the IPK problem for the reduced 6-dof Baxter serial robot left arm with $\theta_3 = 0$ and $L_5 = 0$ is given below. Remember from the 6-dof FPK solution presented earlier, since frame {3} is not included, the frame numbering is such that 7-dof frames i = 4,5,6,7 are now numbered i = 3,4,5,6. Also remember that $L_h = \sqrt{L_2^2 + L_3^2}$.

Given: the constant DH Parameters

and the required end-effector pose $\left[{}^{0}_{6}T\right] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & {}^{0}x_6 \\ r_{21} & r_{22} & r_{23} & {}^{0}y_6 \\ r_{31} & r_{32} & r_{33} & {}^{0}z_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Calculate: the joint angles $(\theta_1, \theta_2, \theta_4, \theta_5, \theta_6, \theta_7)$ to achieve this pose

Actually, in the real world, a more general pose input $\left[{}^{Wo}_{GL}T\right]$ must be given. Then the associated required IPK input $\left[{}^{0}_{6}T\right]$ is calculated from known constant homogeneous transformation matrices as follows:

$$\left[{}^{Wo}_{GL}T\right]=\left[{}^{Wo}_{B}T\right]\left[{}^{B}_{0}T\right]\left[{}^{0}_{6}T\right]\left[{}^{6}_{GL}T\right]$$

$$\left[{}^{0}_{6}T\right]=\left[{}^{B}_{0}T\right]^{-1}\left[{}^{Wo}_{B}T\right]^{-1}\left[{}^{Wo}_{GL}T\right]\left[{}^{6}_{GL}T\right]^{-1}$$

$$\left[{}^{0}_{6}T\right]=\left[{}^{Wo}_{0}T\right]^{-1}\left[{}^{Wo}_{GL}T\right]\left[{}^{6}_{GL}T\right]^{-1}$$

With $L_5 = 0$, the 3-dof Baxter wrist is spherical (three coordinate frames sharing a common origin) and hence an analytical solution is guaranteed to exist by Pieper's Criterion. Assuming $L_5 = 0$, the rotation matrix terms for $\left[{}^{0}_{6}R\right]$ are identical to those given earlier in the FPK solution for this case. The translational terms for this result are further simplified by using $L_5 = 0$:

$$\left\{{}^{0}P_4\right\}=\begin{Bmatrix}{}^{0}x_4\\ {}^{0}y_4\\ {}^{0}z_4\end{Bmatrix}=\begin{Bmatrix}c_1(L_1+L_hc_2+L_4c_{24})\\ s_1(L_1+L_hc_2+L_4c_{24})\\ -L_hs_2-L_4s_{24}\end{Bmatrix}$$

Note that, with $\theta_3 = 0$ and $L_5 = 0$, the translational vector $\left\{{}^{0}P_6\right\}=\left\{{}^{0}P_4\right\}$ is a function of $(\theta_1,\theta_2,\theta_4)$ only. Therefore our solution approach will be to solve $(\theta_1,\theta_2,\theta_4)$ first given values for $\left\{{}^{0}P_6\right\}$, and then solve for $(\theta_5,\theta_6,\theta_7)$ second given values for $\left[{}^{0}_{6}R\right]$. Remember with joint angle θ_3 locked to zero, the numbering has shifted such that *i* = 4, 5, 6 corresponds to frames {5}, {6}, {7}.

Translational joints $(\theta_1,\theta_2,\theta_4)$ solution.

Joint angle θ_1 is found from a ratio of the *Y* to *X* translational equations:

$$\frac{s_1(L_1+L_hc_2+L_4c_{24})}{c_1(L_1+L_hc_2+L_4c_{24})}=\frac{{}^{0}y_4}{{}^{0}x_4}$$

$$\frac{s_1}{c_1}=\tan\theta_1=\frac{{}^{0}y_4}{{}^{0}x_4}$$

$$\theta_1=\text{atan2}({}^{0}y_4,{}^{0}x_4)$$

where **atan2** is the quadrant-specific inverse tangent function. Now that θ_1 is known, isolate the $(\theta_2+\theta_4)$ terms in the *X* and *Z* equations as follows:

$$L_4 c_{24} = \frac{x}{c_1} - L_1 - L_h c_2$$
$$L_4 s_{24} = -z - L_h s_2$$

where for notational convenience we have shortened names of the constants $x = {}^0x_4$ and $z = {}^0z_4$. Square and add these two equations to eliminate the $(\theta_2 + \theta_4)$ terms by using $\cos^2(\theta_2+\theta_4)+\sin^2(\theta_2+\theta_4)=1$, obtaining one equation in one unknown θ_2:

$$E\cos\theta_2 + F\sin\theta_2 + G = 0$$

where:

$$E = 2L_h(L_1 - \frac{x}{c_1})$$
$$F = 2L_h z$$
$$G = \frac{x^2}{c_1^2} + L_1^2 + L_h^2 - L_4^2 + z^2 - 2\frac{L_1 x}{c_1}$$

We can solve this familiar equation for the unknown θ_2 by using the **Tangent Half-Angle Substitution**:

If we define $t = \tan\left(\frac{\theta_2}{2}\right)$ then $\cos\theta_2 = \frac{1-t^2}{1+t^2}$ and $\sin\theta_2 = \frac{2t}{1+t^2}$

Substitute the **Tangent Half-Angle Substitution** into the *EFG* equation:

$$E\left(\frac{1-t^2}{1+t^2}\right) + F\left(\frac{2t}{1+t^2}\right) + G = 0$$

$$E(1-t^2) + F(2t) + G(1+t^2) = 0$$

$$(G-E)t^2 + (2F)t + (G+E) = 0$$

Using the quadratic formula, we can solve for the intermediate parameter *t*:

$$t_{1,2} = \frac{-F \pm \sqrt{E^2 + F^2 - G^2}}{G - E}$$

Then solve for θ_2 by inverting the original Tangent Half-Angle Substitution definition:

$$\theta_{2_{1,2}} = 2\tan^{-1}(t_{1,2})$$

Note that we do not need to use the quadrant-specific **atan2** function in the above solution, since the multiplier 2 takes care of possible the trigonometric uncertainty (dual values) of inverse trigonometric functions. There are two solutions for θ_2, from the $\pm$ in the quadratic formula.

To solve for the third translational unknown θ_4, return to the two equations that were squared and added; use a ratio of the *Z* to the *X* equations:

$$\theta_{4_{1,2}} = \text{atan2}(-z - L_h s_{2_{1,2}}, \frac{x}{c_1} - L_1 - L_h c_{2_{1,2}}) - \theta_{2_{1,2}}$$

Overall, there are two possible solutions for the translational joints as shown in the table below.

θ_1	θ_{2_1}	θ_{4_1}
θ_1	θ_{2_2}	θ_{4_2}

These two solutions are provided by a unique θ_1 and (θ_2, θ_4) pairs corresponding to Elbow Up and Elbow Down solution branches.

Wrist joints $(\theta_5, \theta_6, \theta_7)$ solution.

With the translational joint angles $(\theta_1, \theta_2, \theta_4)$ now known, we can use the rotation matrix expressions to solve for the unknown wrist joint angles $(\theta_5, \theta_6, \theta_7)$. The rotation matrix partitions as follows, where the r_{ij} terms are given from the required pose (i.e. $\left[{}^0_6R\right]$ is known from the IPK input). Remember, *i* = 3,4,5,6 correspond to frames 4,5,6,7.

$$\left[{}^0_6R\right] = \left[{}^0_3R(\theta_1, \theta_2, \theta_4)\right]\left[{}^3_6R(\theta_5, \theta_6, \theta_7)\right] = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

Since $(\theta_1, \theta_2, \theta_4)$ are now known, we can calculate the required numbers R_{ij} for $\left[{}^3_6R\right]$ of the given pose, where the expressions for $\left[{}^0_3R(\theta_1, \theta_2, \theta_4)\right]$ were given in the 6-dof FPK solution.

$$\left[{}^3_6R(\theta_5, \theta_6, \theta_7)\right] = \left[{}^0_3R(\theta_1, \theta_2, \theta_4)\right]^{-1}\left[{}^0_6R\right] = \left[{}^0_3R(\theta_1, \theta_2, \theta_4)\right]^T\left[{}^0_6R\right] = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}$$

The equations to solve come from the FPK expressions:

$$\left[T_{\text{shoulder/elbow}}\right] = \left[{}^{0}_{3}T(\theta_1,\theta_2,\theta_4)\right] = \begin{bmatrix} -c_1 s_{24} & -c_1 c_{24} & -s_1 & c_1(L_1 + L_h c_2) \\ -s_1 s_{24} & -s_1 c_{24} & c_1 & s_1(L_1 + L_h c_2) \\ -c_{24} & s_{24} & 0 & -L_h s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\left[{}^{3}_{6}R(\theta_5,\theta_6,\theta_7)\right] = \begin{bmatrix} -s_5 s_7 + c_5 c_6 c_7 & -s_5 c_7 - c_5 c_6 s_7 & c_5 s_6 \\ s_6 c_7 & -s_6 s_7 & -c_6 \\ c_5 s_7 + s_5 c_6 c_7 & c_5 c_7 - s_5 c_6 s_7 & s_5 s_6 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}$$

Due to the six constraints on an orthonormal rotation matrix (three perpendicular and three unit vector constraints), only three of these nine equations are independent. We will use three combinations of the equations with the simplest terms in order to solve for $(\theta_5,\theta_6,\theta_7)$.

From a ratio of the (3,3) to (1,3) equations, we can solve for wrist angle θ_5.

$$\frac{s_5 s_6}{c_5 s_6} = \frac{R_{33}}{R_{13}}$$

$$\theta_5 = \text{atan2}(R_{33}, R_{13})$$

From a ratio of the (2,2) to (2,1) equations, we can solve for wrist angle θ_7.

$$\frac{-s_6 s_7}{s_6 c_7} = \frac{R_{22}}{R_{21}}$$

$$\theta_7 = \text{atan2}(-R_{22}, R_{21})$$

And finally, knowing θ_7 and using a ratio of the (2,1) to (2,3) equations, we can solve for wrist angle θ_6.

$$\frac{s_6 c_7}{-c_6} = \frac{R_{21}}{R_{23}}$$

$$\theta_6 = \text{atan2}\left(\frac{R_{21}}{c_7}, -R_{23}\right)$$

6.2 Numerical IPK Solution for the General 7-dof Baxter Left Arm

The statement of the IPK problem for the general 7-dof Baxter serial robot left arm with all seven joints active and all kinematics lengths and offsets included is given below.

Given: the constant DH Parameters

and the required end-effector pose $\left[{}^{0}_{7}T\right] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & {}^{0}x_7 \\ r_{21} & r_{22} & r_{23} & {}^{0}y_7 \\ r_{31} & r_{32} & r_{33} & {}^{0}z_7 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Calculate: the joint angles $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7)$ to achieve this pose

We will implement the numerical iterative Newton-Raphson method to solve this IPK problem. This requires a good initial guess to the solution and will only yield one solution set. If the initial guess is 'sufficiently close' to an answer, quadratic convergence is guaranteed and generally the solution set closest to the initial guess will be found.

The Newton-Raphson method requires the definition of six scalar functions for the solution process. These are the three translational functions and three of the nine rotational matrix terms from the general 7-dof FPK expressions (presented earlier).

The first three of these nonlinear, coupled functions are the three translational functions from FPK; the functions must all be rearranged so their right-hand-side is zero. Again, since the origins of frames {6} and {7} are coincident at the wrist point, the translational terms are only functions of the first five joint angles:

$$\left\{ {}^{0}P_7 \right\} = \left\{ {}^{0}P_7(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) \right\} = \begin{Bmatrix} {}^{0}x_7 \\ {}^{0}y_7 \\ {}^{0}z_7 \end{Bmatrix}$$

The first three functions for the Newton-Raphson IPK solution method are:

$$\begin{aligned} f_1(\Theta) = {} & L_1c_1 + L_2c_1c_2 - L_3(s_1s_3 + c_1s_2c_3) - L_4((s_1s_3 + c_1s_2c_3)s_4 - c_1c_2c_4) \\ & -L_5((s_1c_3 - c_1s_2s_3)s_5 + ((s_1s_3 + c_1s_2c_3)c_4 + c_1c_2s_4)c_5) - {}^{0}x_7 \end{aligned}$$

$$\begin{aligned} f_2(\Theta) = {} & L_1s_1 + L_2s_1c_2 + L_3(c_1s_3 - s_1s_2c_3) + L_4((c_1s_3 - s_1s_2c_3)s_4 + s_1c_2c_4) \\ & +L_5((c_1c_3 + s_1s_2s_3)s_5 + ((c_1s_3 - s_1s_2c_3)c_4 - s_1c_2s_4)c_5) - {}^{0}y_7 \end{aligned}$$

$$f_3(\Theta) = -L_2s_2 - L_3c_2c_3 - L_4(s_2c_4 + c_2c_3s_4) + L_5((s_2s_4 - c_2c_3c_4)c_5 + c_2s_3s_5) - {}^{0}z_7$$

The second three functions for the Newton-Raphson IPK solution method must be chosen from the rotation matrix terms. They must be independent (not all three from one row or column) and we will choose those elements with the simplest analytical terms: the (1,3), (2,3), and (3,2) terms.

$$f_4(\Theta) = -((s_1 s_3 + c_1 s_2 c_3)s_4 - c_1 c_2 c_4)c_6 - ((s_1 c_3 - c_1 s_2 s_3)s_5 + ((s_1 s_3 + c_1 s_2 c_3)c_4 + c_1 c_2 s_4)c_5)s_6 - r_{13}$$

$$f_5(\Theta) = ((c_1 s_3 - s_1 s_2 c_3)s_4 + s_1 c_2 c_4)c_6 + ((c_1 c_3 + s_1 s_2 s_3)s_5 + ((c_1 s_3 - s_1 s_2 c_3)c_4 - s_1 c_2 s_4)c_5)s_6 - r_{23}$$

$$f_6(\Theta) = -((s_2 s_4 - c_2 c_3 c_4)s_5 - c_2 s_3 c_5)c_7 - ((s_2 c_4 + c_2 c_3 s_4)s_6 + ((s_2 s_4 - c_2 c_3 c_4)c_5 + c_2 s_3 s_5)c_6)s_7 - r_{32}$$

The vectors for Newton-Raphson method implementation are:

$$\{F(\Theta)\} = \begin{Bmatrix} f_1(\Theta) \\ f_2(\Theta) \\ f_3(\Theta) \\ f_4(\Theta) \\ f_5(\Theta) \\ f_6(\Theta) \end{Bmatrix} \qquad \{\Theta\} = \begin{Bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \\ \theta_7 \end{Bmatrix}$$

The Newton-Raphson Method involves numerical iteration to solve coupled sets of *m* nonlinear equations (algebraic transcendental) in *n* unknowns. It requires a good initial guess of the solution to get started and it only yields one of the possible multiple solutions. The Newton-Raphson method is an extension of Newton's single function/single variable root-finding technique to *m* functions and *n* variables. The following is the form of the given functions to solve.

$$\{F(\Theta)\} = \{\mathbf{0}\}$$

Perform a Taylor Series Expansion of $\{F\}$ about $\{\Theta\}$:

$$f_i\left(\{\Theta\} + \{\delta\Theta\}\right) = f_i\left(\{\Theta\}\right) + \sum_{j=1}^{n} \frac{\partial f_i}{\partial \theta_j}\delta\theta_j + O\left(\{\delta\Theta^2\}\right) \qquad i = 1, 2, \cdots, m$$

Introduce $\left[J_{NR}\right] = \left[J_{NR}(\Theta)\right] = \left[\frac{\partial f_i}{\partial \theta_j}\right]$ as the Newton-Raphson Jacobian Matrix, a multi-dimensional form of the derivative and a function of $\{\Theta\}$. If $\{\delta\Theta\}$ is small, the higher-order terms $O\left(\{\delta\Theta^2\}\right)$ from the Taylor Series Expansion are negligible. For solution, we require:

$$f_i\left(\{\Theta\} + \{\delta\Theta\}\right) = 0 \qquad i = 1, 2, \cdots, m$$

Now with $O\left(\left\{\delta\Theta^2\right\}\right)\to 0$ we have:

$$f_i\left(\{\Theta\}+\{\delta\Theta\}\right)=f_i\left(\{\Theta\}\right)+\sum_{j=1}^{n}\frac{\partial f_i}{\partial\theta_j}\delta\theta_j \qquad i=1,2,\cdots,m$$
$$=f_i\left(\{\Theta\}\right)+\left[J_{NR}\right]\{\delta\Theta\}=\{0\}$$

So to calculate the required correction factor $\delta\Theta$ at each iterative solution step, we must solve $\left\{F\left(\{\Theta\}\right)\right\}+\left[J_{NR}\right]\{\delta\Theta\}=\{0\}$. However, in the case of the 7-dof kinematically-redundant Baxter robot arm, the Newton-Raphson Jacobian matrix is not square, but 6 x 7, so these equations are underconstrained, the good case since there are infinite solutions to the $\delta\Theta$ vector at each iteration step. A possible solution is:

$$\{\delta\Theta\}=-\left[J_{NR}\right]^{*}\left\{F\left(\{\Theta\}\right)\right\}$$

where we use the right Moore-Penrose pseudoinverse of the Newton-Raphson Jacobian matrix:

$$\left[J_{NR}\right]^{*}=\left[J_{NR}\right]^{T}\left(\left[J_{NR}\right]\left[J_{NR}\right]^{T}\right)^{-1}$$

Note that $\left[J_{NR}\right]^{*}$ is of size 7 x 6. One very cool aspect of this formula is that, no matter how many joints your kinematically-redundant robot has beyond 7, only a 6 x 6 (*m* x *m*) inversion is ever required.

In this numerical IPK iterative solution approach, no attempt is made to optimize anything, i.e. we are not exploiting the kinematic redundancy of this robot (see the Velocity chapter to remedy this limitation). Instead, the Moore-Penrose Pseudoinverse ensures that the vector Euclidean-norm of $\delta\Theta$ is the minimum possible at each step (not necessarily a great idea).

Newton-Raphson Method Algorithm Summary

0) Establish the functions and variables to solve for: $\left\{F\left(\{\Theta\}\right)\right\}=\{\mathbf{0}\}$

1) Make an initial guess to the solution: $\{\Theta_0\}$

2) Solve $\left[J_{NR}\right]\left(\{\Theta_k\}\right)\{\delta\Theta_k\}=-\left\{F\left(\{\Theta_k\}\right)\right\}$ for $\{\delta\mathbf{X}_k\}$, where *k* is the iteration counter.

3) Update the current best guess for the solution: $\{\Theta_{k+1}\}=\{\Theta_k\}+\{\delta\Theta_k\}$

4) Iterate until $\left\|\{\delta\Theta_k\}\right\|<\varepsilon$, where we use the Euclidean norm and ε is a small, user-defined scalar solution tolerance. Also, halt the iteration if the number of steps becomes too high (which means the solution is diverging).

If the initial guess to the solution $\{\Theta_0\}$ is sufficiently close to an actual solution, the Newton-Raphson technique guarantees quadratic convergence.

Now, for Baxter Robot inverse pose kinematics problems, the Newton-Raphson technique requires a good initial guess to ensure convergence and yields only one of the possible multiple solutions. However, this does not present any difficulty since the existing known pose configuration makes an excellent initial guess for the next solution step (if the control rate is high, many cycles per second, the robot cannot move too far from this known initial guess in one step). Also, except in the case of singularities where the multiple solution branches converge, the one resulting solution is generally the one you want, closest to the initial guess, most likely the preferred configuration of the real robot.

There is a very interesting and ***beautiful*** relationship between numerical inverse pose solution and the velocity problem for serial robots (presented next). The Newton-Raphson Jacobian Matrix is nearly identical to the serial arm Velocity Jacobian Matrix. This reduces computation if you need both inverse pose computation and inverse-velocity-based resolved-rate control.

7. Baxter Velocity Kinematics and Resolved-Rate Control

So above we have seen that the Inverse Pose Kinematics (IPK) problem for the 7-dof Baxter robot arm is a huge problem. Due to the arm offsets there is no closed-form analytical solution to the IPK problem. Therefore, a numerical procedure is required, with no set number of iterations; this is a challenge for real-time control.

An attractive alternative for control of kinematically-redundant serial robot arms is the resolved-rate control method, based on the inverse velocity solution. The inverse velocity solution uses a linear set of equations which can easily be solved in a control loop at real-time rates. This section presents the Baxter velocity kinematics, including the Jacobian matrix, followed by the resolved-rate control method for the kinematically-redundant Baxter robot arm.

A kinematically-redundant robot arm is one whose joint-space degrees of freedom, *n*, is greater than the Cartesian-space degrees-of-freedom, *m*, it is required to operate in. The $n = 7$ dof Baxter robot arm is kinematically-redundant for operating in the $m = 6$ 3D Cartesian space (three translations and three rotations), since $n > m$.

7.1 Baxter Robot Jacobian Matrix

The Jacobian matrix [*J*] is a linear transformation mapping joint rates $\left\{\dot{\Theta}\right\}$ to Cartesian velocities $\left\{\dot{X}\right\}$:

$$\left\{{}^{k}\dot{X}\right\} = \left[{}^{k}J(\Theta)\right]\left\{\dot{\Theta}\right\}$$

$$m \times 1 \equiv (m \times n)(n \times 1)$$

Where *m* is the dimension of the Cartesian (task) space, *n* is the dimension of the joint space, and we can express the resulting Cartesian velocities in any frame {*k*}; $\left\{\dot{\Theta}\right\}$ are the relative joint angle rates and hence are expressed about the *n* different local *Z* axes. The Jacobian matrix is a function of the *n* joint angles Θ, in general; therefore, it must be calculated anew with each motion.

The Jacobian matrix is a multi-dimensional form of the derivative:

$$\left[J\right] = \left[\frac{\partial f_i}{\partial \theta_j}\right]$$

where f_i are the six pose functions, and θ_j are the seven joint angles. A thousand and one references state this about the Jacobian matrix, but it is only half-true. It works well for translational terms, where:

$$\begin{Bmatrix} f_1 \\ f_2 \\ f_3 \end{Bmatrix} = \begin{Bmatrix} x(\Theta) \\ y(\Theta) \\ z(\Theta) \end{Bmatrix}$$

but there are no possible functions with respect to which we can take the partial derivatives to obtain the rotational terms of the Jacobian matrix. The rotational terms may be found using a relative angular velocity equation.

For the n = 7-joint Baxter robot arm operating in the standard m = 6-dimensional Cartesian space:

$$\left\{\dot{\Theta}\right\} = \begin{Bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \\ \dot{\theta}_6 \\ \dot{\theta}_7 \end{Bmatrix} \qquad \left\{{}^k\dot{X}\right\} = \begin{Bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{Bmatrix}$$

There are at least four methods to derive the Jacobian matrix for a serial-chain robot; my favorite is now presented.

Physical interpretation of the Jacobian matrix

The i^{th} Jacobian matrix column is the end-effector translational and rotational velocity due to joint i, with the joint rate factored out. Then by linear superposition, the overall end-effector Cartesian velocity is the sum of all n columns (each multiplied by the respective joint rate). Each Jacobian matrix column i is the absolute Cartesian velocity vector of the last active joint frame $\{N\}$ with respect to the base frame, due to joint i only, and with the variable joint rate $\dot{\theta}_i$ factored out.

$${}^k\left[J\right] = {}^k\begin{bmatrix} | & | & \cdots & | \\ \left\{{}^0\dot{X}_N\right\}_1 & \left\{{}^0\dot{X}_N\right\}_2 & \cdots & \left\{{}^0\dot{X}_N\right\}_N \\ | & | & \cdots & | \end{bmatrix}$$

$${}^k\left\{{}^0\dot{X}_N\right\}_i = \begin{Bmatrix} {}^k\left\{{}^0V_N\right\}_i \\ {}^k\left\{{}^0\omega_N\right\}_i \end{Bmatrix}$$

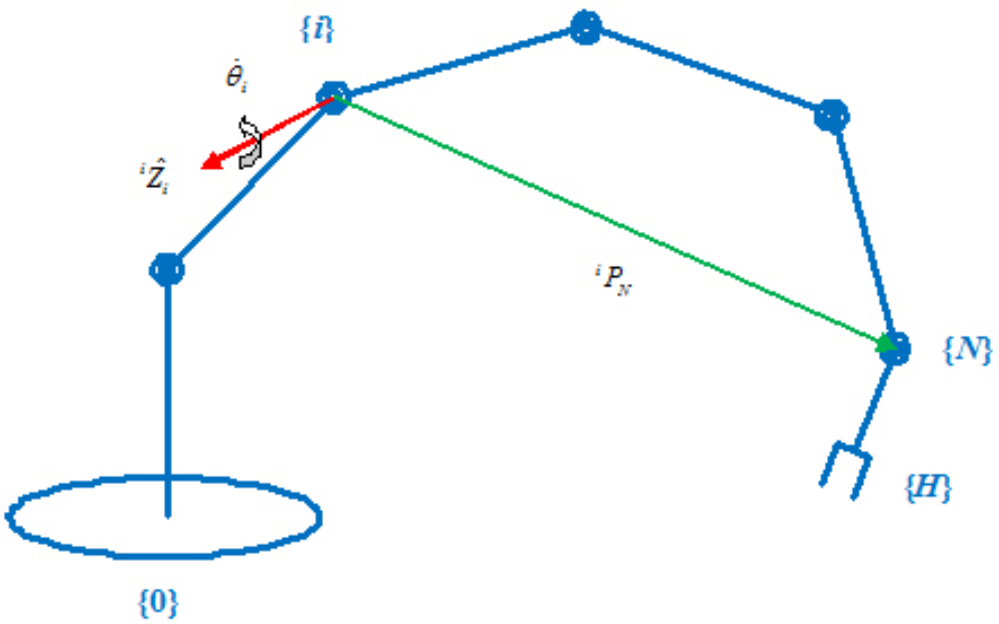


Figure 11. Jacobian Matrix Column *i* Derivation Image, Revolute Joints

Here is Jacobian matrix column *i*, for a revolute joint:

$$ {}^k\{J\}_i = \begin{Bmatrix} \{{}^k\hat{Z}_i\} \times {}^k\{{}^iP_N\} \\ \{{}^k\hat{Z}_i\} \end{Bmatrix} = \begin{Bmatrix} [{}^k_iR]\{{}^i\hat{Z}_i \times {}^iP_N\} \\ [{}^k_iR]\{{}^i\hat{Z}_i\} \end{Bmatrix} $$

where:

$$ \{{}^k\hat{Z}_i\} = [{}^k_iR]\{{}^i\hat{Z}_i\} $$

is the third column of orthonormal rotation matrix $[{}^k_iR]$ and:

$$ {}^k\{{}^iP_N\} = [{}^k_iR]^i\{{}^iP_N\} $$

where ${}^i\{{}^iP_N\}$ is the translational part of homogeneous transformation matrix $[{}^i_NT]$ (the fourth column, rows 1 through 3). Here is the Jacobian matrix for an all-revolute-joint manipulator:

$$ {}^k[J] = {}^k\begin{bmatrix} \begin{Bmatrix} \{{}^k\hat{Z}_1\} \times {}^k\{{}^1P_N\} \\ \{{}^k\hat{Z}_1\} \end{Bmatrix} & \cdots & \begin{Bmatrix} \{{}^k\hat{Z}_i\} \times {}^k\{{}^iP_N\} \\ \{{}^k\hat{Z}_i\} \end{Bmatrix} & \cdots & \begin{Bmatrix} \{{}^k\hat{Z}_N\} \times {}^k\{{}^NP_N\} \\ \{{}^k\hat{Z}_N\} \end{Bmatrix} \end{bmatrix} $$

For the 7-dof Baxter arm, $N = 7$, $i = 1,2,3,4,5,6,7$, and the Jacobian and Cartesian velocity frame of expression (basis) is any convenient coordinate frame *k*. Often *k* is chosen to be {0}.

Jacobian Matrix Expressed in Another Frame

Here the Jacobian matrix still relates the end-effector frame velocity with respect to the base frame. But simpler analytical expressions are possible for the Jacobian matrix, by choosing an intermediate frame to express the coordinates of the velocity vectors (different basis of expression). For the Baxter robot arm, intermediate frame k = {4} will yield the simplest analytical expressions for the Jacobian matrix, for analytical singularity analysis and other purposes.

$$ {}^{k}\begin{Bmatrix}\{V\}\\ \{\omega\}\end{Bmatrix} = \begin{bmatrix}\left[{}^{k}_{0}R\right] & [0]\\ [0] & \left[{}^{k}_{0}R\right]\end{bmatrix} {}^{0}\begin{Bmatrix}\{V\}\\ \{\omega\}\end{Bmatrix} $$

$$ {}^{k}\begin{Bmatrix}\{V\}\\ \{\omega\}\end{Bmatrix} = \left[{}^{k}J\right]\left\{\dot{\Theta}\right\} $$

$$ {}^{0}\begin{Bmatrix}\{V\}\\ \{\omega\}\end{Bmatrix} = \left[{}^{0}J\right]\left\{\dot{\Theta}\right\} $$

Remember, $\left\{\dot{\Theta}\right\}$ is not dependent on a frame, since all seven joint rates are relative to their previous moving link.

$$ \left[{}^{k}J\right]\left\{\dot{\Theta}\right\} = \begin{bmatrix}\left[{}^{k}_{0}R\right] & [0]\\ [0] & \left[{}^{k}_{0}R\right]\end{bmatrix}\left[{}^{0}J\right]\left\{\dot{\Theta}\right\} $$

$$ \left[{}^{k}J\right] = \begin{bmatrix}\left[{}^{k}_{0}R\right] & [0]\\ [0] & \left[{}^{k}_{0}R\right]\end{bmatrix}\left[{}^{0}J\right] $$

For numerical implementation, this transformation is not as important as for analytical analysis.

Crucial - Units

The translational rows of the Jacobian matrix have length units (m). The rotational rows of the Jacobian matrix are unitless. Therefore, since Cartesian velocity units of $\left\{\dot{X}\right\}$ are m/sec and rad/sec, for translational and rotational terms, respectively, in the overall velocity equation $\left\{\dot{X}\right\} = \left[J\right]\left\{\dot{\Theta}\right\}$, one MUST use units of rad/sec for $\left\{\dot{\Theta}\right\}$, not deg/sec!

Further, the units and size mismatch between translational and rotational rows of the Jacobian matrix can cause numerical troubles, which is well-known for serial robots. Using m rather than mm for length units will help this problem for the Baxter robots arms.

Cartesian Transformation of Translational and Rotational Velocities

The Jacobian matrix presented earlier relates the frame {7} translational and rotational velocities with respect to the {0} frame. For real-world applications, it is more useful to command translational and rotational velocities at the end-effector frame {*G*} instead. The same Jacobian may be used if the following velocity transformations are used first:

$$\{V_7\} = \{V_G\} - \{\omega_G\} \times \{{}^7P_G\}$$
$$\{\omega_7\} = \{\omega_G\}$$

That is, since link 7 is a rigid link containing both {7} and {*G*}, the rotational velocity vector is the same over the whole link. However, the angular velocity crossed into the position vector must be added to the translational velocity in {7} to yield that of {*G*} (this statement must be reversed, i.e. subtracted, since we are given the velocities in {*G*}). As always, a common basis frame, such as {0}, must be used above to ensure the coordinates of all vectors are expressed in a single frame.

Cartesian Wrench / Joint Torques Statics Transformation

It is well-known (Craig, 2005) that the relationship between static Cartesian wrenches (forces / moments) applied to the environment by the robot end-effector and the required robot joint torques to do this are calculated as follows:

$$\{\tau\} = [J]^T \{W\}$$

Where $\{\tau\} \equiv n \times 1$ is the vector of *n* joint torques, $\{W\} \equiv m \times 1$ is the Cartesian wrench (*m* forces and moments), *n* is the joint space dimension, and *m* is the Cartesian space dimension. The Jacobian matrix [*J*] and Cartesian wrench {*W*} must be expressed in the basis coordinates of the same frame {*k*}. Just like the joint rates, {τ} has no dependence on frame, since these are relative joint torques about the *n Z* axes.

The Jacobian matrix [*J*] for static torque calculations is the same as that for velocity analysis. This statics transformation is a mapping from Cartesian space to joint space which does not require an inverse. That is indeed a rare and beautiful property. It can never be singular and any number of joints is allowed. Very little computation is required compared to matrix inversion, since only matrix transposition and multiplication is required. Further, this wrench/joint torques transformation equally applies to kinematically-redundant robots such as the 7-dof Baxter arms. Note that the associated wrench is applied at the frame for which the Jacobian matrix was derived for velocities, {7} in our work. Therefore, another transformation is required when the robot should apply the wrench at {*G*} (again using a common basis):

$$\{F_7\} = \{F_G\}$$
$$\{M_7\} = \{M_G\} + \{{}^7P_G\} \times \{F_G\}$$

Analytical Jacobian Matrices for the Baxter Robot Arm

The Baxter Robot arm Jacobian matrix is given below, for the general 7-dof case with $L_5 \neq 0$. This Jacobian matrix expresses the velocity of {7} with respect to {0}, expressed in {4} coordinates. Generally if the frame of expression is midway between {7} and {1} (i.e. {4}), the resulting analytical Jacobian matrix terms will be the simplest possible.

$$\left[{}^4J\right] = \begin{bmatrix} j_{11} & (L_2c_4 + L_3s_4 + L_4)c_3 - L_5s_3s_4s_5 & -L_5c_4s_5 & L_4 & -L_5s_5 & 0 & 0 \\ j_{21} & (-L_2s_4 + L_3c_4 + L_5c_5)c_3 - L_5s_3c_4s_5 & L_5s_4s_5 & L_5c_5 & 0 & 0 & 0 \\ j_{31} & -(L_2 + L_4c_4 - L_5s_4c_5)s_3 & L_3 + L_4s_4 + L_5c_4c_5 & 0 & L_5c_5 & 0 & 0 \\ s_2s_4 - c_2c_3c_4 & s_3c_4 & -s_4 & 0 & 0 & -s_5 & c_5s_6 \\ s_2c_4 + c_2c_3s_4 & -s_3s_4 & -c_4 & 0 & -1 & 0 & -c_6 \\ c_2s_3 & c_3 & 0 & 1 & 0 & c_5 & s_5s_6 \end{bmatrix}$$

where:

$$j_{11} = ((L_1 + L_2c_2)c_4 + (L_3s_4 + L_4)c_2)s_3 + L_5(s_2c_4 + c_2c_3s_4)s_5$$
$$j_{21} = -((L_1 + L_2c_2)s_4 - L_3c_2c_4)s_3 - L_5(s_2s_4 - c_2c_3c_4)s_5 + L_5c_2s_3c_5$$
$$j_{31} = (L_1 + L_2c_2)c_3 - L_3s_2 - L_4(s_2s_4 - c_2c_3c_4) - L_5(s_2c_4 + c_2c_3s_4)c_5$$

The Baxter Robot arm Jacobian matrix is given below, for the 7-dof case with $L_5 = 0$. This Jacobian matrix expresses the velocity of {7} with respect to {0}, again expressed in {4} coordinates for simplest analytical terms.

$$\left[{}^4J\right] = \begin{bmatrix} ((L_1 + L_2c_2)c_4 + (L_3s_4 + L_4)c_2)s_3 & (L_2c_4 + L_3s_4 + L_4)c_3 & 0 & L_4 & 0 & 0 & 0 \\ -((L_1 + L_2c_2)s_4 - L_3c_2c_4)s_3 & (-L_2s_4 + L_3c_4)c_3 & 0 & 0 & 0 & 0 & 0 \\ (L_1 + L_2c_2)c_3 - L_3s_2 - L_4(s_2s_4 - c_2c_3c_4) & -(L_2 + L_4c_4)s_3 & L_3 + L_4s_4 & 0 & 0 & 0 & 0 \\ s_2s_4 - c_2c_3c_4 & s_3c_4 & -s_4 & 0 & 0 & -s_5 & c_5s_6 \\ s_2c_4 + c_2c_3s_4 & -s_3s_4 & -c_4 & 0 & -1 & 0 & -c_6 \\ c_2s_3 & c_3 & 0 & 1 & 0 & c_5 & s_5s_6 \end{bmatrix}$$

The Baxter Robot arm Jacobian matrix is given below, for the 6-dof case with $L_5 = 0$. This Jacobian matrix expresses the velocity of {7} with respect to {0}, expressed in {0} coordinates.

$$\left[{}^0J\right] = \begin{bmatrix} -(L_1 + L_hc_2 + L_4c_{24})s_1 & -(L_hs_2 + L_4s_{24})c_1 & -L_4c_1s_{24} & 0 & 0 & 0 \\ (L_1 + L_hc_2 + L_4c_{24})c_1 & -(L_hs_2 + L_4s_{24})s_1 & -L_4s_1s_{24} & 0 & 0 & 0 \\ 0 & -(L_hc_2 + L_4c_{24}) & -L_4c_{24} & 0 & 0 & 0 \\ 0 & -s_1 & -s_1 & c_1c_{24} & -s_1c_5 + c_1s_{24}s_5 & c_1c_{24}c_6 - (s_1s_5 + c_1s_{24}c_5)s_6 \\ 0 & c_1 & c_1 & s_1c_{24} & c_1c_5 + s_1s_{24}s_5 & s_1c_{24}c_6 + (c_1s_5 - s_1s_{24}c_5)s_6 \\ 1 & 0 & 0 & -s_{24} & c_{24}s_5 & -(s_{24}c_6 + c_{24}c_5s_6) \end{bmatrix}$$

The following Jacobian matrix is for the same 6-dof Baxter arm, with $L_5 = 0$. This Jacobian still expresses the velocity of {7} with respect to {0}, but expressed in {4} coordinates.

$$\left[{}^4J\right] = \begin{bmatrix} (L_1 + L_h c_2 + L_4 c_{24}) s_5 & (L_h c_4 + L_4) c_5 & L_4 c_5 & 0 & 0 & 0 \\ (L_1 + L_h c_2 + L_4 c_{24}) c_5 & -(L_h c_4 + L_4) s_5 & -L_4 s_5 & 0 & 0 & 0 \\ 0 & L_h s_4 & 0 & 0 & 0 & 0 \\ -c_{24} c_5 & s_5 & s_5 & 0 & 0 & s_6 \\ c_{24} s_5 & c_5 & c_5 & 0 & 1 & 0 \\ -s_{24} & 0 & 0 & 1 & 0 & c_6 \end{bmatrix}$$

Baxter Arm Singularity Analysis (6-dof case, $L_5 = 0$)

Both $\left[{}^0J\right]$ and $\left[{}^4J\right]$ yield the same determinant, since matrix determinant is invariant under coordinate rotation transformations (any basis frame *k* will yield the same matrix determinant):

$$\left|{}^0J\right| = \left|{}^4J\right| = \left|{}^kJ\right| = -L_h L_4 (L_1 + L_h c_2 + L_4 c_{24}) s_4 s_6$$

When this Jacobian matrix determinant is zero, the Baxter arm (6-dof case, $L_5 = 0$) is at a robot singularity. Since $L_h \neq 0$ and $L_4 \neq 0$ for the Baxter robot arm, we have three singularity conditions:

$$L_1 + L_h \cos\theta_2 + L_4 \cos(\theta_2 + \theta_4) = 0$$

$$\sin\theta_4 = 0$$

$$\sin\theta_6 = 0$$

The second two singularity conditions are easy to analyze, yielding workspace-boundary singularity $\theta_4 = 0, 180^\circ, \cdots$, and workspace-interior singularity $\theta_6 = 0, 180^\circ, \cdots$. The former is the well-known elbow-out (or -folded) singularity at the edge of the workspace, and the latter is the well-known wrist-pitch singularity for a roll-pitch-roll wrist (only the $L_5 = 0$ case is this simple; including $L_5 \neq 0$ such as in Baxter moves this well-known singularity to an unknown place, even for small L_5).

The first singularity condition $L_1 + L_h \cos\theta_2 + L_4 \cos(\theta_2 + \theta_4) = 0$ leads to a workspace-interior singularity that is not so easy to analyze. The practical Baxter shoulder and elbow joint angle limits are $-123^\circ \leq \theta_2 \leq 60^\circ$ and $-3^\circ \leq \theta_4 \leq 150^\circ$. For these limits the figure below shows the singularity surface $L_1 + L_h \cos\theta_2 + L_4 \cos(\theta_2 + \theta_4)$ as a function of θ_2 and θ_4. This singularity occurs whenever this surface crosses through zero.

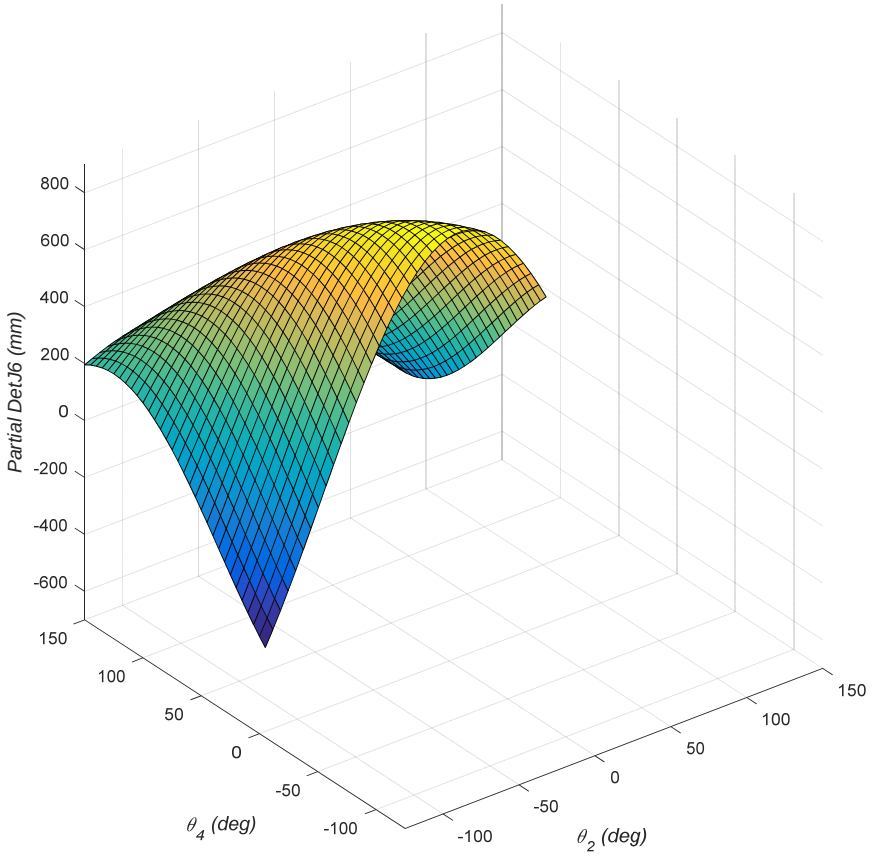


Figure 12a. $L_1 + L_h \cos\theta_2 + L_4 \cos(\theta_2 + \theta_4)$ **Singularity Surface**

For an analytical solution for $L_1 + L_h \cos\theta_2 + L_4 \cos(\theta_2 + \theta_4) = 0$, let us specify θ_4 as the independent variable, vary it over the joint limits $\theta_{4MIN} \le \theta_4 \le \theta_{4MAX}$, and calculate the (two) θ_2 values leading to a singularity for each θ_4. Expanding $\cos(\theta_2 + \theta_4)$ using the sum-of-angles formula leads to the following equation to solve for θ_2.

$$E\cos\theta_2 + F\sin\theta_2 + G = 0$$

where:

$$E = L_h + L_4 c_4$$
$$F = -L_4 s_4$$
$$G = L_1$$

This equation form was encountered earlier in the analytical IPK solution; two values of θ_2 for each θ_4 can be solved, in the same manner, for the updated *E*, *F*, *G* above. This solution is overlaid as black topographic curves on the surface; this is shown in the figure below. This is a top view of the same surface shown earlier, with the same color scheme.

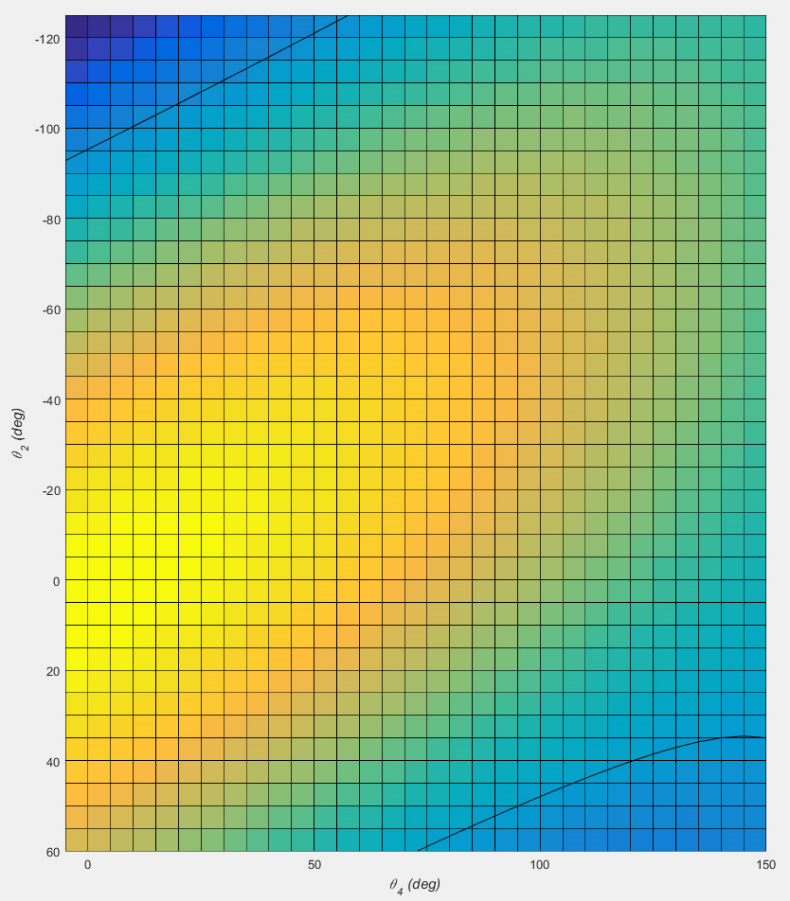


Figure 12b. $L_1 + L_h \cos\theta_2 + L_4 \cos(\theta_2 + \theta_4)$ **Singularity Surface with Zero Curves**

At first these appear to be a straight-forward line and curve; however, as shown below ignoring joint limits ($-180^\circ \le \theta_2, \theta_4 \le 180^\circ$), the singularity curves are very complicated (albeit symmetric).

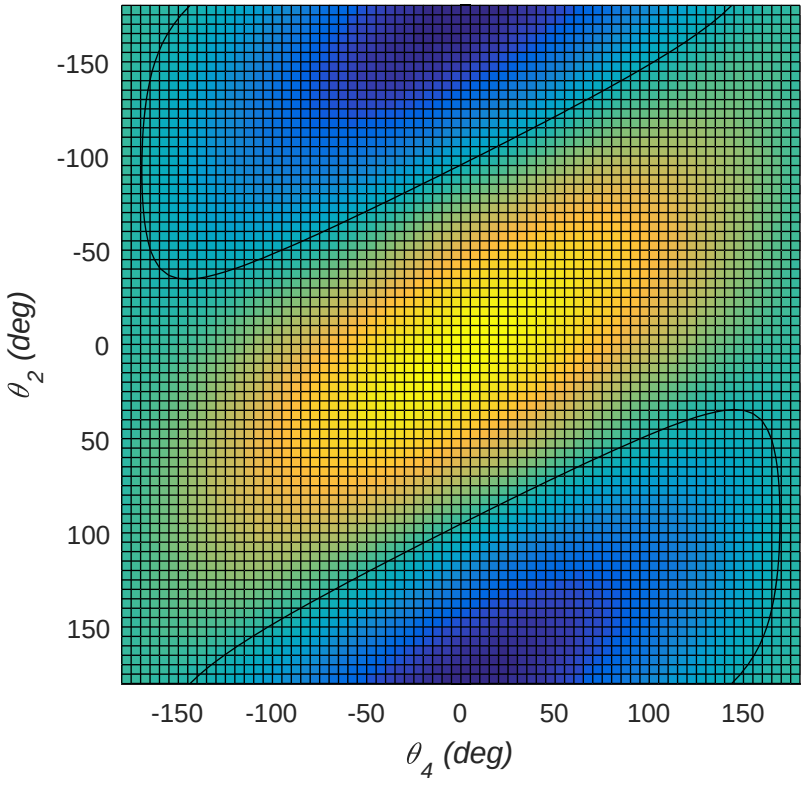


Figure 12c.
$L_1 + L_h \cos\theta_2 + L_4 \cos(\theta_2 + \theta_4)$ **Singularity Surface with Zero Curves for** $-180^\circ \le \theta_2, \theta_4 \le 180^\circ$

The following Jacobian matrix is for the 6-dof Baxter arm, again with the third joint angle locked to $\theta_3 = 0$, but this time with L_5 included, again expressed in {4} coordinates for simplest analytical terms.

$$\left[{}^4J\right] = \begin{bmatrix} (L_1 + L_h c_2 + L_4 c_{24}) s_5 & (L_h c_4 + L_4) c_5 & L_4 c_5 & 0 & 0 & 0 \\ (L_1 + L_h c_2 + L_4 c_{24}) c_5 - L_5 s_{24} & -(L_h c_4 + L_4) s_5 & -L_4 s_5 & L_5 & 0 & 0 \\ -L_5 c_{24} s_5 & L_h s_4 - L_5 c_5 & -L_5 c_5 & 0 & 0 & 0 \\ -c_{24} c_5 & s_5 & s_5 & 0 & 0 & s_6 \\ c_{24} s_5 & c_5 & c_5 & 0 & 1 & 0 \\ -s_{24} & 0 & 0 & 1 & 0 & c_6 \end{bmatrix}$$

Baxter Left Arm Singularity Analysis (6-dof case, L_5 included)

With this one simple change, $L_5 \neq 0$, the Jacobian matrix determinant and hence singularity analysis, becomes quite complicated:

$$\left|{}^4J\right| = \{-L_4(L_1 + L_h c_2 + L_4 c_{24}) s_4 s_6 + [(L_1 + L_h c_2)(-c_4 c_5 s_6 + s_4 s_5^2 c_6) + L_4 c_{24}(s_4 c_6 - c_4 c_5 s_6) + L_5 c_{24} c_4 c_5 c_6] L_5\} L_h$$

First, we see that substituting $L_5 = 0$ in the above expression yields the previous-case determinant $-L_h L_4 (L_1 + L_h c_2 + L_4 c_{24}) s_4 s_6$, providing at least partial validation of the new determinant.

The new $L_5 \neq 0$ determinant:

- Is now a function of 4 variables instead of 2 in the previous complicated analytical analysis.
- No longer has $s_4 s_6$ factored out, so those easily-known elbow and wrist singularities have changed. They did not disappear, they just moved to locations very difficult to determine.
- Is essentially impossible to analyze analytically. Screw Theory provides an elegant method to identify singularities, but when applied to a robot arm with offsets like Baxter, there is no magic bullet, i.e. Screw Theory will still yield a hyper-complicated analytical formula for singularity analysis.

7.2 Resolved-Rate Control Method for Kinematically-Redundant Arms

A super-useful application for the Jacobian matrix is the **Inverse Velocity Solution**, which is the basis for the **Resolved-Rate Control Algorithm** (Whitney, 1969). The Inverse Velocity Solution for non-kinematically-redundant robots is:

$$\left\{\dot{\Theta}\right\} = \left[{}^{k}J(\Theta)\right]^{-1}\left\{{}^{k}\dot{X}\right\}$$

where we calculate the required relative joint rates $\left\{\dot{\Theta}\right\}$ to achieve the given desired Cartesian velocities $\left\{{}^{k}\dot{X}\right\}$, using the inverse of the configuration-dependent Jacobian matrix $\left[{}^{k}J(\Theta)\right]$. This works for an $m = n$ square matrix, assuming full rank, i.e. the Jacobian matrix determinant is not zero.

For an $n > m$ kinematically-redundant robot arm such as Baxter, the plain matrix inverse will not work in the inverse velocity solution above, since the matrix inverse is not defined for non-square Jacobian matrices (6 x 7). This case is underconstrained, i.e. more unknowns than equations, and so optimization may be accomplished in addition to satisfying Cartesian motions. This is because there are infinite solutions to the kinematically-redundant inverse velocity problem.

Again, a kinematically-redundant robot (KRR) has more joint freedoms than required for the Cartesian task. That is, the dimension of the joint space n is greater than the dimension of Cartesian task space m.

$m < n$ degree of redundancy = $n - m$

The inverse velocity problem is ***underconstrained***, that is, infinite valid solutions exist (actually, $n - m$ infinities of solutions exist).

Self-motion – additional task optimization takes place in joint rate combinations such that $\left\{\dot{X}\right\} = \{0\}$.

Primary Task	Particular Solution	achieves required trajectory $\left\{\dot{X}\right\}$
Secondary Task	Homogeneous Solution	optimization of some performance criteria in null space (self-motion) $\left\{\dot{X}\right\} = \{0\}$

Pseudoinverse-based Inverse Velocity Solution

Since the plain matrix inverse is undefined for a non-square Jacobian matrix, we will use the well-known pseudoinverse of the Jacobian matrix. This will be used in both the Particular and Homogeneous Solutions below.

Particular Solution

The particular solution satisfies the **Primary Task**, i.e. it provides the commanded Cartesian trajectory $\left\{\dot{X}\right\}$. The particular solution is obtained from the minimum joint rate problem, stated as follows. That is, from the infinite possible solutions we will choose the unique solution with the minimum joint rates.

Minimize the scalar cost function $$f = \frac{1}{2}\left\{\dot{\Theta}\right\}^T \left\{\dot{\Theta}\right\}$$

Subject to the rate equations $$\left\{\dot{X}\right\} = \left[J\right]\left\{\dot{\Theta}\right\}$$

The particular solution is:

$$\left\{\dot{\Theta}_P\right\} = \left[{}^k J(\Theta)\right]^* \left\{{}^k \dot{X}\right\}$$

where $\left[J\right]^*$ is the underconstrained **Moore-Penrose pseudoinverse** of the Jacobian matrix:

$$\left[J\right]^* = \left[J\right]^T \left[\left[J\right]\left[J\right]^T\right]^{-1}$$

The dimensions of this pseudoinverse are *n* x *m* (7 x 6 for the Baxter robot arm). One beautiful aspect of this formula is that, no matter how many joints beyond *n* = 7 exist in a KRR, the worst matrix inversion is for an *m* x *m* square matrix (6 x 6).

The pseudoinverse $\left[J\right]^*$ can be calculated by the MATLAB function **`pinv`**. Note that $\left[J\right]\left[J\right]^* = \left[I_m\right]$ but $\left[J\right]^*\left[J\right] \neq \left[I_n\right]$ so the underconstrained Moore-Penrose pseudoinverse is called the right inverse of $\left[J\right]$. As a matter of fact we do not want $\left[J\right]^*\left[J\right]$ to be $\left[I_n\right]$, otherwise there would be no null space for additional task optimization (there would be no self-motion).

Note that the left inverse of an overconstrained matrix may be used to find the least-squares solution, where there is no solution, but a solution with the smallest possible error is found (for data fitting). This is opposite to the KRR problem we are considering.

Homogeneous Solution

The homogeneous solution satisfies the **Secondary Task**, which can be used for optimization to avoid obstacles, avoid joint limits, and avoid singularities, among other secondary tasks. The homogeneous solution doesn't affect the Cartesian trajectory: $\left\{\dot{\Theta}_H\right\}$ causes $\left\{\dot{X}\right\} = \{0\}$. The homogeneous solution projects an arbitrary vector *z* into the null-space of the Jacobian matrix. The homogeneous solution is:

$$\left\{\dot{\Theta}_H\right\} = k_H \left[\left[I_n\right] - \left[J\right]^* \left[J\right]\right]\{z\}$$

where k_H is a scalar gain, and $\{z\}$ is an arbitrary *n* x 1 vector. For optimization, choose $\{z\} = \{\nabla H(\Theta)\}$, where $\{H(\Theta)\}$ is an objective function of joint angles to be minimized or maximized. Use $k_H > 0$ for maximization and $k_H < 0$ for minimization.

Joint Limit Avoidance $$H_J(\Theta) = \sum_{i=1}^{n} \left(\frac{\theta_i - \theta_{ci}}{\Delta\theta_i} \right)^2$$

Singularity Avoidance (Manipulability Maximization) $$H_M(\Theta) = \sqrt{\left|\left[J\right]\left[J\right]^T\right|}$$

Total Solution

The total solution to the underconstrained KRR inverse velocity problem is the sum of the **particular** and **homogeneous solutions** (obtained via linear superposition):

$$\left\{\dot{\Theta}\right\} = \left\{\dot{\Theta}_P\right\} + \left\{\dot{\Theta}_H\right\}$$

Kinematically-Redundant Robot Singularities

The Moore-Penrose pseudoinverse is still subject to singularities. **Singular Value Decomposition (SVD)**, can be used in the vicinity of singularities to obtain a reliable numerical pseudoinverse to move through singularities, but it will not change the physical robot singularity (loss of motion).

A kinematically-redundant robot is singular when the following determinant is zero.

$$\left|\left[J\right]\left[J\right]^T\right| = 0$$

due to the previously-given definition of the underconstrained Moore-Penrose pseudoinverse $\left[J\right]^*$.

8. Baxter Robot Arm Kinematics Examples

Left/Right Arms Zero Configuration FPK Example

Given $\begin{bmatrix}\theta_1 & \theta_2 & \theta_3 & \theta_4 & \theta_5 & \theta_6 & \theta_7\end{bmatrix} = \begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}$ for both left and right arms.

For such a simple example (all zero joint angles) we can predict the answers, for validation of the MATLAB FPK implementation. The active joints homogeneous transformation matrix $\left[{}^{0}_{7}T\right]$ is identical for the left and right arms, but the overall homogeneous transformation matrices $\left[{}^{Wo}_{G}T\right]$ are different (though there is significant symmetry evident).

$$\left[{}^{0}_{7}T\right] = \begin{bmatrix} 0 & 0 & 1 & L_1 + L_2 + L_4 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & -L_3 - L_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \left[{}^{0}_{7}T\right] = \begin{bmatrix} 0 & 0 & 1 & 0.808 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & -0.079 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\left[{}^{Wo}_{GL}T\right] = \begin{bmatrix} 0 & 0.707 & 0.707 & 1.110 \\ 0 & 0.707 & -0.707 & -0.896 \\ -1 & 0 & 0 & 1.295 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \left[{}^{Wo}_{GR}T\right] = \begin{bmatrix} 0 & 0.707 & -0.707 & -1.110 \\ 0 & -0.707 & -0.707 & -0.896 \\ -1 & 0 & 0 & 1.295 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

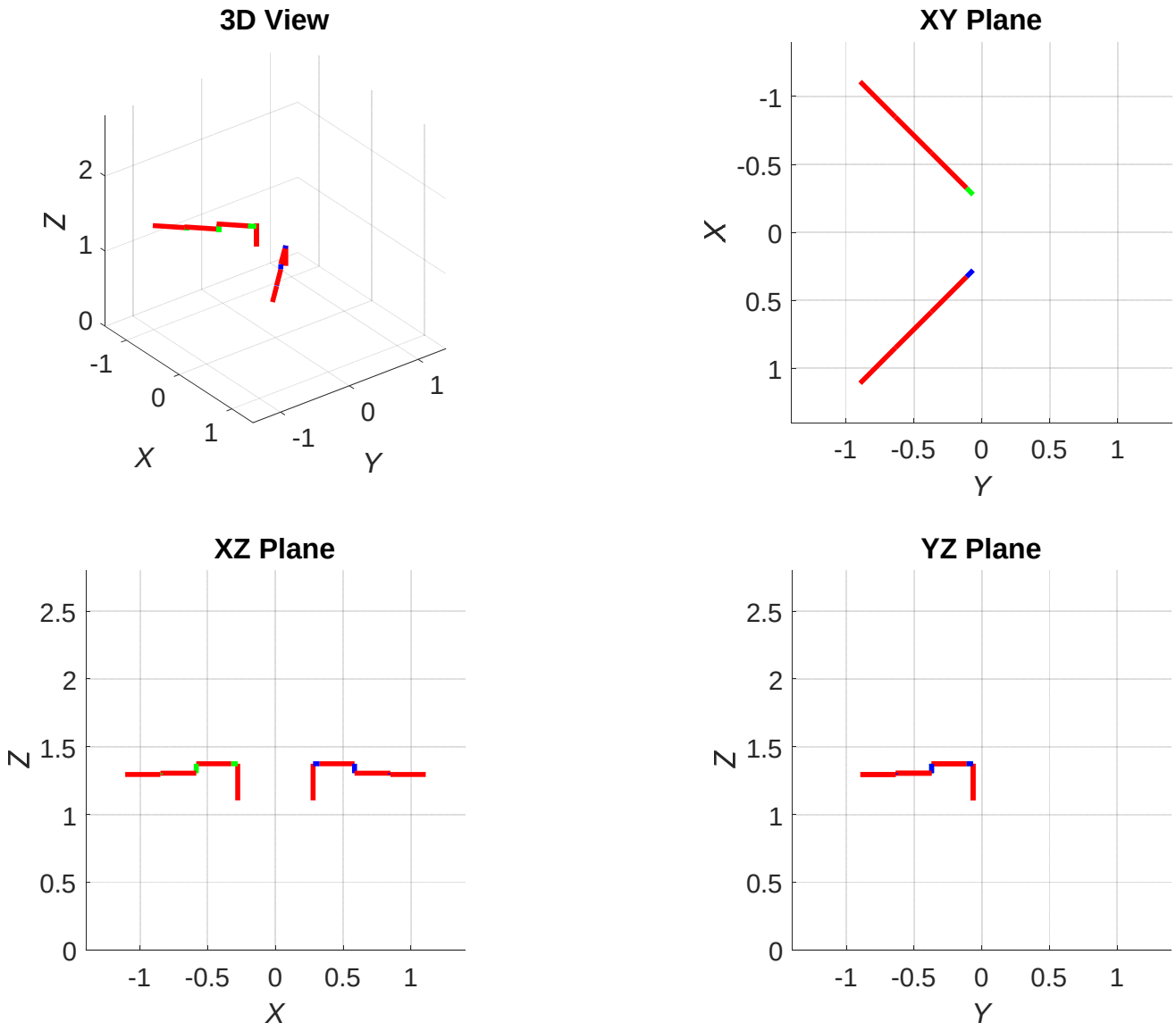


Figure 13a. 7-dof Zero Configuration FPK Example

Left/Right Arms Neutral Configuration FPK Example

Given $\begin{bmatrix}\theta_1 & \theta_2 & \theta_3 & \theta_4 & \theta_5 & \theta_6 & \theta_7\end{bmatrix} = \begin{bmatrix}0 & -31^\circ & 0 & 43^\circ & 0 & 72^\circ & 0\end{bmatrix}$ for both left and right arms.

Again, for this Neutral joint angles input example, the active joints homogeneous transformation matrix $\begin{bmatrix}{}^{0}_{7}T\end{bmatrix}$ is identical for the left and right arms, but the overall homogeneous transformation matrices $\begin{bmatrix}{}^{Wo}_{G}T\end{bmatrix}$ are different (with significant symmetry).

$$\begin{bmatrix}{}^{0}_{7}T\end{bmatrix} = \begin{bmatrix} -0.995 & 0 & 0.105 & 0.781 \\ 0 & 1 & 0 & 0 \\ -0.105 & 0 & -0.995 & 0.041 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix}{}^{Wo}_{GL}T\end{bmatrix} = \begin{bmatrix} -0.703 & 0.707 & 0.074 & 0.857 \\ 0.703 & 0.707 & -0.074 & -0.643 \\ -0.105 & 0 & -0.995 & 1.049 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix}{}^{Wo}_{GR}T\end{bmatrix} = \begin{bmatrix} 0.703 & 0.707 & -0.074 & -0.857 \\ 0.703 & -0.707 & -0.074 & -0.643 \\ -0.105 & 0 & -0.995 & 1.049 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

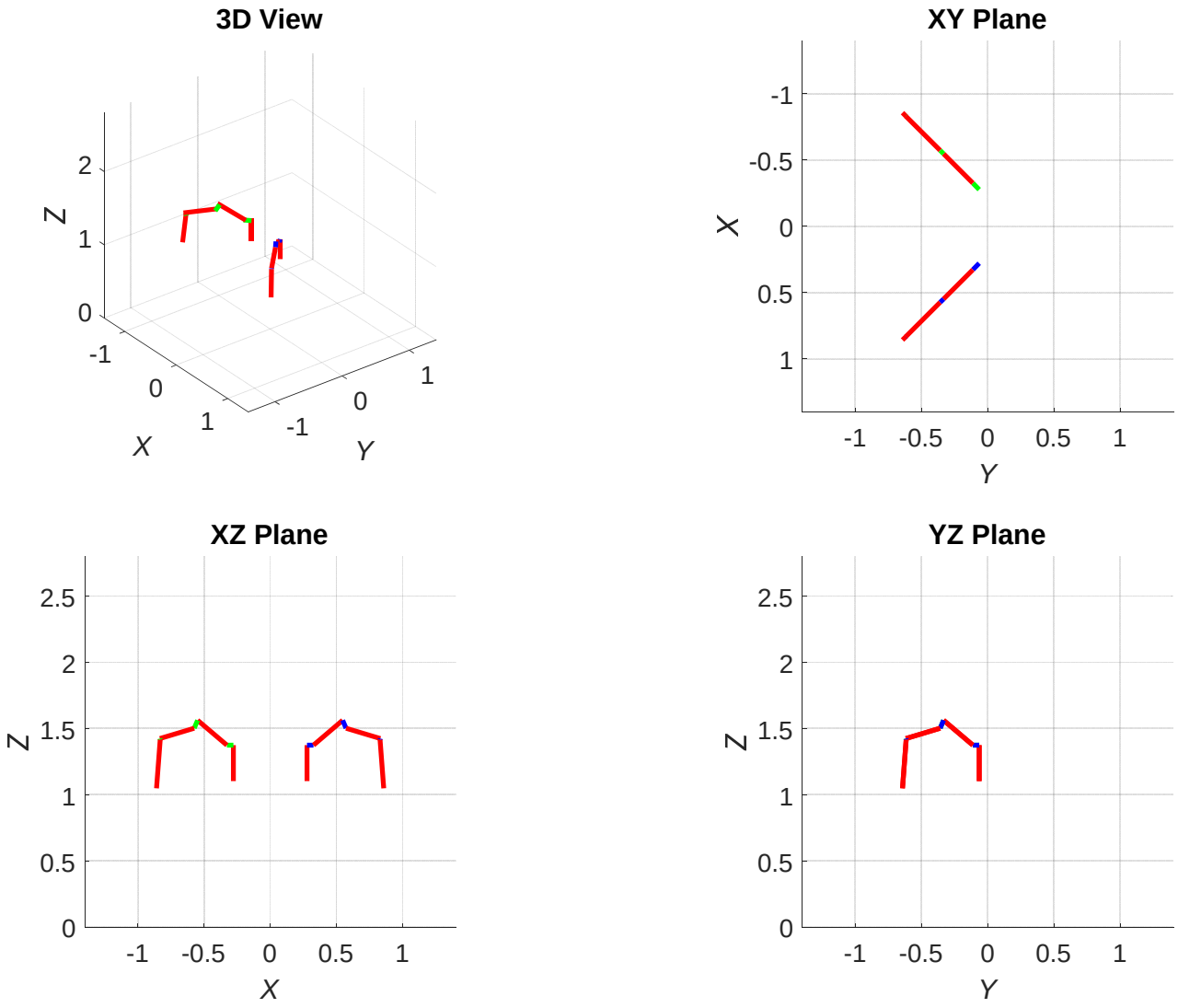


Figure 13b. 7-dof Neutral Configuration FPK Example

Left/Right Arms General FPK Example

Given $\begin{bmatrix}\theta_1 & \theta_2 & \theta_3 & \theta_4 & \theta_5 & \theta_6 & \theta_7\end{bmatrix} = \begin{bmatrix}10^\circ & 20^\circ & 30^\circ & 40^\circ & 50^\circ & 60^\circ & 70^\circ\end{bmatrix}$ for both left and right arms.

Again, the active joints homogeneous transformation matrix $\left[{}^{0}_{7}T\right]$ is identical for the left and right arms. The matrices $\left[{}^{Wo}_{G}T\right]$ are different (with some symmetry and identical terms).

$$\left[{}^{0}_{7}T\right] = \begin{bmatrix} 0.415 & 0.875 & -0.248 & 0.548 \\ 0.386 & 0.077 & 0.919 & 0.263 \\ 0.824 & -0.477 & -0.306 & -0.474 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\left[{}^{Wo}_{GL}T\right] = \begin{bmatrix} 0.566 & 0.674 & 0.475 & 1.026 \\ -0.021 & -0.564 & 0.825 & 0.039 \\ 0.824 & -0.477 & -0.306 & 0.788 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \left[{}^{W0}_{GR}T\right] = \begin{bmatrix} -0.021 & -0.564 & 0.825 & -0.175 \\ -0.566 & -0.674 & -0.475 & -0.812 \\ 0.824 & -0.477 & -0.306 & 0.788 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

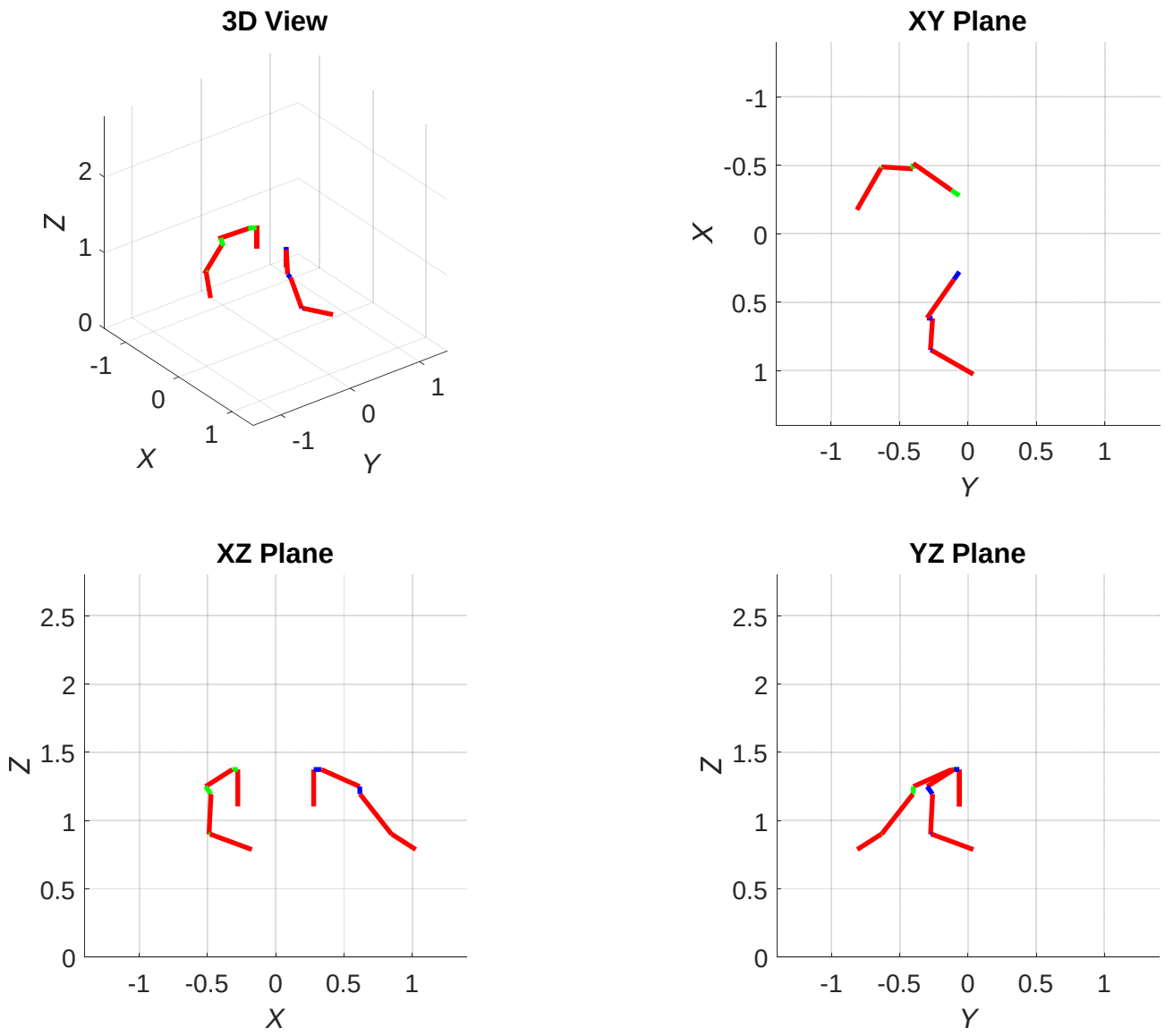


Figure 13c. 7-dof General Configuration FPK Example

Left Arm 6-dof FPK Example, Zero Configuration

Given $\begin{bmatrix}\theta_1 & \theta_2 & \theta_4 & \theta_5 & \theta_6 & \theta_7\end{bmatrix} = \begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}$ and assuming $\theta_3 = 0$ for the left arm ($\begin{bmatrix}{}^{0}_{7}T\end{bmatrix}$ is identical for the right arm), the FPK solution is given below.

It is also validated using the standard 7-dof FPK solution, with modified θ_2 and θ_4 inputs, in order to align the two kinematics models. For such a simple example (all zero joint angles) we can predict the answers, for validation of the MATLAB FPK implementation.

$$\begin{bmatrix}{}^{0}_{7}T\end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & L_1 + L_h + L_4 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & -L_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix}{}^{0}_{7}T\end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0.814 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & -0.010 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix}{}^{Wo}_{GL}T\end{bmatrix} = \begin{bmatrix} 0 & 0.707 & 0.707 & 1.114 \\ 0 & 0.707 & -0.707 & -0.900 \\ -1 & 0 & 0 & 1.364 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

To validate this result using the standard 7-dof FPK solution, we must take into account the negative θ_2 offset required to align joints 2 and 4 horizontally (see Figure 7):

$$\theta_{2\text{off}} = \tan^{-1}\left[\frac{L_3}{L_2}\right]$$

Given $\begin{bmatrix}\theta_1 & \theta_2 & \theta_3 & \theta_4 & \theta_5 & \theta_6 & \theta_7\end{bmatrix} = \begin{bmatrix}0 & -10.73^\circ & 0 & 10.73^\circ & 0 & 0 & 0\end{bmatrix}$

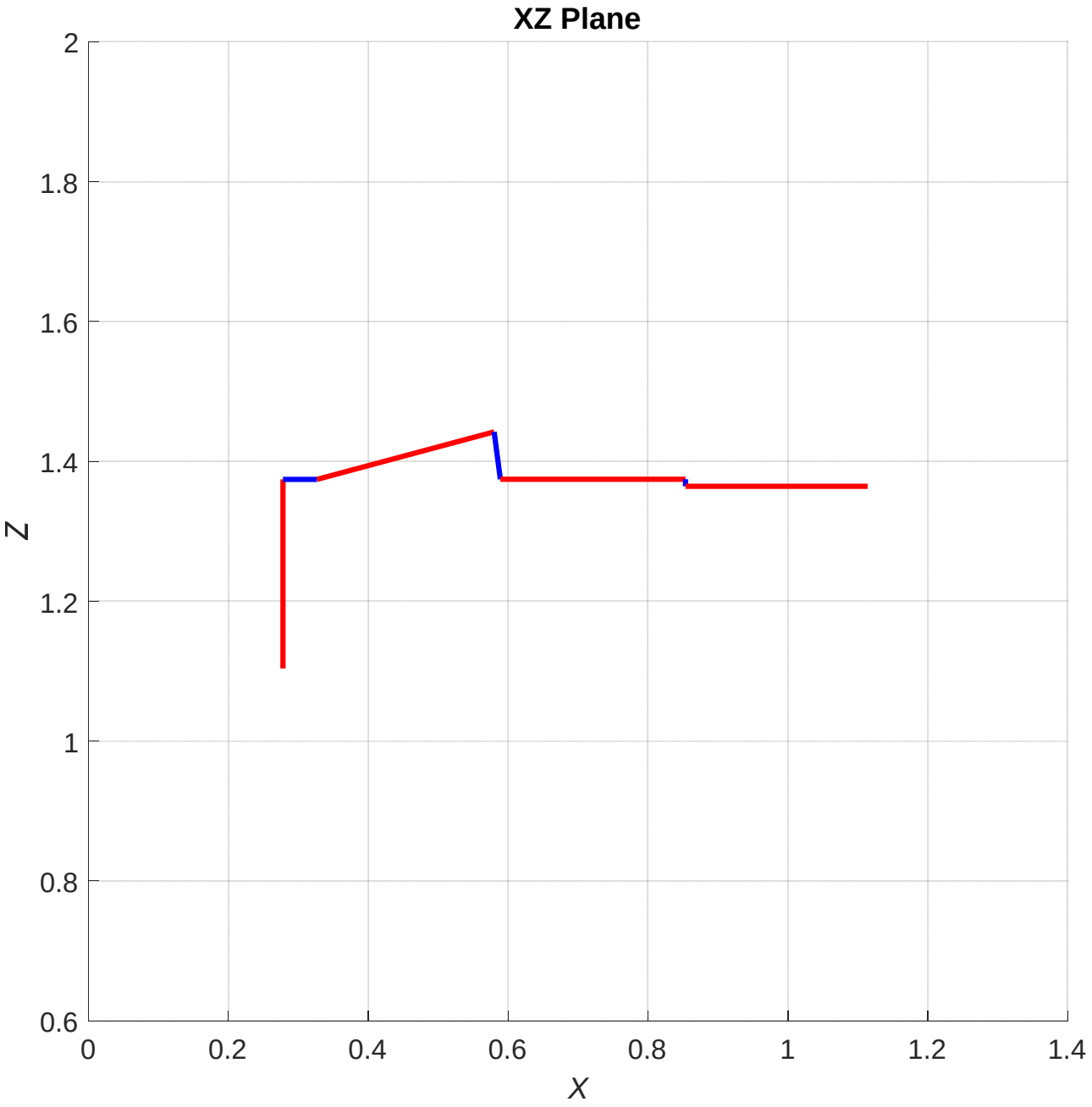


Figure 14a. 6-dof Zero Configuration FPK Example

Left Arm 6-dof General FPK Example

Given $\begin{bmatrix}\theta_1 & \theta_2 & \theta_4 & \theta_5 & \theta_6 & \theta_7\end{bmatrix} = \begin{bmatrix}10^\circ & 20^\circ & 40^\circ & 50^\circ & 60^\circ & 70^\circ\end{bmatrix}$ for the left arm, assuming $\theta_3 = 0$, the FPK solution is given below. Note that this example also assumes $L_5 = 0$, so that it can be used in a circular check with the 6-dof arm analytical IPK example presented later.

$$\left[{}^{0}_{6}T\right] = \begin{bmatrix} 0.247 & 0.906 & -0.344 & 0.595 \\ 0.790 & 0.018 & 0.613 & 0.105 \\ 0.562 & -0.423 & -0.711 & -0.451 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\left[{}^{Wo}_{GL}T\right] = \begin{bmatrix} 0.733 & 0.653 & 0.190 & 0.843 \\ 0.384 & -0.628 & 0.677 & -0.162 \\ 0.562 & -0.423 & -0.711 & 0.661 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

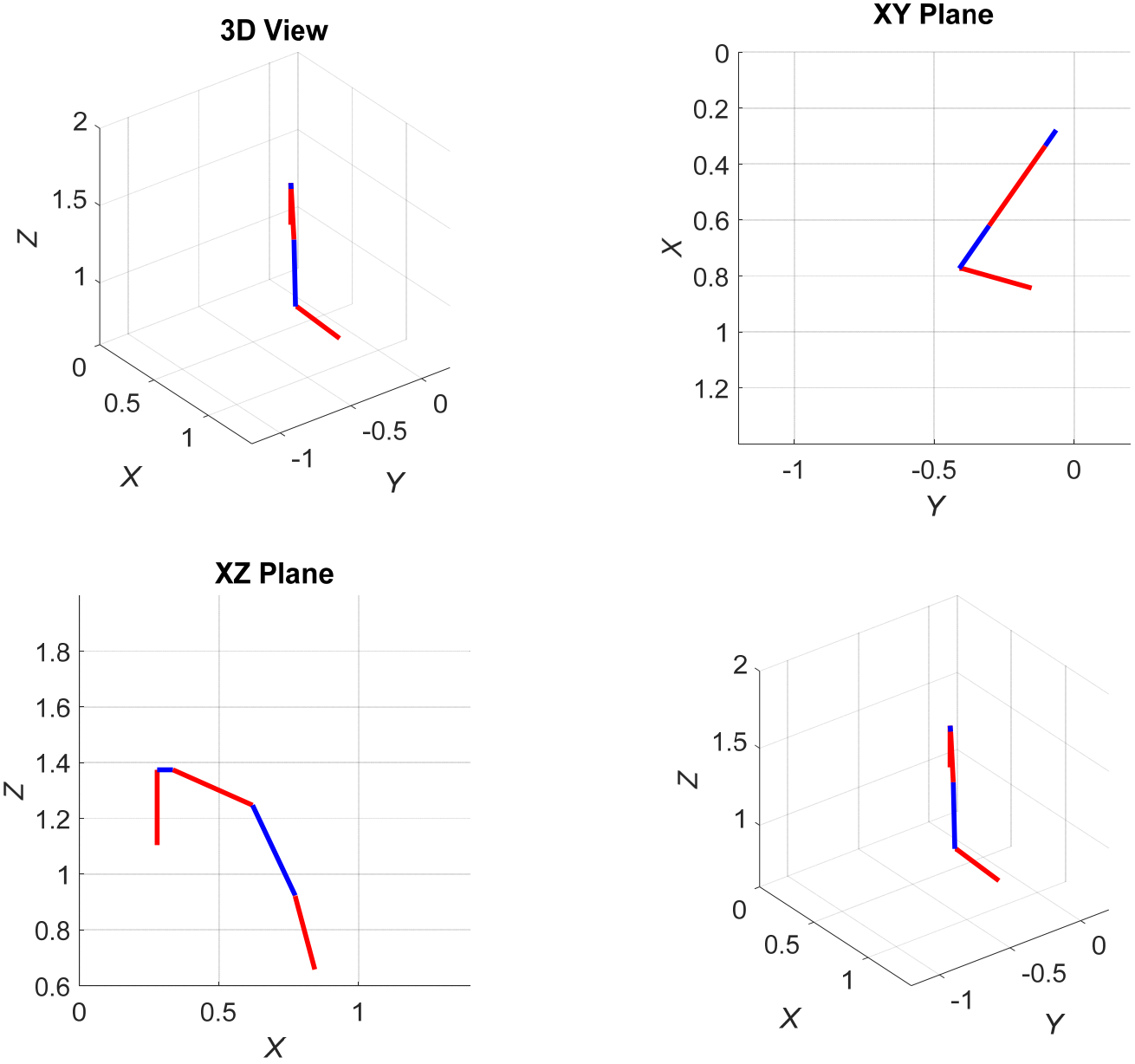


Figure 14b. 6-dof General Configuration FPK Example

Left Arm 6-dof General IPK Example

This example corresponds with the general left-arm 6-dof FPK example given above. It assumes $\theta_3 = 0$ and $L_5 = 0$.

Given:

$$\left[{}^{Wo}_{GL}T\right] = \begin{bmatrix} 0.733 & 0.653 & 0.190 & 0.843 \\ 0.384 & -0.628 & 0.677 & -0.162 \\ 0.562 & -0.423 & -0.711 & 0.661 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

we first must use $\left[{}^{0}_{6}T\right] = \left[{}^{Wo}_{0}T\right]^{-1}\left[{}^{Wo}_{GL}T\right]\left[{}^{6}_{GL}T\right]^{-1}$ to calculate the simpler IPK input $\left[{}^{0}_{6}T\right]$:

$$\left[{}^{0}_{6}T\right] = \begin{bmatrix} 0.247 & 0.906 & -0.344 & 0.595 \\ 0.790 & 0.018 & 0.613 & 0.105 \\ 0.562 & -0.423 & -0.711 & -0.451 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Applying the equations from the analytical 6-dof IPK solution presented earlier, the solutions are given in the table below (the units are degrees).

Solution	θ_1	θ_2	θ_4	θ_5	θ_6	θ_7
elbow up	10	20	40	50	60	70
elbow down	10	60.2	–40	41.6	88.4	99.4

The first solution set was expected since that was the known FPK input for this example pose. The second solution set was shown to be valid by using it as the input to the 6-dof FPK program and proving that the answer was the correct given $\left[{}^{Wo}_{GL}T\right]$ (this is the circular check).

Left/Right 7-dof Arms General IPK Example

This example corresponds with the general left-/right-arm 7-dof FPK example given above.

Given:

$$\left[{}^{Wo}_{GL}T\right]=\begin{bmatrix} 0.566 & 0.674 & 0.475 & 1.026 \\ -0.021 & -0.564 & 0.825 & 0.039 \\ 0.824 & -0.477 & -0.306 & 0.788 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

we first must use $\left[{}^{0}_{7}T\right]=\left[{}^{Wo}_{0}T\right]^{-1}\left[{}^{Wo}_{GL}T\right]\left[{}^{7}_{GL}T\right]^{-1}$ to calculate the simpler IPK input $\left[{}^{0}_{7}T\right]$:

$$\left[{}^{0}_{7}T\right]=\begin{bmatrix} 0.415 & 0.875 & -0.248 & 0.548 \\ 0.386 & 0.077 & 0.919 & 0.263 \\ 0.824 & -0.477 & -0.306 & -0.474 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Applying the numerical Newton-Raphson iterative solution method presented earlier, and using the following close initial guess:

$$\left[\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4 \quad \theta_5 \quad \theta_6 \quad \theta_7\right]=\left[15^\circ \quad 15^\circ \quad 25^\circ \quad 55^\circ \quad 55^\circ \quad 65^\circ \quad 75^\circ\right]$$

the solutions are given in the table below (the units are degrees).

Solution	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7
only one	12.5	18.4	24.5	40.5	57.4	61.5	68.5

Note the Newton-Raphson method yields only one solution set for joint angles, generally the one closest to the initial guess. The above solution was obtained in 4 iterations to a tolerance of $\varepsilon = 0.001$. This solution was not expected since it does not agree with the known FPK input for this example pose. However, for kinematically-redundant robots such as the 7-dof Baxter arm, there are infinite possible solution sets. The solution reported above was used as the input to the FPK program and shown to be valid since it gave the correct $\left[{}^{Wo}_{GL}T\right]$. Note no optimization was attempted in this solution.

7-dof Left Arm Resolved-Rate Example

Starting from the neutral configuration for the Baxter Left Arm:

$$\begin{bmatrix}\theta_1 & \theta_2 & \theta_3 & \theta_4 & \theta_5 & \theta_6 & \theta_7\end{bmatrix} = \begin{bmatrix}0 & -31^\circ & 0 & 43^\circ & 0 & 72^\circ & 0\end{bmatrix}$$

the following Cartesian velocities and wrench are applied at {7}, in the basis frame of {0}:

$${}^0\{{}^0\dot{X}_7\} = \begin{Bmatrix}{}^0\{{}^0V_N\}\\ {}^0\{{}^0\omega_N\}\end{Bmatrix} = \begin{Bmatrix}-0.060\\ 0.050\\ 0.040\\ 0.3\\ 0.1\\ 0.2\end{Bmatrix}$$

(m/sec and rad/sec)

$${}^0\{{}^0W_7\} = \begin{Bmatrix}\{F_7\}\\ \{M_7\}\end{Bmatrix} = \begin{Bmatrix}1\\ 2\\ 3\\ 0.4\\ 0.5\\ 0.6\end{Bmatrix}$$

(N and Nm)

Assuming the input velocity and wrench are constant, the resolved-rate simulation in MATLAB yielded the following plots, running from 0 to 3 sec, using time steps of 0.05 sec. No optimization was attempted, i.e. this example is for the particular solution only.

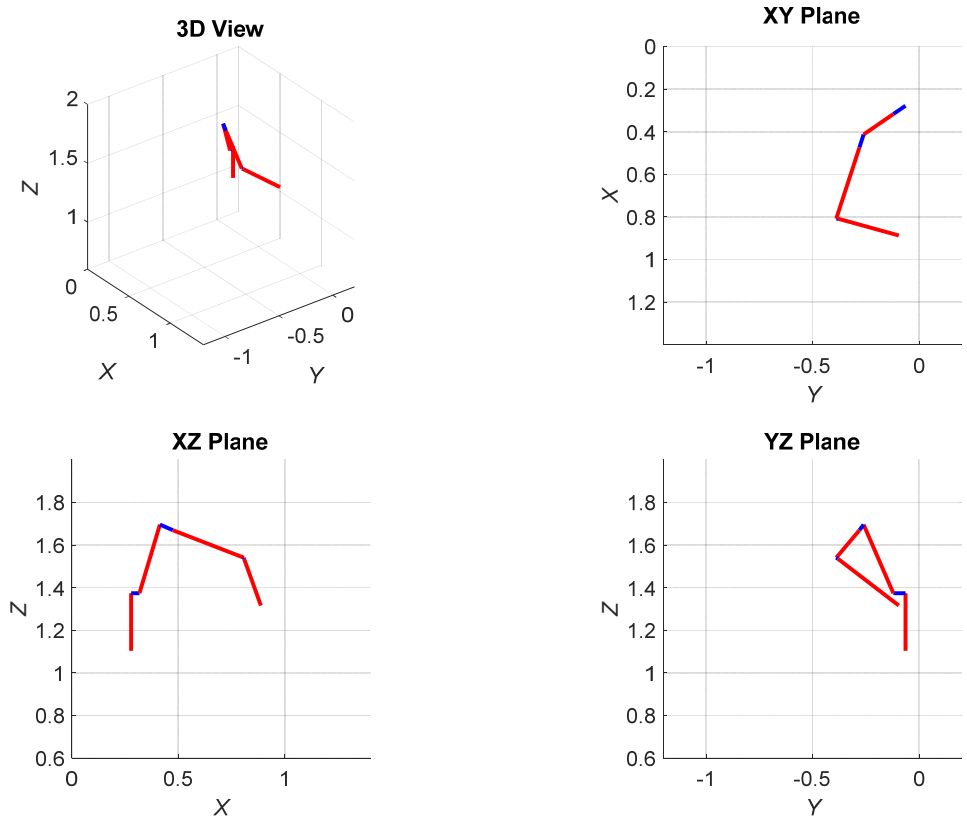


Figure 15a. Resolved-Rate Simulation, Final Left-Arm Configuration

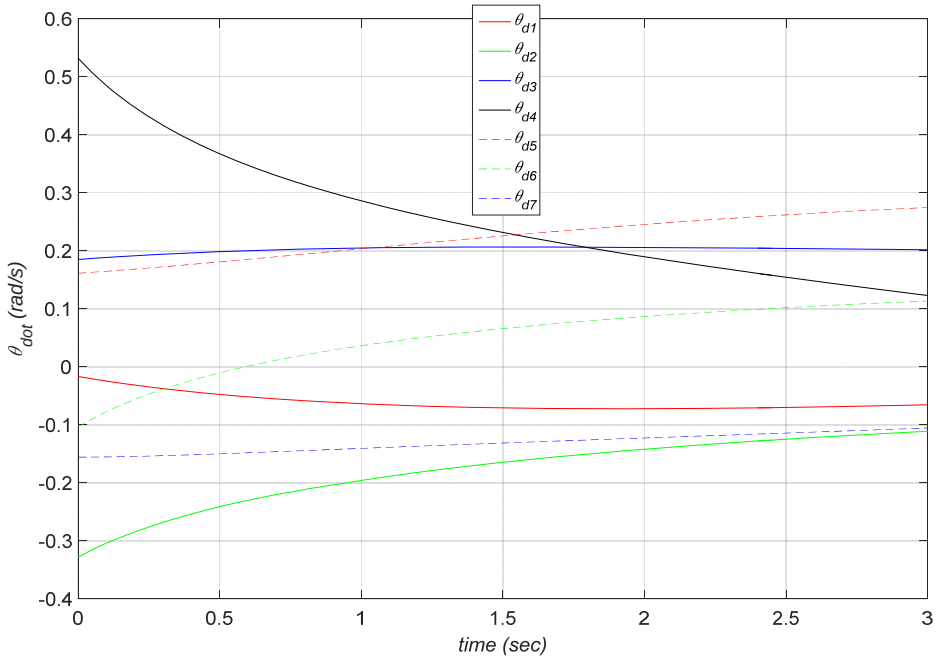


Figure 15b. Resolved-Rate Simulation, Joint Rates

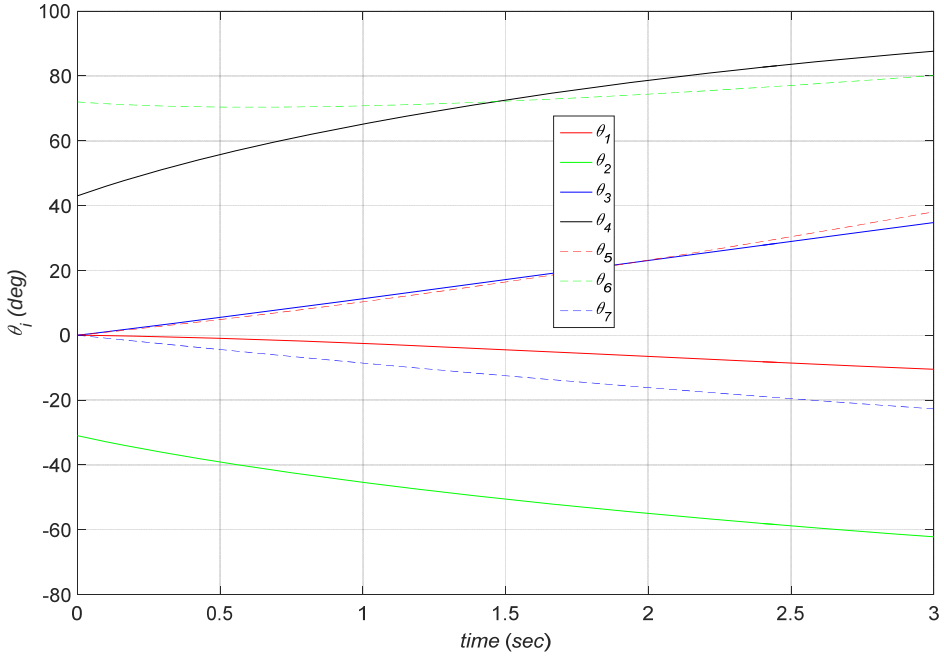


Figure 15c. Resolved-Rate Simulation, Joint Angles

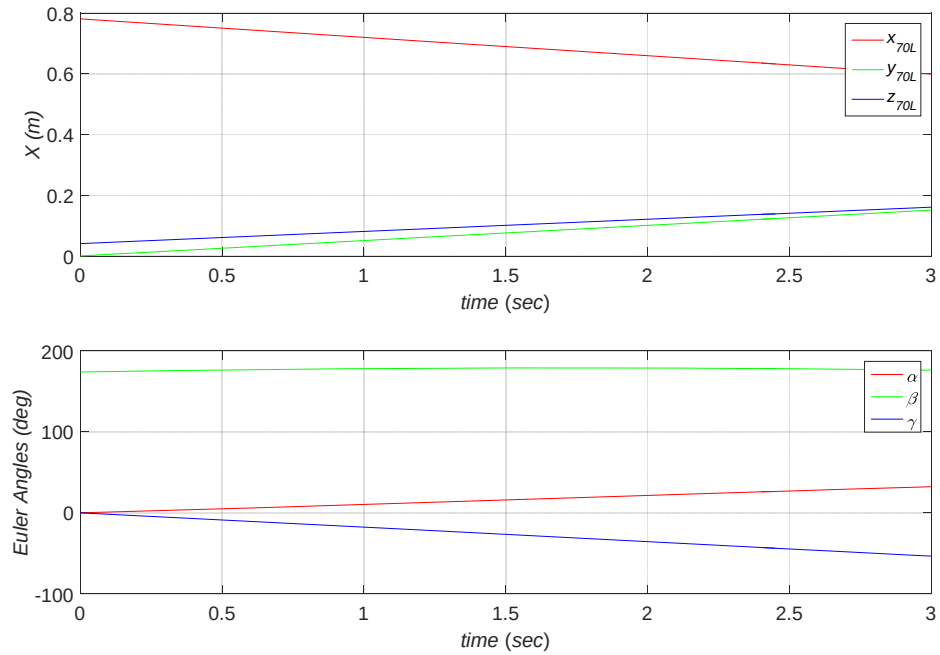


Figure 15d. Resolved-Rate Simulation, Cartesian Displacements

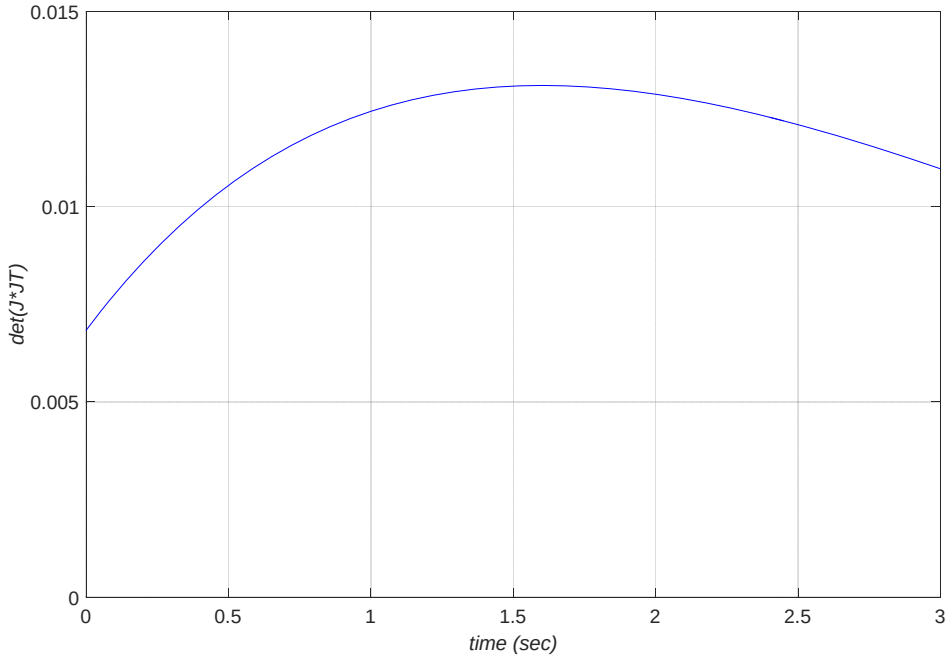


Figure 15e. Resolved-Rate Simulation, $\left|[J][J]^T\right|$ Determinant

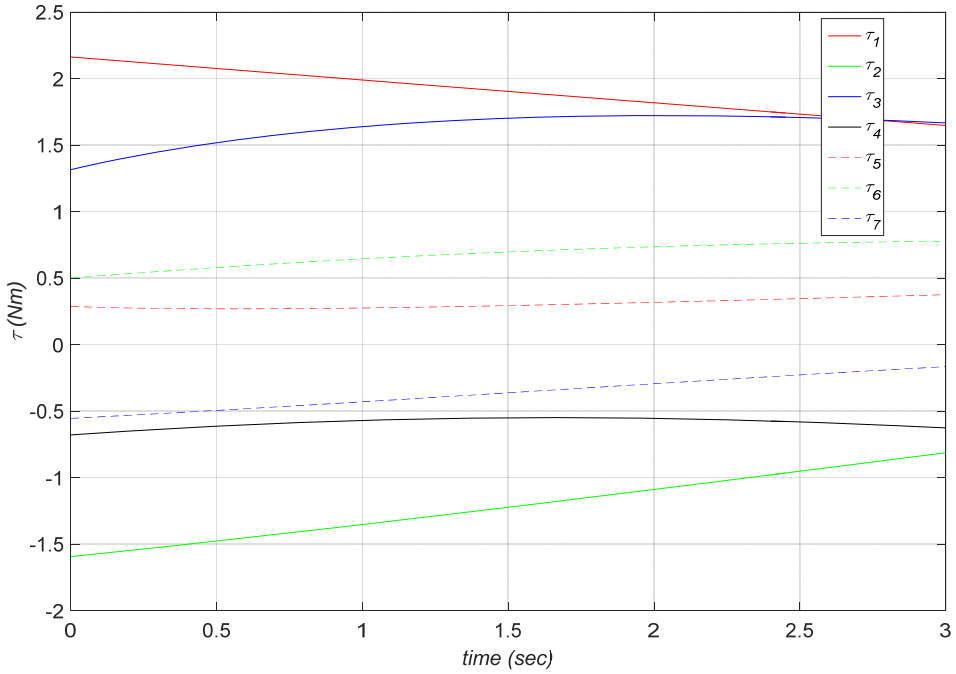


Figure 15f. Resolved-Rate Simulation, Pseudostatic Joint Torques

The first figure shows the Baxter left arm configuration after the simulated motion; the initial configuration was shown earlier, in the 'neutral' configuration snapshot example. The next plots show the required joint rates to achieve the commanded Cartesian rates, followed by the associate joint angles, integrated from the joint rates. The next plots show the 6-dof Cartesian displacements, calculated via the FPK solution; these demonstrate correct motion in the resolved-rate simulation since the constant slope of each Cartesian variable plot is the given constant Cartesian velocity. The next plot shows this simulation is not near singularities, since $\left|[J][J]^T\right|$ never crosses through zero. The final plot shows the required joint torques to exert the given constant Cartesian wrench, ignoring dynamics, at frame {7} onto the environment.

7-dof Left Arm Resolved-Rate Self-Motion Example

Starting from the neutral configuration for the Baxter Left Arm:

$$\begin{bmatrix}\theta_1 & \theta_2 & \theta_3 & \theta_4 & \theta_5 & \theta_6 & \theta_7\end{bmatrix} = \begin{bmatrix}0 & -31^\circ & 0 & 43^\circ & 0 & 72^\circ & 0\end{bmatrix}$$

zero Cartesian velocities were commanded at {7}, in the basis frame of {0}:

$$^0\left\{^0\dot{X}_7\right\} = \begin{Bmatrix} ^0\left\{^0V_N\right\} \\ ^0\left\{^0\omega_N\right\} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

(m/sec and rad/sec)

The purpose of this example is to exercise the homogeneous solution, to demonstrate self-motion (i.e. joint angle motion with zero Cartesian motion at {7}). Using a homogeneous constant $k_H = 2$, running from 0 to 3 sec, using time steps of 0.05 sec, the resolved-rate simulation in MATLAB yielded the following plots. Again, no optimization was attempted, i.e. this example is for simply demonstrating self-motion via the homogeneous solution only.

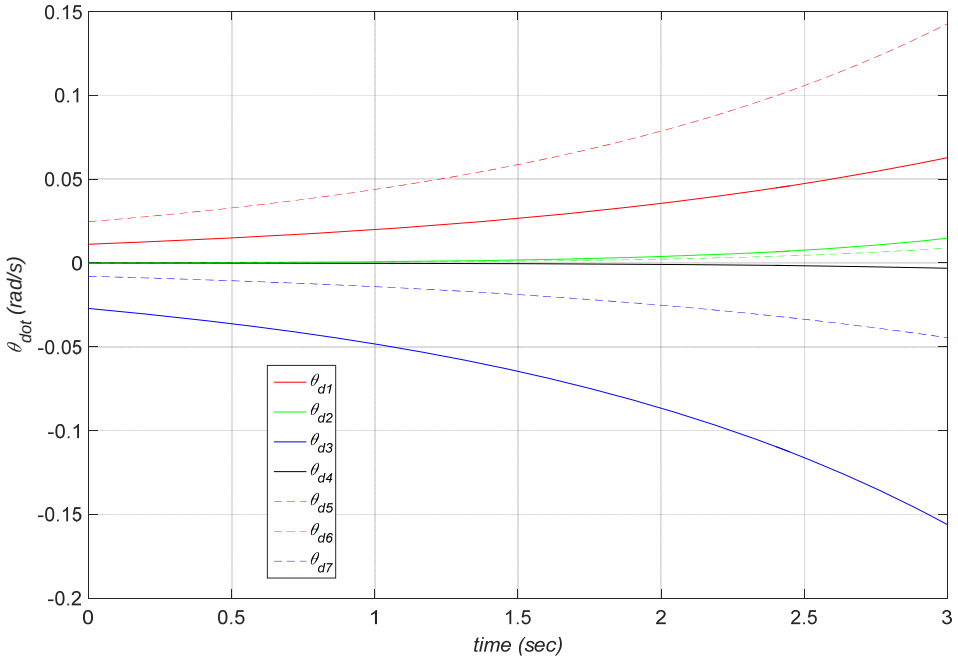


Figure 16a. Self-Motion Simulation, Joint Rates

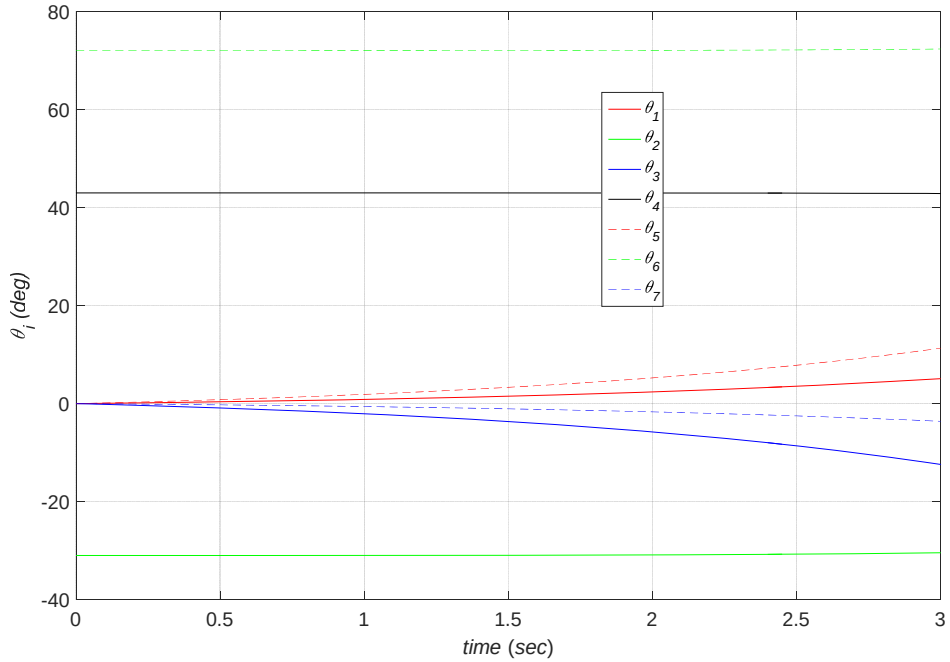


Figure 16b. Self-Motion Simulation, Joint Angles

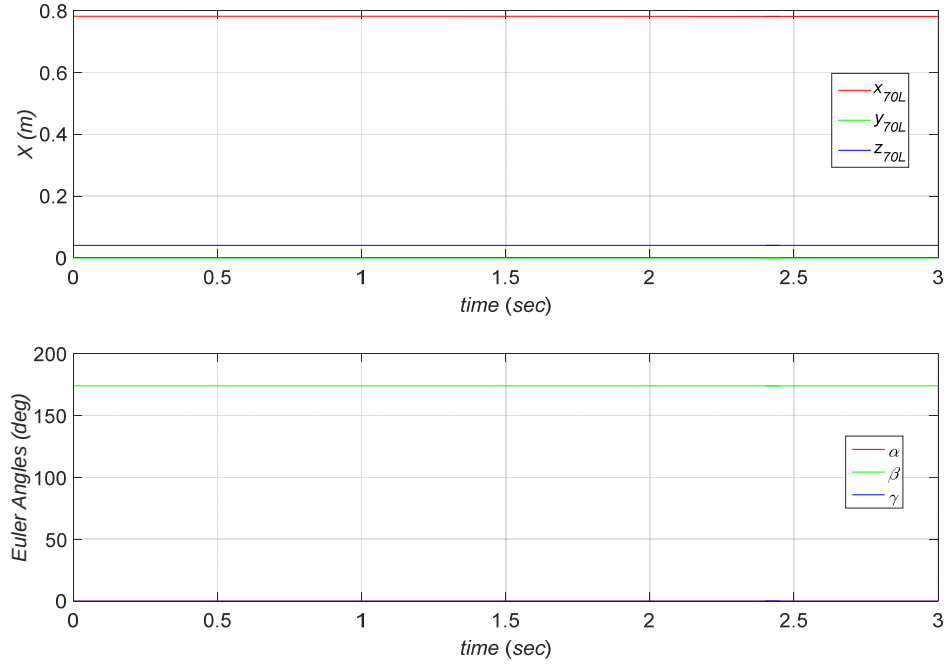


Figure 16c. Self-Motion Simulation, Cartesian Displacements

The first figure shows the required joint rates to achieve the commanded zero Cartesian rates, followed by the associate joint angles, integrated from the joint rates. The last plots show the 6-dof Cartesian displacements, calculated via the FPK solution. These demonstrate correct self-motion in the resolved-rate simulation since each Cartesian variable is constant (with constant slopes, i.e. the no motion zero rates commanded).

9. Conclusion

This paper has presented detailed kinematic analysis for the two 7-dof Baxter Humanoid Robot arms. The Craig (modified) Denavit-Hartenberg Parameters for each serial chain, specific length parameters, and joint angle limits were given. The general 7-dof and reduced 6-dof (locking joint three to zero) forward pose kinematics FPK solutions were developed, with analytical results. The analytical inverse pose kinematics IPK solution was given, for the 6-dof case. A numerical approach was given for the general 7-dof IPK solution. The Jacobian matrix was presented, along with resolved-rate control simulations, and limited singularity analysis. MATLAB Examples were given for all kinematics developments.

References

J.J. Craig, 2005, Introduction to Robotics: Mechanics and Control, Third Edition, Pearson Prentice Hall, Upper Saddle River, NJ.

S. Cremer, L. Mastromoro, and D.O. Popa, 2016, “On the Performance of the Baxter Research Robot”, IEEE International Symposium on Assembly and Manufacturing (ISAM), August.

J. Denavit and R.S. Hartenberg, 1955, A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices, Journal of Applied Mechanics: 215-221.

Z. Ju, C. Yang, Z. Li, L. Cheng, and H. Ma, 2014, “Teleoperation of Humanoid Baxter Robot using Haptic Feedback”, International Conference on Multisensor Fusion and Information Integration for Intelligent Systems (MFI 2014), Beijing China, September.

D.L. Pieper, 1968, “The Kinematics of Manipulators Under Computer Control”, PhD thesis, Stanford University, Department of Mechanical Engineering.

Rethink Robotics, 2016, “Baxter Research Robot: Technical Specification Datasheet & Hardware Architecture Overview”, http://www.active8robots.com/wp-content/uploads/Baxter-Hardware-Specification-Architecture-Datasheet.pdf.

e Silva, T.M. Tennakoon, M. Marques, and A.M. Djuric, 2016, “Baxter Kinematic Modeling, Validation, and Reconfigurable Representation”, SAE Technical Paper 2016-01-0334.

D.E. Whitney, 1969, Resolved Motion Rate Control of Manipulators and Human Prostheses, IEEE Trans on Man-Machine Systems.

C. Yang, H. Ma, and M. Fu, 2016, “Robot Kinematics and Dynamics Modeling”, Chapter 2 in Advanced Technologies in Modern Robotic Applications, 27-48.