

Algorithm to Determine if a Final Configuration is Obtainable by the Baxter Robot

Austin Henthorne, Glenn Ellis

September 18, 2017

Abstract

The Baxter robot is a two armed robot in which each arm is a 7 degree of freedom manipulator. Each arm has an end effector capable of grasping and moving objects. In order to do so Baxter must be aware of the objects position and orientation as well as its own starting position and orientation. Each link has a set length and each joint has a limited range of motion for rotation. We are to design a method for deciding whether the robot can reach a given configuration using the left or right hand. This is done through means of analysis using inverse kinematics with dual quaternions. Given an initial configuration and final configuration we are to determine whether or not it is possible with the limitations of the Baxter robot. This analysis is executed through MATLAB functions written to solve the inverse kinematics problem and compare the results to the possible joint angles given by the speculations of the robot.

1 Introduction

The Baxter Robot is used in industry and academia setting. It is a highly versatile manipulator capable of executing multiple tasks using motion and visual sensors. The sensors allow the robot to keep track of itself as well as desired goal through feedback. Although the robot is highly versatile its work space is limited due to the rotation limitation of joints. We are to devise a method to determine whether or not a given final configuration is reachable. In real life the initial and final configuration will be interpreted through the sensors on the robot; for this project we give the robot a final configuration and hardcode the initial configuration and use the method for simple motion planning to create an algorithm to ensure if the final given configuration is obtainable.

2 Problem Statement

Steps:

- 1) Convert initial (hard coded) and final configurations (4x4 transformation matrices) into dual quaternions.
- 2) Perform dual quaternion interpolation.
- 3) Solve for the joint angles.
- 4) Execute direct kinematics with the previous joint angles.
- 5) Check if the obtained final configuration is close to enough to the users input.

6) If step 5 is satisfied, compare our obtained joint angles with the range from the baxter robots and see if the users position can be met.

3 Solution Approach

Our approach is adapted from the simple motion planning algorithm.

Transformation Matrix to Dual Quaternion:

Transformation Matrix(4x4) = $\begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}$; Where R is a 3x3 Rotation matrix and p is a 3x1 Translation vector

Let the dual quaternion be denoted as $A = \alpha + \epsilon\beta$

α is the unit quaternion representing roatation.

$\beta = \frac{1}{2}p\alpha$; noting that $p\alpha$ is a quaternion multiplication.

The dual quaternion corresponding the transformation matrix is $A = \alpha + \frac{\epsilon}{2}p\alpha$

Dual Quaternion Interpolation:

Let $A = \alpha + \frac{\epsilon}{2}p\alpha$ be the initial configuration in dual quaternion form.

Let $B = \gamma + \frac{\epsilon}{2}p\gamma$ be the initial configuration in dual quaternion form.

Interpolation Equation: $C(\tau) = A \times (A \times B)^\tau$

Note: All multiplications are dual quaternion multiplication as well as power of a dual quaternion.

Calculating Joint Angles:

Frome the interpolation: $C(\tau) = \alpha + \epsilon\beta$

$$J_1 = \begin{bmatrix} -\alpha_1 & \alpha_0 & -\alpha_3 & \alpha_2 \\ -\alpha_2 & \alpha_3 & \alpha_0 & -\alpha_1 \\ -\alpha_3 & -\alpha_2 & \alpha_1 & \alpha_0 \end{bmatrix}$$

$p = 2\beta\alpha^*$; p = translation vector; α^* =conjugate of α

$$\hat{p} = \begin{bmatrix} 0 & -p_3 & p_2 \\ p_3 & 0 & -p_1 \\ -p_2 & p_1 & 0 \end{bmatrix}$$

$$J_2 = \begin{bmatrix} I_{3x3} & 2\hat{p}J_1 \\ 0_{3x3} & 2J_1 \end{bmatrix}; \text{I = Identity matrix; } 0 = \text{Matrix of zeros; } J_2 = 6x7 \text{ matrix}$$

$J^s = \text{Spatial Jacobian}$

$$J^s = \begin{bmatrix} \xi_1 & \xi'_2 & \xi'_3 & \xi'_4 & \xi'_5 & \xi'_6 & \xi'_7 \end{bmatrix}$$

$$\xi = \begin{bmatrix} -\omega \times q \\ \omega \end{bmatrix}; \omega = \text{Axes description of rotation (3x1); } q = \text{Point on the axis of each joint (3x1)}$$

$$\xi' = Adg\xi$$

$$Adg = \begin{bmatrix} \hat{\omega} & \hat{q}\hat{\omega} \\ 0 & \hat{\omega} \end{bmatrix}; Adg = \text{Adjoint of transformation matrix}$$

$$\hat{\omega} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$$

$$\hat{q} = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix}$$

$$B = (J^s)^T (J^s (J^s)^T)^{-1} J_2$$

$$\gamma = \begin{bmatrix} p \\ \alpha \end{bmatrix}$$

The Joint Angles are calculated using the following equation:

$$\theta(\tau + h) = \beta B(\gamma(\tau + h) - \gamma(\tau)) + \theta(\tau)$$

We set $\tau=1$ because we are only concerned about the joint angles for the last configuration.

h = small time-step

β = step size parameter which we set to 0.1

Direct Kinematics:

We use the joint angles from the previous step, and the link lengths of the baxter robot to find the final 4x4 transformation matrix using direct kinematics. The initial configuration of both end effectors is with respect to the global from S which is in the middle of the robot, level to the shoulders joint.

Check:

We use the obtained final 4x4 configuration and compare it with the users final configuration input.

First we isolate the roation matrices and convert them into quaternions.

Then we use the following equation to compare them:

$$\phi_2(q_1, q_2) = \sqrt{2(1 - |q_1 \cdot q_2|)}$$

If the answer is in the range of $[0, \sqrt{2}]$ then the final configuration is close enough to the users input.

Compare Angles:

We then compare each of the 7 angles obtained to the respective angles on the baxter robot (seen in results section) to see if the computed angles fall into the range. If the angles fall into the range then the user specified configuration is reachable by the robot. If the angles do not fall into the range then the robot is not able to reach the users configuration.

4 Results

The results of the method are determined by the given configuration. Since each iteration of the method is determined on whether or not the final configuration is possible there will be different results each time. The joint angles are checked within the range of their limitations and the joint that fails the configuration is identified if the problem occurs. The Code is iterated twice once for each arm. Below is a chart (a) displaying the joint angle ranges that are capable by the baxter robot. To verify that our code runs correctly, we hard coded an input that is reachable by the robot and ran the master function. The function outputted that the baxter robot can reach the configuration with both end effectors, confirming that our program runs correctly.

Joints	Ranges (rad)
θ_1	1.047 to -2.147
θ_2	2.618 to -0.052
θ_3	2.094 to -1.571
θ_4	3.028 to -3.028
θ_5	0.890 to -2.461
θ_6	3.059 to -3.059
θ_7	3.059 to -3.059

a) The maximum and minimum joint angles of the baxter robot.

5 Conclusion

Implementing the method using MATLAB with a given desired final configuration of the Baxter robot we are successfully able to ensure whether or not the configuration is obtainable. The code returns whether or not the hands of the baxter robot can reach a specified final configuration. The code itself is very computationally indepth and takes approximately an hour to run. Using a higher processing computer or a higher level coding program such as C++ the execution time may be shortened and more practical for use. This method could also be adapted to carry out other functions such as path generation by adding a few other functions to the main code. Overall, this method for determining validity of final configuration is accurate and can be used in applications for the Baxter Robot.