•

•

Rochester Institute of Technology

Electrical and Microelectronics Engineering Department

Technical Assistants' Office

Rm 09-3271 – MABL

R I T

**EEEE 585/685– Principles of Robotics Lab**

Experiment #5: Introduction to Baxter

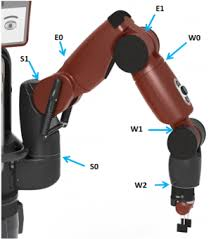
1. **OBJECTIVE:**

In this lab we will go through the basics of using a collaborative robot called Baxter. Utilize Robot Operating System. (ROS) to subscribe to Baxter’s joint states and publish joint values to move the arms. Understand the difference between Forward and Inverse Kinematics and Use an IK Solver to identify valid positions the arms can move to. And follow it up with a Pseudo Pick and Place.

1. **INTRODUCTION:** The Baxter Robot System is a human-sized humanoid robot with dual 7-degree-of-freedom (d-o-f) 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. 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. [1]



Figure 1 Front view of Baxter

****

­­­­­­­

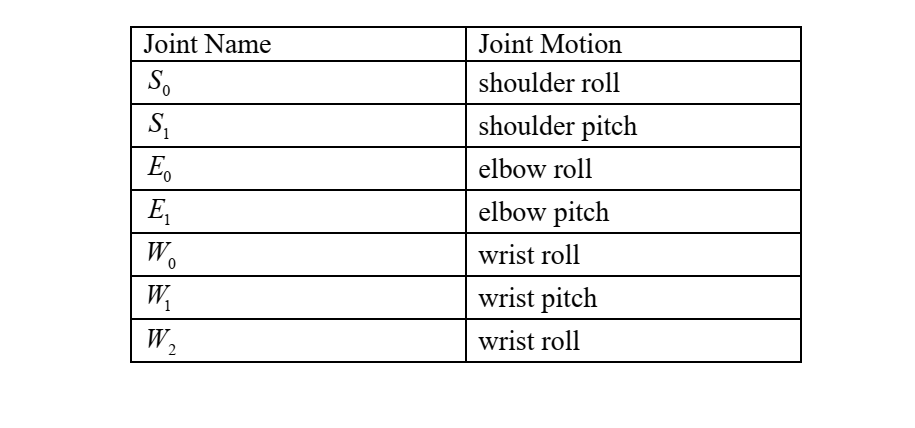


Figure 2 Baxter Joints

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, opensource 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 degrees, which

works out to 0.02197 degrees per encoder count. [1]

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

1. **LAB ASSIGNMENTS**

1.Launch Baxter in Gazebo and run a keyboard tele-op example and try moving the arms.

* + Source baxter\_ws
  + Run ./baxter.sh sim
  + Launch the appropriate ‘baxter\_world.launch’ file under baxter\_gazebo package

2. Write a subscriber to joint states and robot state.

* Use rostopic echo to find out correct topics to subscribe to

3. Write a publisher to move an arm of the robot.

* Write a publisher to the appropriate topic with a correct message to move the arm

4. Run the IK\_service\_client program to verify if the positions given to the robot are valid

* Rosrun ik\_service\_client.py under baxter\_examples package to find out valid movable locations for Baxter;s arm

5. Use Valid positions to Create your own Pseudo-Pick-and-Place routine for Baxter.

* Create a pick and place routine by providing joint angles. Use skeleton python script provided.

6. Perform a Pick and Place routine using IK (example provided in baxter\_simulator package)

* Make the necessary changes to run the routine on the real robot.

1. **REPORT**

Your report must contain the following:

* RQT plot of Baxter’s Nodes
* Flow chart of the implemented pseudo pick and place program
* Differences between Forward and Inverse Kinematics