The callback function accepts as input a buffer sequence (which stores all the data to be analyzed in the previous few seconds) and we update the buffer sequence every second to issue the next command to the robot. Since the callback function is in a `while(true)` loop after each group of 250 samples is collected (the sample rate of the EEG device is 250 Hz), we run our analysis every second on the buffer, which we define to be 1250 samples, or the past 5 seconds. This analysis includes the filtering and classification tasks discussed in the previous section. In order to communicate between the robot and the stream, we have a global variable `ALPHA_WAVE_VALUE` that we update in the callback function that is also accessed by the robot control function.

2.6 Robot Control

We use the WAM robot, which is set up as a 4-dof RRRR-bot without orientation control, as shown in Figure 4.

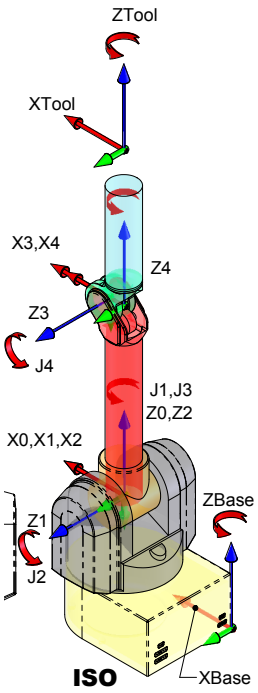

Figure 4: In the WAM, there is a base revolute joint (along $Z_{\text{Base}}, Z_0, Z_2$) which exerts little work and thus requires minimal torque to rotate, the revolute "heavy-lifter" (along Z_1) joint that is used to lift the arm for various tasks, another revolute joint (along $Z_{\text{Base}}, Z_0, Z_2$) at the same position as the heavy lifter to twist the arm, and finally, a revolute joint (along Z_3) to move the end effector around.

2.6.1 Task 1: Alpha Wave Control

Moving onto our first robot control task, we started out with a few simple tasks in simulation for debugging using `EEGWAMSimulator`. The simple tasks include moving in an ellipse or cycling through a list of positions, implemented using a position control law that calculates the forces to move joints along our defined trajectory.