ME 133a Final Project

Avi Patel
Kemal Pulungan

December 8, 2023

Introduction

For our final project, we made Atlas act as a chef. Specifically, it does the task of cutting with a knife with its right arm and moves its left arm to reach and grab something from a shelf around it. This involves dealing with singularities if the arm has to move to a desired position in task space outside of its range if the back remains straight.

Atlas is a humanoid robot developed by Boston Dynamics with 30 degrees of freedom. The workspace for Atlas is defined as the reachable region in the task space. Within our scope, we only consider the joint movements of the left and right arms and the torso.

Implementation

The goal with implementation is to evaluate the performance of Atlas cutting with the right hand and reaching with the left. We approached this problem in a few ways. Firstly, we used one primary task to define the cutting motion with the right arm and a secondary task for the left arm to reach. Additionally, we implemented two primary tasks with the right hand cutting and the left hand reaching away from the robot.

With the right hand's knife-cutting motion, the right hand moves in a line back and forth between (0.6, -0.33, 1.13) and (0.8, -0.33, 1.13), relative to world:[1]

$$p_d(t) = \begin{bmatrix} 0.1 \cdot \sin(3t) + 0.6 \\ -0.33 \\ 1.13 \end{bmatrix} \tag{1}$$

$$v_d(t) = \begin{bmatrix} 0.3 \cdot \cos(3t) \\ 0 \\ 0 \end{bmatrix} \tag{2}$$

For the left arm, we use the generalized path variable to describe the periodic motion of reaching to grab an item from a shelf and bringing it back to the cutting board,

$$s_p = \sin(0.2t) \tag{3}$$

$$\dot{s}_p = 0.2\cos(0.2t) \tag{4}$$

The desired position and velocity trajectories are then defined as,

$$p_d(t) = \frac{1}{2}(p_{0,\text{left}} + p_{f,\text{left}}) + \frac{1}{2}(p_{f,\text{left}} - p_{0,\text{left}}) \cdot s_p \tag{5}$$

$$v_d(t) = \frac{1}{2}(p_{f,\text{left}} - p_{0,\text{left}}) \cdot \dot{s}_p \tag{6}$$

[1] In code, link coordinates are expressed relative to pelvis.

where $p_{0,\text{left}}$ is positioned at the cutting board in front of Atlas as shown in Figure 4a, and pf, left is defined at a position either to the left of or behind Atlas.

The left hand moves between the point in front of it that stands in for the kitchen table at (0.76, 0.228, 1.29) relative to world, and another point around it that stands in for a shelf. There are two targets that the left hand moves to, one where the target is close enough to the robot that it doesn't need to stretch out its left arm at (0.113, 0.68, 1.59), and another where the target is too far to reach by standing upright and reaching at (-0.8777, 0.62, 1.29) relative to world. By treating both targets with different implementations, we see different results in whether the robot can reach the targets or not.

In images of Atlas, the GREEN ball is the desired position of the RIGHT hand, and the RED ball is the desired position of the LEFT hand. Furthermore, all hands' positions in this report are declared relative to the world, but in code are declared relative to the pelvis.

Right Hand as Primary Task, Left Hand as Secondary

In this case, the right hand's motion as the cutting hand is the primary task with $\lambda_{\text{primary}} = 20$. The left hand's reaching motion is the secondary motion, with $\lambda_{\text{secondary}} = 20$. In the simple case, Atlas's left arm goes from reaching in front of and above its left side, and then back to directly in front of Atlas at the table. This behavior can be seen in Figures 1a and 1b.

In the singularity case, Atlas's left arm reaches behind itself, and the point is beyond the reach of the arm if Atlas's back stays straight. Since the primary task is to cut with the right arm, the secondary task of reaching behind itself goes to the wayside, and the left arm doesn't reach the target. This can be seen in Figure 1c, where the red ball is the target, which the left hand is fairly far away from reaching.

We apply the generalized inverse kinematics equation with secondary motion given by,

$$\dot{\mathbf{q}} = \text{diag}(\dot{q}_{\text{max}})[J\text{diag}(\dot{q}_{\text{max}})]^{\dagger}[\dot{\mathbf{x}}_r - \lambda\mathbf{e}] + (I_{17\text{x}17} - [J\text{diag}(\dot{q}_{\text{max}})]^{\dagger}[J\text{diag}(\dot{q}_{\text{max}})])\dot{q}_{\text{sec}} \quad (7)$$

where $\text{diag}(\dot{q}_{\text{max}}) = \text{diag}(0.1, 0.1, 0.1, 1, ..., 1)$ is a 17x17 diagonal matrix. We scaled the joint velocities this way to slow down the motion of our back joints (`back_bkz`, `back_bky`, `back_bkx`) so the robot would not prioritize those joints when carrying out arm tasks. J is the 6x17 Jacobian of all 17 joints. J_{sec} is the Jacobian specifically for the left arm, and gamma-weighting is applied to it to deal with singularities.

$$J_{\text{Wsec}} = J_{\text{sec}}\text{diag}(\dot{q}_{\text{max}}) \quad (8)$$

$$J^{\dagger}_{\text{Wsec}} = J^{\text{T}}_{\text{Wsec}}(J_{\text{Wsec}}J^{\text{T}}_{\text{Wsec}} + \gamma^2 I_{6\text{x}6})^{\dagger} \quad (9)$$

where $\gamma = 0.5$. J_{Wsec} goes into

$$\dot{q}_{\text{sec}} = \lambda(\text{diag}(\dot{q}_{\text{max}})J^{\dagger}_{\text{Wsec}})\mathbf{e}_{\text{sec}} \quad (10)$$

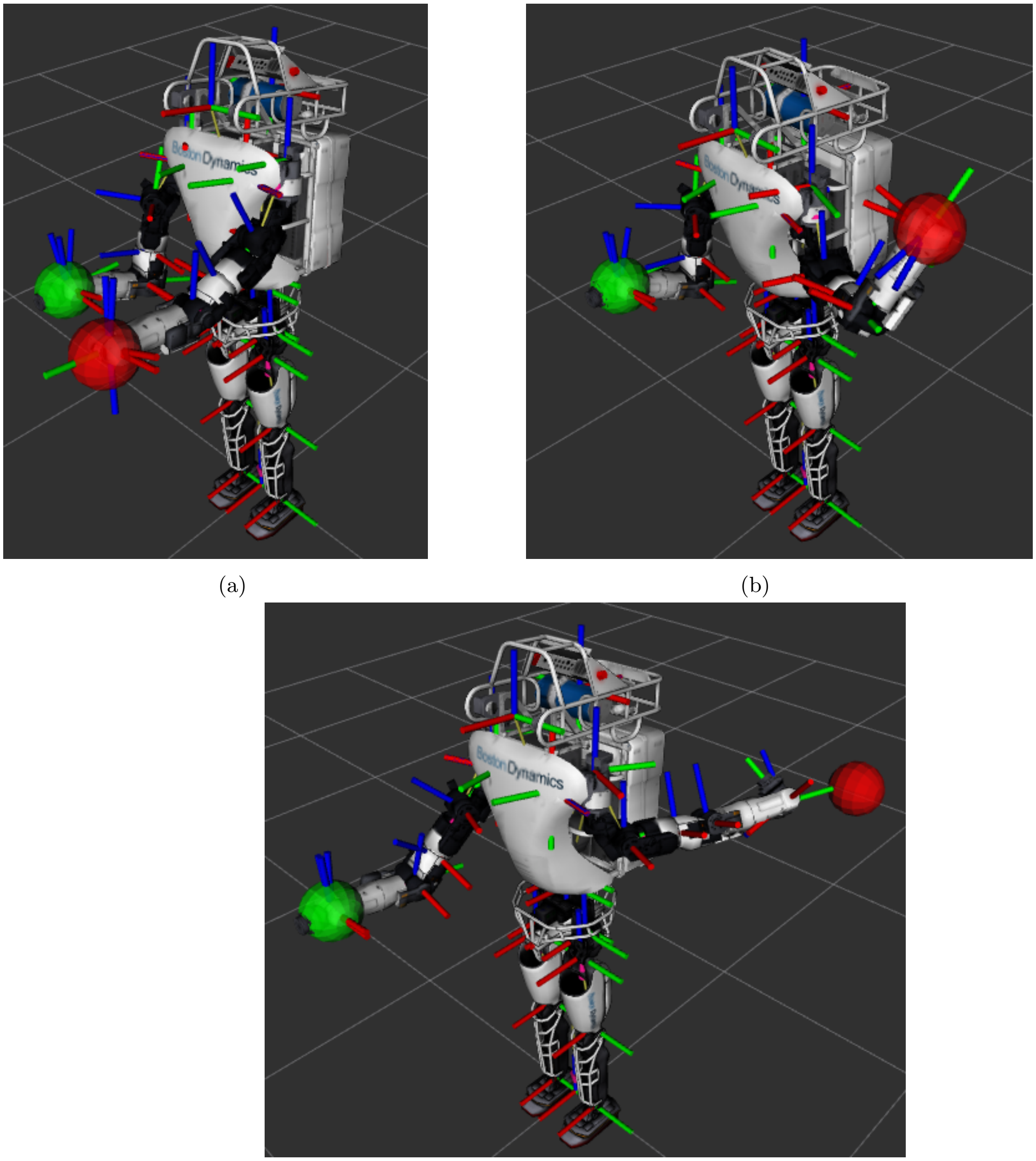
In the graphs of joint positions and velocities, most of the right arm joints followed sinusoidal paths, while the back and left arm joints followed paths that were comprised of more sinusoidal components, as seen in Figure 2. When the arm is at singularity, some of the left arm joints max out their positions, as well as move fast at a specific time, a result of the singulariry, as seen in Figure 3.

Right and Left Hands as Primary Tasks

The primary task of the right hand is to perform the cutting and the left hand is to reach and grab items on a neighboring shelf. We therefore defined two kinematic chains from `pelvis` to `r_hand` and `pelvis` to `l_hand` for Atlas. The trajectory of the cutting motion for right-hand primary task in task space is given by Equations 1 and 2 and the trajectory of the left arm motion to reach and grab items from a nearby shelf is given by Equations 5 and 6. We apply the generalized inverse kinematics to solve for the joint angles using,

$$\dot{\mathbf{q}} = \text{diag}(\dot{q}_{\text{max}})[J\text{diag}(\dot{q}_{\text{max}})]^{\dagger}[\dot{\mathbf{x}}_r - \lambda\mathbf{e}] \quad (11)$$

where J is the 6x17 Jacobian of all 17 joints, $\text{diag}(\dot{q}_{\text{max}}) = \text{diag}(0.1, 0.1, 0.1, 1, ..., 1)$ is a 17x17 diagonal matrix. We scaled the joint velocities this way to slow down the motion of our back joints (`back_bkz`,



(a) (b)

(c)

Figure 1: Positions of Atlas for the right hand as the primary task and left hand as secondary task: *a:* Initial position of the left arm. *b:* Final position of the left arm in simple reach mode. *c:* Final position of the left arm in singularity reach mode.

`back_bky`, `back_bkx`) so the robot would not prioritize those joints when carrying out arm tasks. Our Jacobian J for both left and right hand primary tasks is the 12x17 matrix,

$$J = \begin{bmatrix} \text{Right Torso} & \text{Right Arm} & 0 \\ \text{Left Torso} & 0 & \text{Left Arm} \end{bmatrix} \tag{12}$$

where Right Torso and Left Torso are each 6x3 matrices corresponding to the back joints of Atlas, and right and left arm are each 6x7 matrices corresponding to the joints in the right and left arms. The zeros represent in the above equation represent 6x7 matrices of zeros.

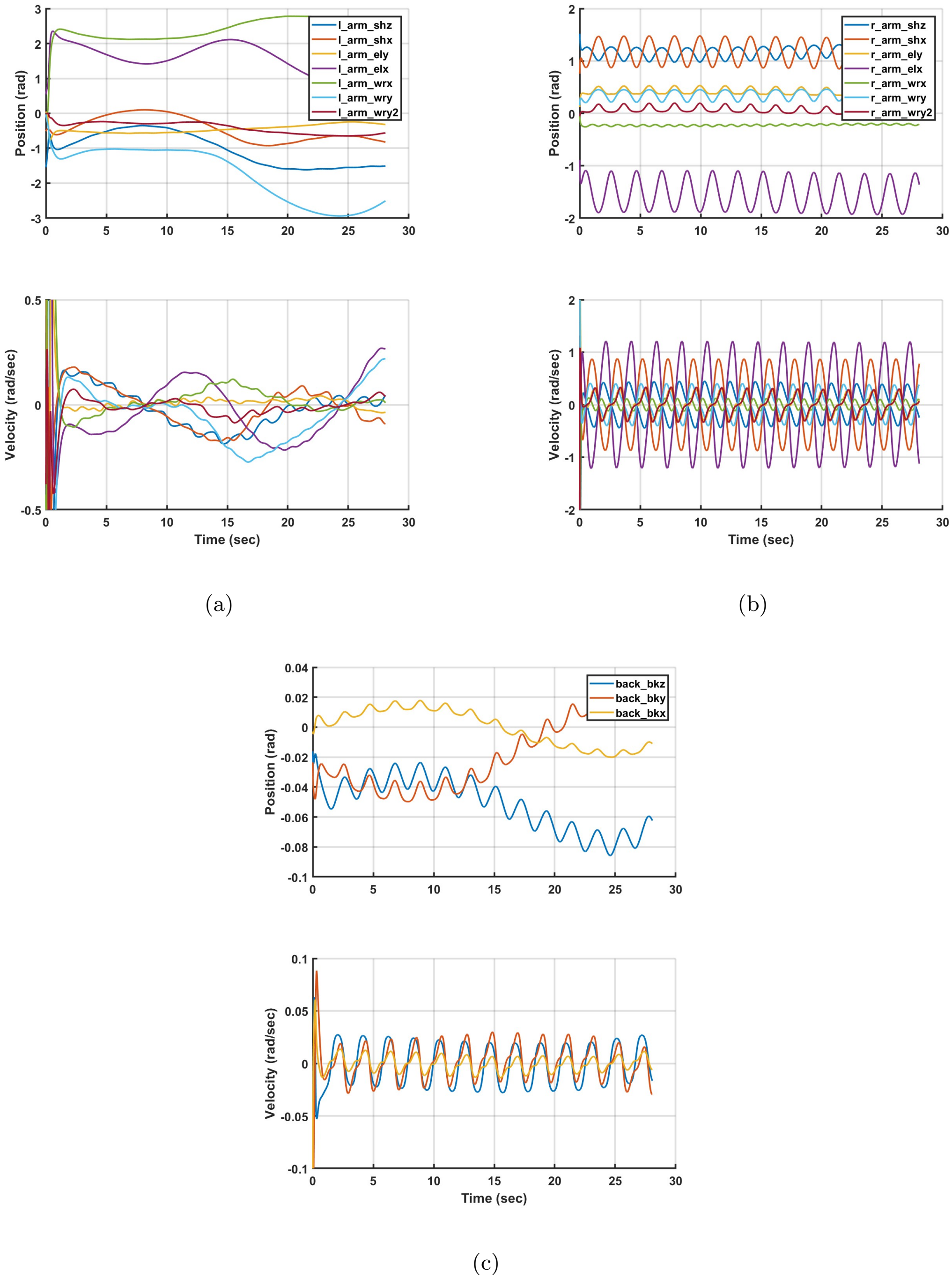
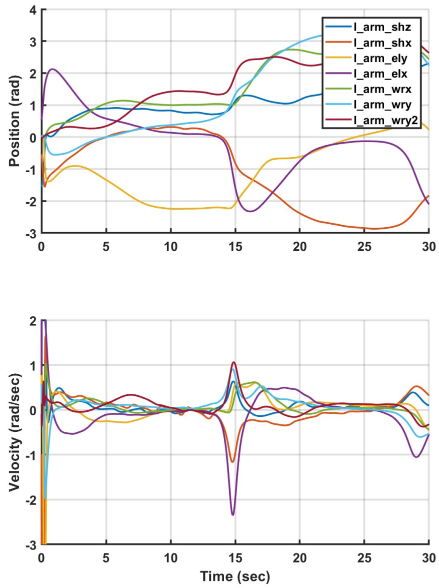
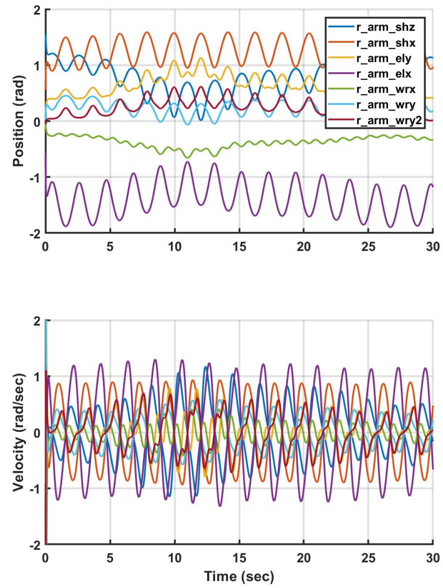


Figure 2: Joint positions and velocities for simple trajectory of right hand as primary task and left hand as secondary task. *(a)* Left arm joint positions and velocities. *(b)* Right arm joint positions and velocities. *(c)* Back joint positions and velocities.

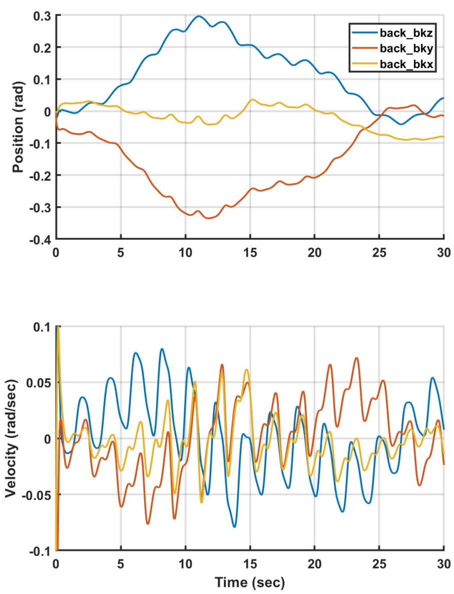
The results of this method can be seen in Figure 4. When the target is close by, the left arm acts much the same as when it acted as the secondary motion. However, when the target is further, as seen in 4c, Atlas will move its back so the left arm can get as close as possible, and continue doing its primary motion of cutting with the right hand.



(a)



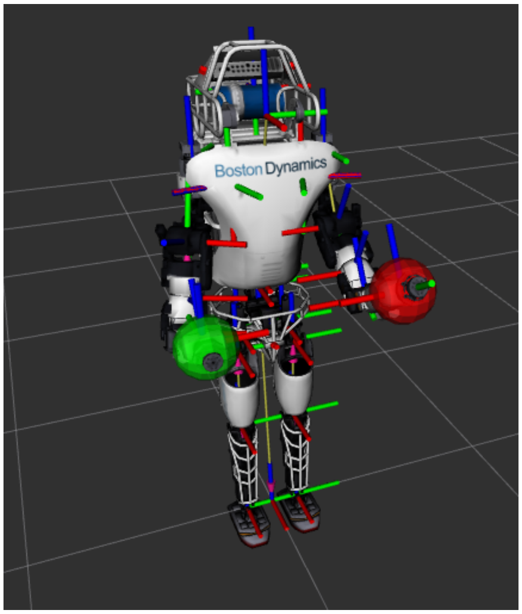
(b)



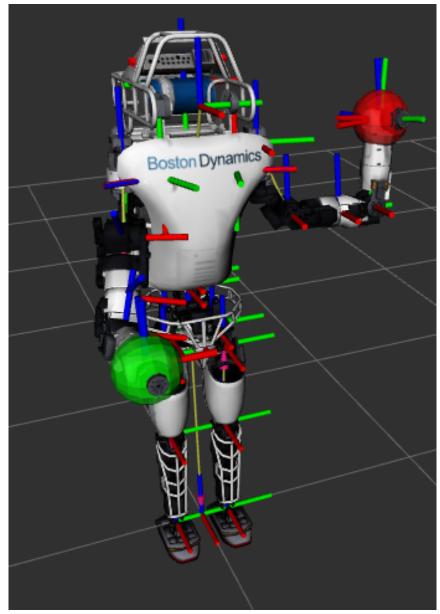
(c)

Figure 3: Joint positions and velocities for trajectory reaching singularity of right hand as primary task and left hand as secondary task. *(a)* Left arm joint positions and velocities. *(b)* Right arm joint positions and velocities. *(c)* Back joint positions and velocities.

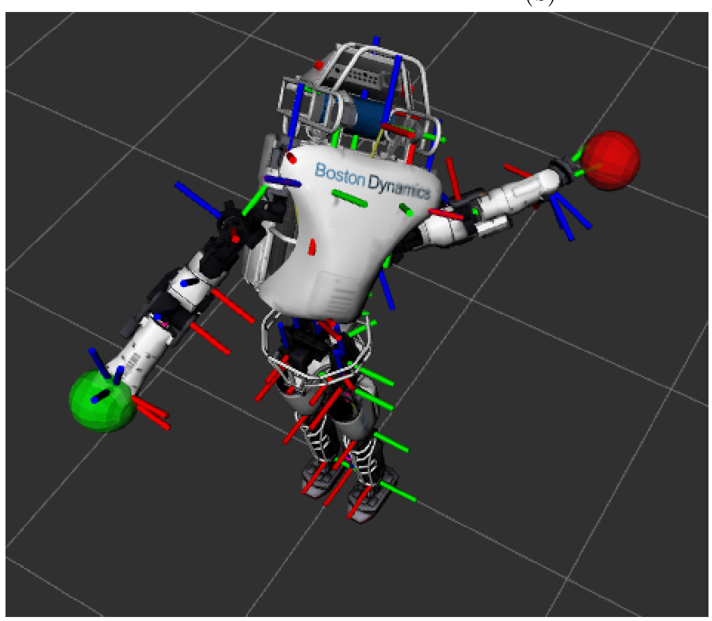
Again, in the graphs of joint positions and velocities, most of the right arm joints followed sinusoidal paths, while the back and left arm joints followed paths that were comprised of more sinusoidal components, as seen in Figure 5. When the arm is at singularity, some of the left arm joints max out their positions, as well as move fast at a specific time, a result of the singularity, as seen in Figure 6. Compared to that with



(a)



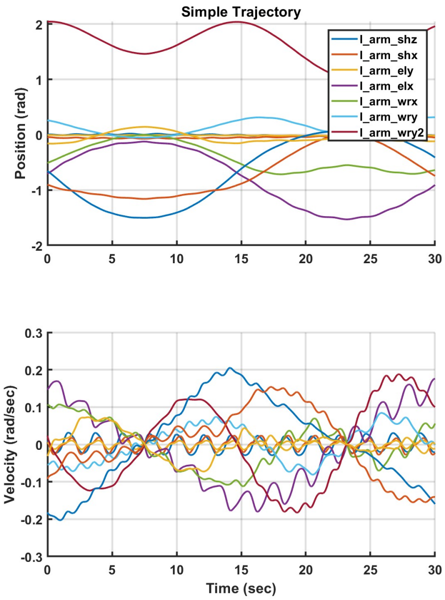
(b)



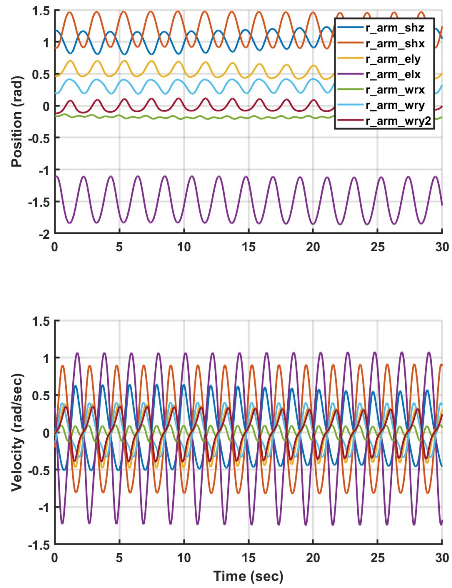
(c)

Figure 4: Atlas when the two arms are primary motions. *a:* Initial position of the left arm. *b:* Final position of the left arm in simple reach. *c:* Final position of the left arm in singularity reach.

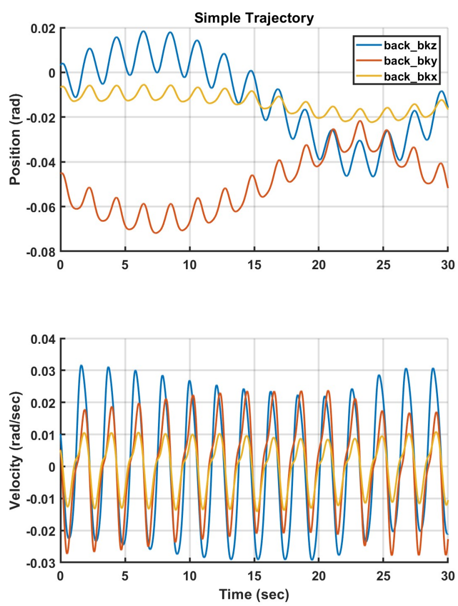
the left hand as a primary task, the back moves more in this case when the left arm is in singularity, which shows that the robot has prioritized using these joints to reach further away because that task is a primary task.



(a)

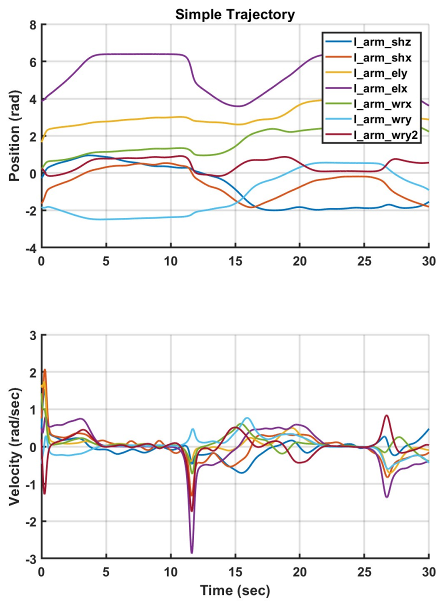


(b)

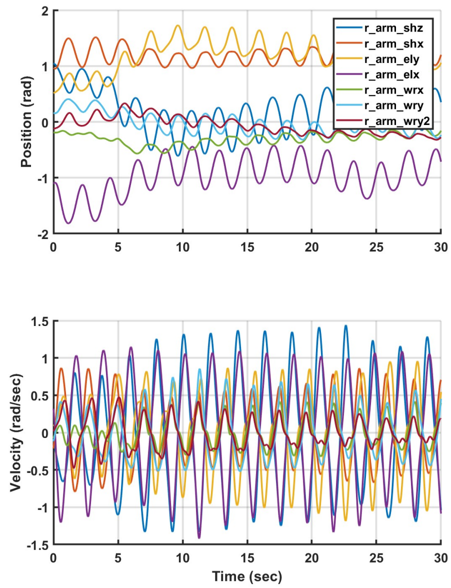


(c)

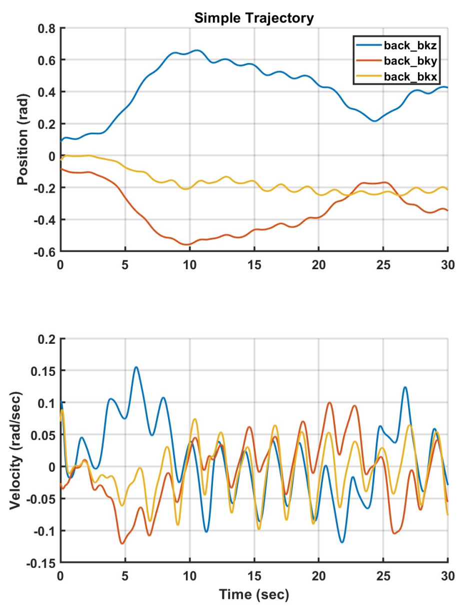
Figure 5: Joint positions and velocities for simple trajectory of right hand as primary task and left hand as secondary task. *(a)* Left arm joint positions and velocities. *(b)* Right arm joint positions and velocities. *(c)* Back joint positions and velocities.



(a)



(b)



(c)

Figure 6: Joint positions and velocities for trajectory reaching singularity of right and left hands as primary tasks. *(a)* Left arm joint positions and velocities. *(b)* Right arm joint positions and velocities. *(c)* Back joint positions and velocities.