ME 133a Project Report: Atlas Does Pushups!

Nishka Arora, Shevali Kadakia, Krish Mehta

I. INTRODUCTION

After a term of learning about robotic arm manipulation, we thought it would be interesting to work with a large humanoid robot to apply what we've learned. We thought a pushup would be especially fun. A pushup is a full-body movement and requires a lot of coordination. Moreover, there are many variations of pushups (asymmetric pushups). We wanted to solve the inverse kinematics for a general pushup (you can define the position of the hands and feet). One of the main ideas we explored was escaping the singularity that arises when the arms and legs reach a straight position. We decided to use Atlas since the urdf was already provided to us, and it had a lot of degrees of freedom to accurately mimic a humanoid. We decided to use RViz due to its simplicity, and because we were not using forces in our model. While Atlas has 30 joints, excluding the neck, we used 29 joints. We added 6 extra degrees of freedom to move the robot relative to the world (explained later). The robot is controlled using inverse kinematics, with 5 primary tasks: one to keep each hand stationary at the target position, and the last one to induce a sinousodial verical motion at one point on the robot to do the actual pushup motion. There were a wide variety of secondary tasks that we used to help us do different variations of pushups.

II. DESCRIBING THE SYSTEM

While Atlas has 30 joints, excluding the neck, we used 29 joints (the neck was irrelevant to our intended movement). The robot is rooted at the pelvis, with 4 chains leading out to each of the 4 limbs (2 arms and 2 legs). We also decided to add 6 more joints (XYZ prismatic, XYZ rotation), which connected Atlas to the origin frame of RVIZ so that the entire robot could move in all 6 degrees of freedom in the 3D world space. This led to the joint space being 35 (29 Atlas + 6) joints, with 3 of them being prismatic (the 3 degrees controlling where the pelvis was in the world frame), and 32 revolute joints.

Here are some questions we answered to arrive at this system:

A. Why can't we fix Atlas' hand and feet to the ground without inverse kinematics?

Without inverse kinematics, fixing Atlas' hands and feet would lead to a closed-chain model. That is, we would have to specify 4 different origin points in the robot description. We abandoned this approach because:

- We didn't discuss closed-chain kinematics in this course.
- The URDF format doesn't allow multiple origin points. Another robot description format SDF (Simulation Description Format) does.

B. Why add 6 extra joints instead of controlling the relative position of Atlas using a broadcast?

The other option we considered was to broadcast the up-and-down movement of the pelvis and use inverse kinematics to ”fix” the position of the hands and feet to the ground. The problem with this arrangement is that the position of the hand and feet is defined as relative to the origin of the robot (the pelvis). So, if the pelvis position changes, the position that we want the hands and feet to be fixed to will also change. Again, while this is theoretically possible to implement, it complicates the matter.

So, finally, we decided to have the robot defined starting from the world frame. The 6 degrees of freedom from the origin of the world to the origin of the robot allow the robot to move efficiently in the 3D world space. Since the hand and feet position is now defined relative to the world frame, we can use inverse kinematics to fix their position to the ground.

III. DESCRIBING THE TASK

We define the pushup movement with 2 main characteristics: all 4 tips (2 hands and 2 feet) are stuck to the ground, and the chest moves up and down. So, each limb was tasked to be stationary relative to the world. This made it such that the robot always had all 4 limbs attached to the ground, which ensured that the robot was never “flying” (4 tips $\times$ 6 tasks each = 24 tasks). The 5th task was to move the chest up and down (1 tip $\times$ 6 tasks = 6 tasks). This gives us a total of $24 + 6 = 30$ tasks. Since there are 30 tasks and 35 joints, we expected redundancy in the movement, i.e. the robot was under-defined. We later add secondary tasks to make the movement look more realistic [See Section IV Part C].

IV. ALGORITHM AND IMPLEMENTATION

Trajectory Generation:

The motion of the robot is controlled only by generating a sinusoidal trajectory (in task space) that increases and decreases the z-axis coordinate of the point where we want to focus the movement which was the pelvis (later the chest) of the robot. Let the vector $\vec{x}$ be the position coordinate of the pelvis. Then we have:

$$\vec{x} = x_0 + \begin{bmatrix} 0 \\ 0 \\ \cos(A\frac{t\pi}{T}) \end{bmatrix} \tag{1}$$

$$\dot{x} = \begin{bmatrix} 0 \\ 0 \\ -A\frac{\pi}{T}\sin(\frac{t\pi}{T}) \end{bmatrix} \tag{2}$$

We use the joint state publisher to move Atlas into a starting position, we then measure the initial position and orientation of the hands and feet. There is no change with time for these values, so they remain constant. Similarly, the initial orientation of the pelvis/chest is also measured initially and tasked to remain constant. The corresponding velocities for these stationary tasks are 0.

A. *Pelvis Pushup*

Our first task was to set up the kinematic chain class to include the chains for all 5 tips.

- world $\rightarrow$ left hand
- world $\rightarrow$ right hand
- world $\rightarrow$ left foot
- world $\rightarrow$ right foot
- world $\rightarrow$ pelvis

The problem we faced with this was that there were overlapping joints in the 5 chains. The way we get around the overlapping joints is to generate Jacobians for all 5 chains and combine the Jacobians to create one big Jacobian with 35 columns for each joint and 30 rows for each task such that:

$$J_{(i,k)} = \left\{ \begin{array}{ll} \frac{\partial x_i}{\partial q_k}, & \text{if joint } q_k \in \text{chain for task } x_i \\ 0, & \text{otherwise} \end{array} \right\} \tag{3}$$

After the Jacobian was set up, the next task was to implement inverse kinematics. At this stage, we used a fairly standard form of inverse kinematics. We decided to use the pseudo-inverse in order to minimize joint velocity.

$$\dot{q} = J^{+}(\dot{x} + \lambda * \text{err}(x_{des}, x))$$

We then integrated it to find the updated joint position:

$$\vec{q} = \vec{q}_{prev} + dt * \dot{q}$$

We noticed the following in the motion of our robot:

- While the movement looked somewhat like a pushup, it didn't look quite right because the movement was centered at the pelvis. The robot was using its knee joints a lot more than its elbow joints, making the motion look unnatural and unlike a pushup. Moreover, the robot would bend at the pelvis in a concave and then a convex angled position. Indicating that our choice of trajectory needed fixing.
- Because the lower half of the robot was being used more, the knee would reach a singularity and glitch out often.

We added a weighted pseudoinverse with a gamma value of 0.05 to remove the singularity.

$$J_w^{+} = V\text{diag}(\frac{s_i^2}{s_i^2 + \gamma^2})U^T \tag{4}$$

B. *Chest Pushup*

The reason we started with the pelvis pushup in part A was that the Atlas URDF defines the origin of the robot (the root of its frames) at the pelvis. So, it was easiest (least overlap in joints) for us to use that frame as a starting point for where to define the up-and-down motion trajectory. However, a pushup motion is generally thought of as an up-and-down movement of the chest (using elbow movement), not as an up-and-down motion of the pelvis using knee movement.

Since there was no existing chest frame in the Atlas URDF, we defined our own chest frame and changed the trajectory to use the chest frame for the primary task instead of the pelvis. The resulting motion was a lot more realistic - the robot no longer bent at the pelvis in both directions.

C. *More Realistic Pushup*

Although the chest pushup looked like a conventional pushup, it was still using the legs and back to move, while at the same time its elbows were bending out of the range of the bending of the normal elbows. Since the only task information that the robot had was to move its chest, we decided to incorporate a secondary task, and minimize a weighted joint velocity, to prioritize motion in some joints over others.

Weighted Joint Velocity: Since a pushup is usually done by isolating the arms and shoulders and using those muscles to move the body, one of the best ways to motivate this motion would be to weigh different joints differently, when minimizing the joint velocities.

We know that the right inverse J^{-1} minimizes the norm of the joint velocities $||\dot{q}||^2 = \dot{q_1}^2 + \dot{q_2}^2 ... \dot{q_n}^2$. We can instead minimize the function $||M\dot{q}||^2 = \lambda_1^2\dot{q_1}^2 + \lambda_2^2\dot{q_2}^2 ... \lambda_n^2\dot{q_2}^2$, where M is a diagonal matrix, and $M_{ii} = \lambda_i$. To minimize this, we can instead minimize $\frac{1}{2}||Mq||$, subject to $\dot{x_r} = J\dot{q}$. Using a Lagrange multiplier:

$$\dot{q} = M^{-2}J^T(JM^{-2}J^T)^{-1}\dot{x_r} \tag{5}$$

Thus, we can use M; in particular, λ_i to weigh how expensive moving joint q_i would be.

Since the pushup would require the legs and the back to stay straight and not move, all the legs and back were weighted at 25, which means that it was favored for the robot to move its arms instead of its feet. Additionally, to incentivize physically moving the body downwards, the prismatic joints from the world to the body were weighed to 0.1, to encourage the body's movement in the pushup.

Centering Joints: Sometimes, the elbows would bend in the incorrect direction, so we decided to use a cost-function minimization method to push the joint values closer to their central position.

Assume that there is some joint i, and q_i is the current position of the joint. Let q_i^M be the maximum value that q_i can take, and q_i^m be the minimum value.

Since each joint could have a different minimum value, we decided to rescale the bounds of the joint from $[q_i^m, q_i^M]$ to $[-1, 1]$. Let this transform be $T(q)$.

$$T(q_i) = \frac{q_i - (q_i^M + q_i^m)/2}{(q_i^M - q_i^m)/2}$$

The cost function $C(q_i)$ was:

$$C(q_i) = T(q_i)^4$$

$$\nabla C = 4T(q_i)^3 \times T'(q_i) \tag{6}$$

And thus, $q_s = -\sum \lambda \nabla C(q_i)$, where $\nabla C(q_i)$ is 0 for all $(\nabla C(q_i))_j$, $j \neq i$. Then, we can project this into the null space as a secondary task, $\dot{q} = J_p^+ v_p + (I_n - J_p^+ J_p)\dot{q_s}$.

D. Asymmetric Pushup

Finally, we tried to apply our work so far to variations on the regular symmetrical pushups. We tried asymmetrical pushups we changed the initial x and y position of the hands to be asymmetrical. We also tried asymmetrical pushups with one hand higher up in the z axis than the other (mimicking a pushup with one hand on a table or book). Our primary reason for this experiment was to check how our implementation of a pushup behaves as the robot arms reach singularity from different positions.

Our implementation generalized well, and we didn't have to implement any major algorithmic changes. One change we did make was to remove the rows that correspond to the x and y position of the chest from the Jacobian matrix. By doing this we allowed for the sideways movement of the robot.

One reason for the robot behaving well even with asymmetrical hand pushups was that we kept the orientation of the chest of the robot in the initial position (parallel to the ground) with inverse kinematics. So, it was difficult for the robot arms to reach a singularity in one hand before the other.

V. Analysis

A. Do the hands and feet stay on the floor?

B. Does the chest move as expected?

1) *Symmetric - Pelvis Pushup:*
2) *Symmetric - Chest Pushup:*
3) *Weighted Joint Velocity, Centering:*
4) *Asymmetric - Z is different:*
5) *Asymmetric - XYZ is different:*

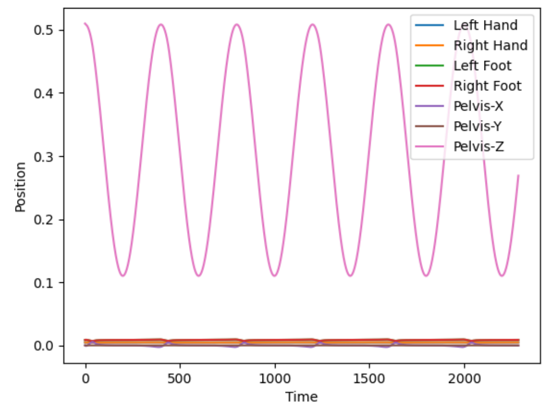

Fig. 1. Tip Positions for Pelvis Pushup. The hands and feet do a good job of staying on the ground (Z=0), and the pelvis moves freely up and down in the Z axis. The x and y position doesn't change by a lot.

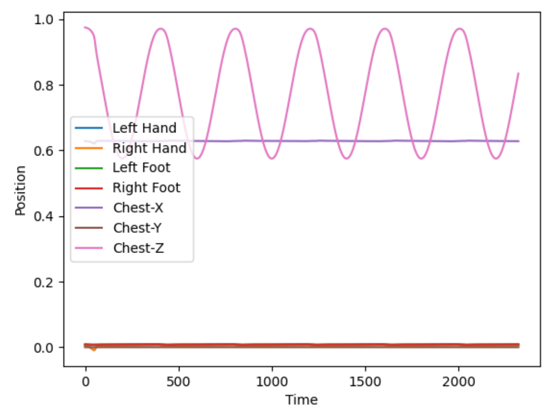

Fig. 2. Tip Positions for Chest Pushup. The chest pushup has a similar result. Notice that the x position of the chest is larger than the x position of the pelvis because the chest is higher up on the robot.

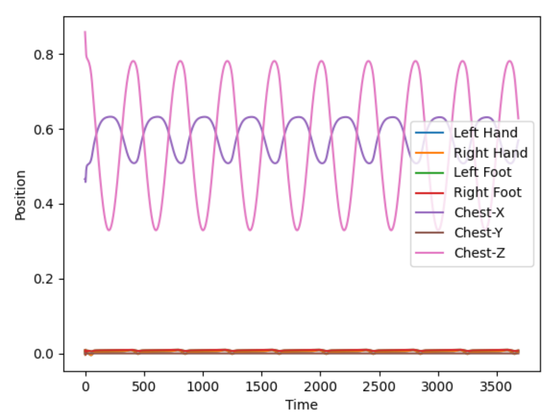

Fig. 3. Tip Positions for Pushup with Weighted Joint Velocity, Centering. Freeing up the x and y position of the chest from the Jacobian was helpful because we can see that there is sinusoidal movement in the x position of the robot which wasn't there in Fig 2.

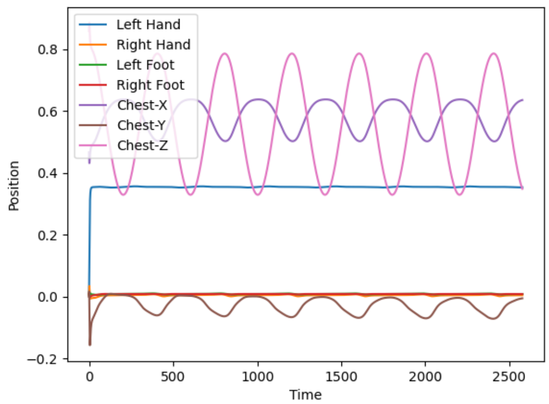

Fig. 4. Tip Positions for Asymmetric Pushup where Z is different. All joints work as expected.

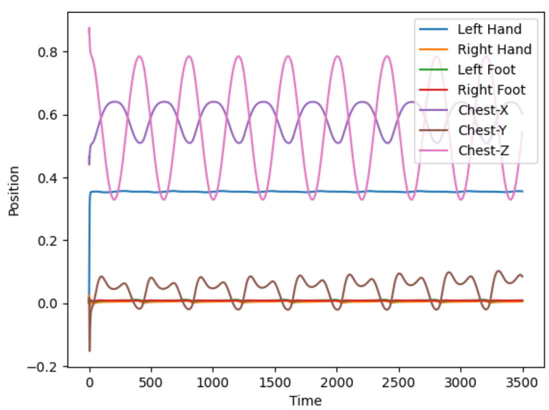

Fig. 5. Tip Positions for Asymmetric Pushup where XYZ are different. All joints work as expected.

C. Further Work

Our project presents a lot of prospects for future analysis. Atlas doing pushups was a great problem for us to apply a lot of the concepts we learned in lectures and through the course of the class. The most apparent possible future direction is to relax our assumptions and include forces in the model, which would also require us to balance the center of gravity of the robot. Another might be to explore closed-chain kinematics in a humanoid robot (perhaps with fewer DOFs than Atlas).

APPENDIX

Github: Atlas Does Pushups!
Video link: Atlas Workout

ACKNOWLEDGMENT

Our group is very grateful to Professor Günter Niemeyer and the TAs for putting together a very interesting class, and lots of help and motivation throughout the course of the project!

IMAGES

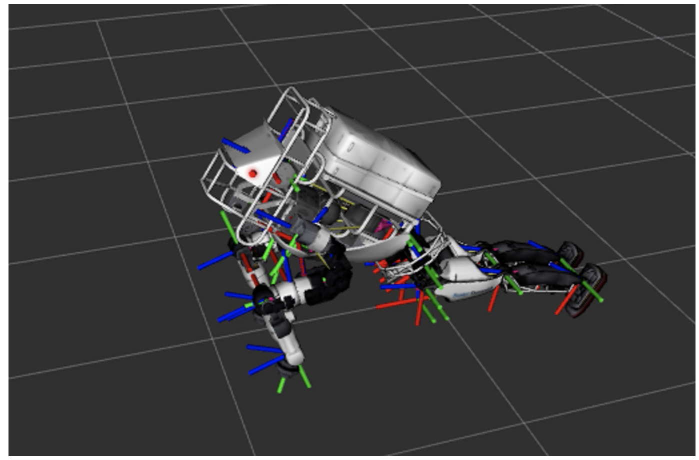

Fig. 6. Chest Pushups Result in Elbows in the Wrong Direction

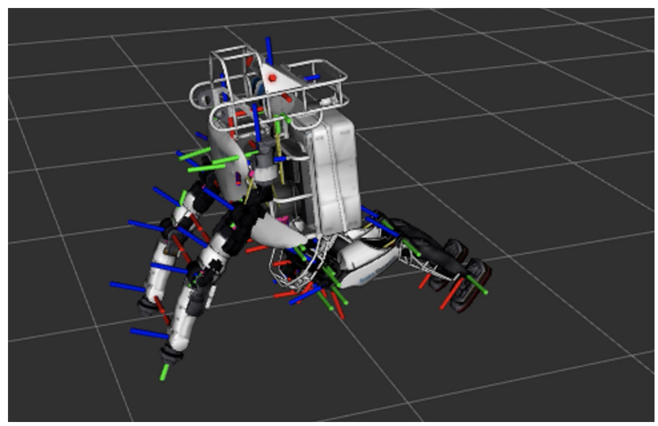

Fig. 7. Lower Pushup Position when Pelvis was the Trajectory Point

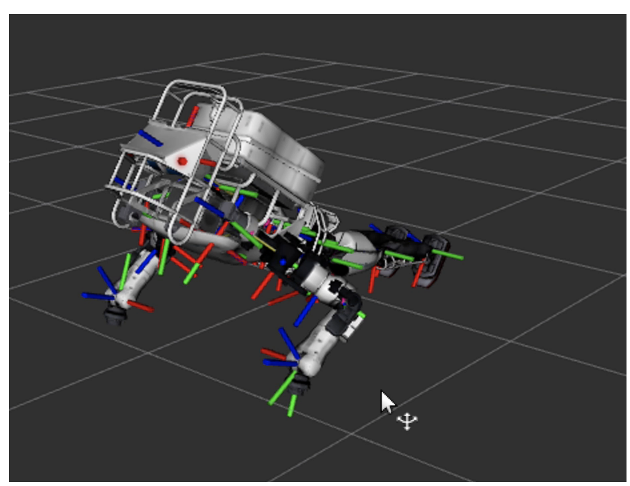

Fig. 8. After Weighting the Joints and Centering, elbows are in the correct position