Chester that Dawg

ME/CS 133A Final Project Report

Alex Vazquez, Gael Moran, Alicia Zhang

December 2023

1 Links:

1. Code: `https://github.com/GaelMor/133a_Project`
2. Board STL File: `https://www.thingiverse.com/thing:197149`
3. Video: `https://drive.google.com/file/d/1pinTAVLcd_7Fv4MEzfPGn_rWWimuEtNk/view?usp=sharing`

2 Introduction

Our team used the spotmicro.urdf and added in our own snowboard.stl, resulting in a robotic dog whose left legs stand on a snowboard while its remaining legs remain off to skate on the ground. The goal of our robot was to create a skating motion with the right legs while keeping the left legs stationary on the board. We also had a task of having the left legs move the dog body up and down relative to the snowboard. We wanted a specific elbow configuration for the dog legs as well as a path of robot motion, which ultimately ended up with it skating through 3-D space using a trajectory determined by vector of these inputs: base_roll, base_vert, and world_yaw.

2.1 Robot and System:

Spotmicro consists of four 3-DOF legs. For Chester, we added in additional axes of movement to the base (spotmicro's main body/trunk) consisting of base_vert, base_roll and base_pitch. We also connected the robot body back to a world frame, with additional axes for the world_horiz, world_vert, world_y, and world_yaw.

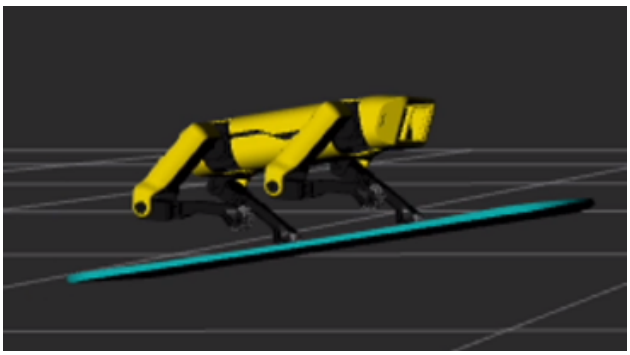


Figure 1: Chester in motion

The robot world path's x, y, and z locations as a whole are determined by the vector that changes as the base_roll and world_yaw change. We can think of it like a 3-DOF arm with pan and tilt joints.

We have a moving base with no redundancies. Our constraints are on the front and rear left shoulder_link, locking them and thus end up with one less joint than our original system.

3 Algorithm and Implementation:

Please refer to page 8 (the last page) for the graph of kinematic chains of our spotmicro URDF.

3.1 Tasks For Left Toes

For the left toes, the tasks were defined relative to the world_yaw_d link frame. The primary tasks were of dimension 2 corresponding to the (x,z) position of the toes. The desired position for the front left toe (relative to the world_yaw_d link frame) was kept constant at fl_{p0}, which corresponds to the toe's position when all joints in its kinematic chain are set to 0. The desired position of the rear left toe was set to $fl_{p0} + Rot_y(\alpha)(rl_{p0} - fl_{p0})$, where rl_{p0} corresponds to the toe's position when all joints are zero and α corresponds to the value of the base_roll joint. In other words, the position of the rear left toe is set such that its distance from the front left toe remains constant and the direction from the front left toe is dictated by α. Note the joints base_vert, base_roll, base_pitch in the graph of the urdf.

The columns in the Jacobian for the primary task of the front left and rear left toes that correspond to the base_roll, base_pitch, base_vert joints are set to zero. This is because these joints are changed over time to see how the robot's legs respond. The usage of base_roll was described in the previous paragraph. Like the base_roll, the base_pitch joint controls the orientation of the robot relative to the world_yaw_d frame. Unlike the base_roll joint, the base_pitch joint controls the rotation about the x-axis. Because of how the desired position was set up, the base_vert joint essentially controls the height of the robot's base relative to the board. If the value of the base_vert joint is too high, this can cause a singularity in the tasks of the left toes (refer to the video). This is because the robot would be set too high in the world_yaw_d frame, leaving the toes incapable of reaching their desired positions.
Furthermore, the column in the Jacobian that corresponds to left shoulder joints (front_left_shoulder for the the front left toe and rear_left_shoulder for the rear left toe) is also set to zero. This is because we wanted the shoulder joint on the left side of the robot to align with the base_pitch joint such that the robot's board orientation also moves when the base_pitch joint is set. To account for this singularity we used the psuedo weighted inverse of the Jacobian J to calculate the inverse kinematics for both the front left and rear left toes tasks. Thus for the primary task of each toe we used:

$$J_W^+ = J^T(JJ^T + \gamma_l^2 I_3))^{-1}$$
$$\dot{q}(t_k) = J_W^+(v_d + \lambda_l(pd(t_{k-1}) - p(t_{k-1})))$$

The task dimension is 2, and the dimension of the Jacobian J is 2×6. The desired velocity is v_d, while $pd(t_{k-1})$ is the desired position of the last iteration and $p(t_{k-1})$ is the actual position in the last iteration. Everything is with respect to the world_yaw_d frame. The positions and velocities are set accordingly whenever this equation is applied to either the front left toe or rear left toe.

We wanted to keep a certain configuration in the legs of the dog as it bends down and also in the skating motion. We specified an 'elbow down' configuration to keep the knees from bumping into each other and (to keep the legs from switching between elbow up and elbow down configuration, refer to the video between 1:50 to 2:00). To accomplish this, we updated the equation from above to have:

$$\dot{q}(t_k) = J_W^+(v_d + \lambda_l(pd(t_{k-1}) - p(t_{k-1}))) + ((I_6 - J_W^+ J)\dot{q}_s)$$

$$\dot{q}_s(t_k) = \lambda_{sl} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -\frac{\pi}{2} - \beta(t_{k-1}) \end{bmatrix}^T$$

where β corresponds to the value of the left_foot joint (front_left_foot for the front left toe and rear_left_foot for the rear left toe).

3.2 Tasks for Right Toes

The tasks for the right toes were defined relative to the world_yaw_d link frame. The primary tasks were of dimension 3 corresponding to the position of the toes. We wanted these toes to move in an elliptical pattern such that they make contact with the 3D plane that the board is on. First, we created the elliptical motion with $\alpha = 0$ (base_roll is set to zero).

$$pd_{fr,\alpha=0}(t) = fr_{p0} + \begin{bmatrix} -0.06 \cdot sin(\omega t) \\ 0 \\ 0.04 - 0.04 * cos(\omega * t) \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0.02 \end{bmatrix}$$

where fr_{p0} is the position of the front right toe relative to the world_yaw_d frame when all joints in its kinematic chain are set to zero. Note that an offset in the z-direction of -0.02 is chosen to cause the board to make contact with the plane that it is placed on. This offset accounts for the thickness of the board. To generalize this for any value of α we set the desired position of the front right toe like so:

$$pd_{fr}(t) = fr_{p0} + Rot_y(\alpha)(pd_{fr,\alpha=0}(t) - fr_{p0})$$

Similar to the left toes, the position of the rear right toe is set such that its distance from the front right toe remains constant and the direction from the front right toe is dictated by α. Hence we use the following for the desired position of the rear right toe:

$$pd_{rr}(t) = pd_{fr}(t) + Rot_y(\alpha)(rr_{p0} - fr_{p0})$$

where rr_{p0} is the toe's position relative to the world_yaw_d frame when all joints in its kinematic chain are set to zero.
For the task of the right toes, the columns in the Jacobian for the primary task of the front right and rear right toes that correspond to the base_roll, base_pitch, base_vert joints are set to zero for the same reason as stated before. Like before, we also used a weighted pseudo inverse for the right toes:

$$J_W^+ = J^T(JJ^T + \gamma_r^2 I_3))^{-1}$$

$$\dot{q}(t_k) = J_W^+(v_d + \lambda_r(pd(t_{k-1}) - p(t_{k-1})))$$

where γ_r and λ_r are the gains corresponding to the right toes. Since the task dimension is 3, then the dimension of the Jacobian J is 3×6. We also included the same secondary tasks as before (elbow down configuration for right legs), so the full inverse kinematics is:

$$\dot{q}(t_k) = J_W^+(v_d + \lambda_l(pd(t_{k-1}) - p(t_{k-1}))) + ((I_6 - J_W^+ J)\dot{q}_s)$$

$$\dot{q}_s(t_k) = \lambda_{sr} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -\frac{\pi}{2} - \beta(t_{k-1}) \end{bmatrix}^T$$

where λ_{rl} is the gain of the secondary task corresponding to the right toes and β corresponds to the value of the right_foot joint (front_right_foot for the front right toe and rear_right_foot for the rear rightx toe).

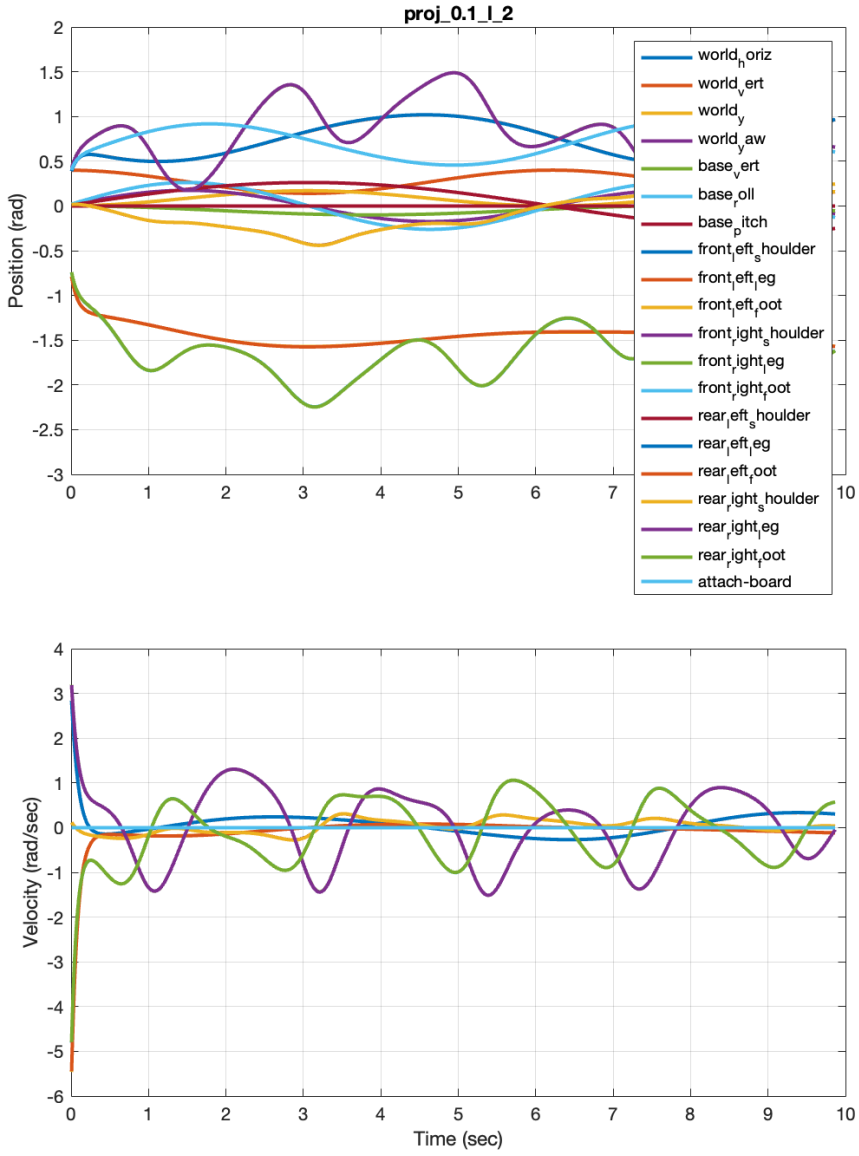
4 Particular Features and Obstacles:

We ran into a singularity when changing the vertical position of the base (base_vert), in which the right legs could not reach the ground constant we defined. This is demonstrated in the video at 01:12

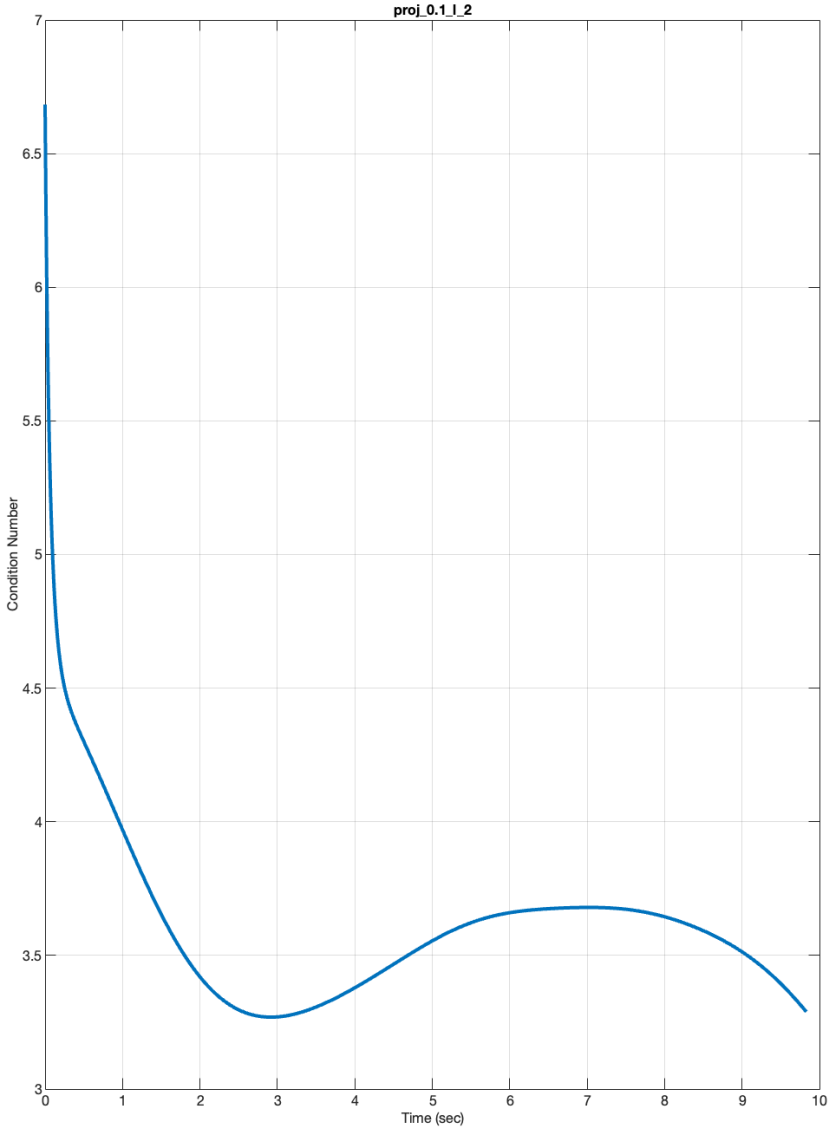
When we change the height of base with respect to the board and keep the front left toe as a constant, we need to lock the shoulder joints.
We had an initial approach to tether the snowboard to the left front foot. However, this would cause the snowboard orientation to change with the toe, which we did not want. This was fixed when we changed our Jacobian to a 3x3, since we were previously overspecifying our rotation and position. This simplified the motion. We also changed the toe_link to attach-board to a revolute joint. This allowed us to adjust the angle such that we keep the snowboard 'parallel' with the ground.

5 Analysis:

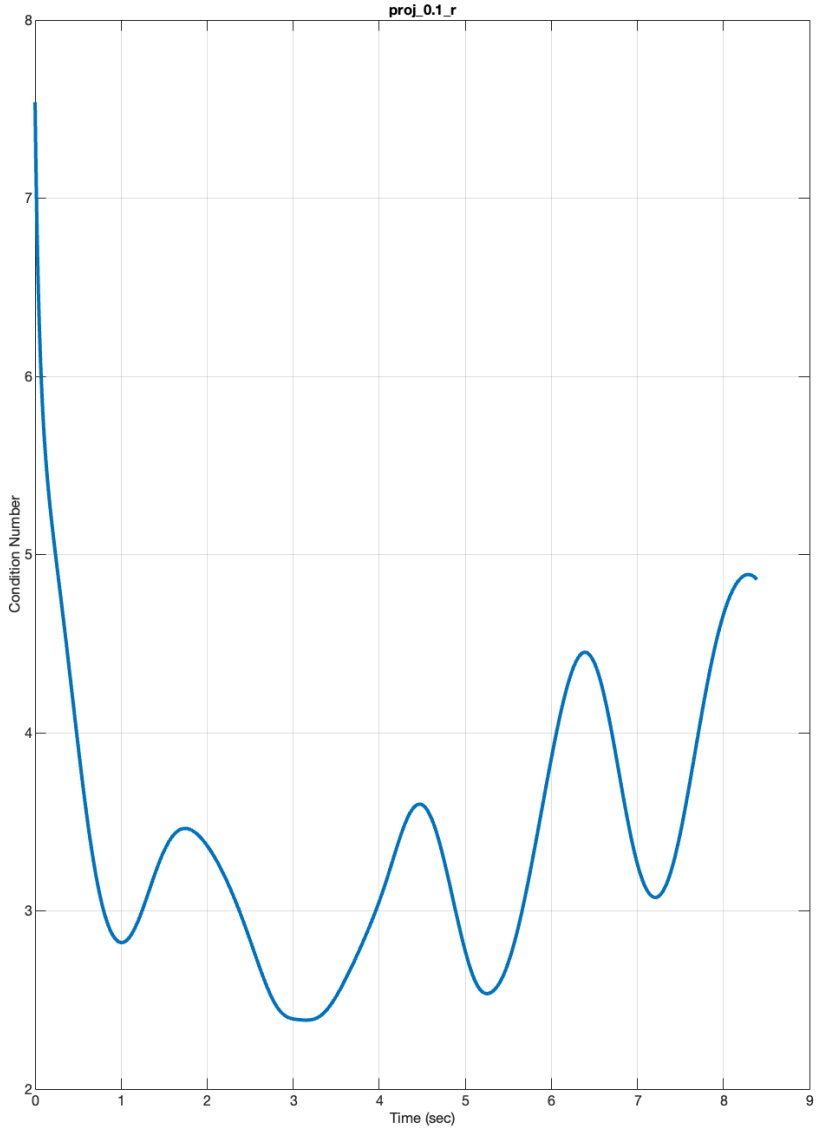
Plot of the Joint Values



Plot of Condition Number for Left Toes



Plot of Condition Number for Right Toes

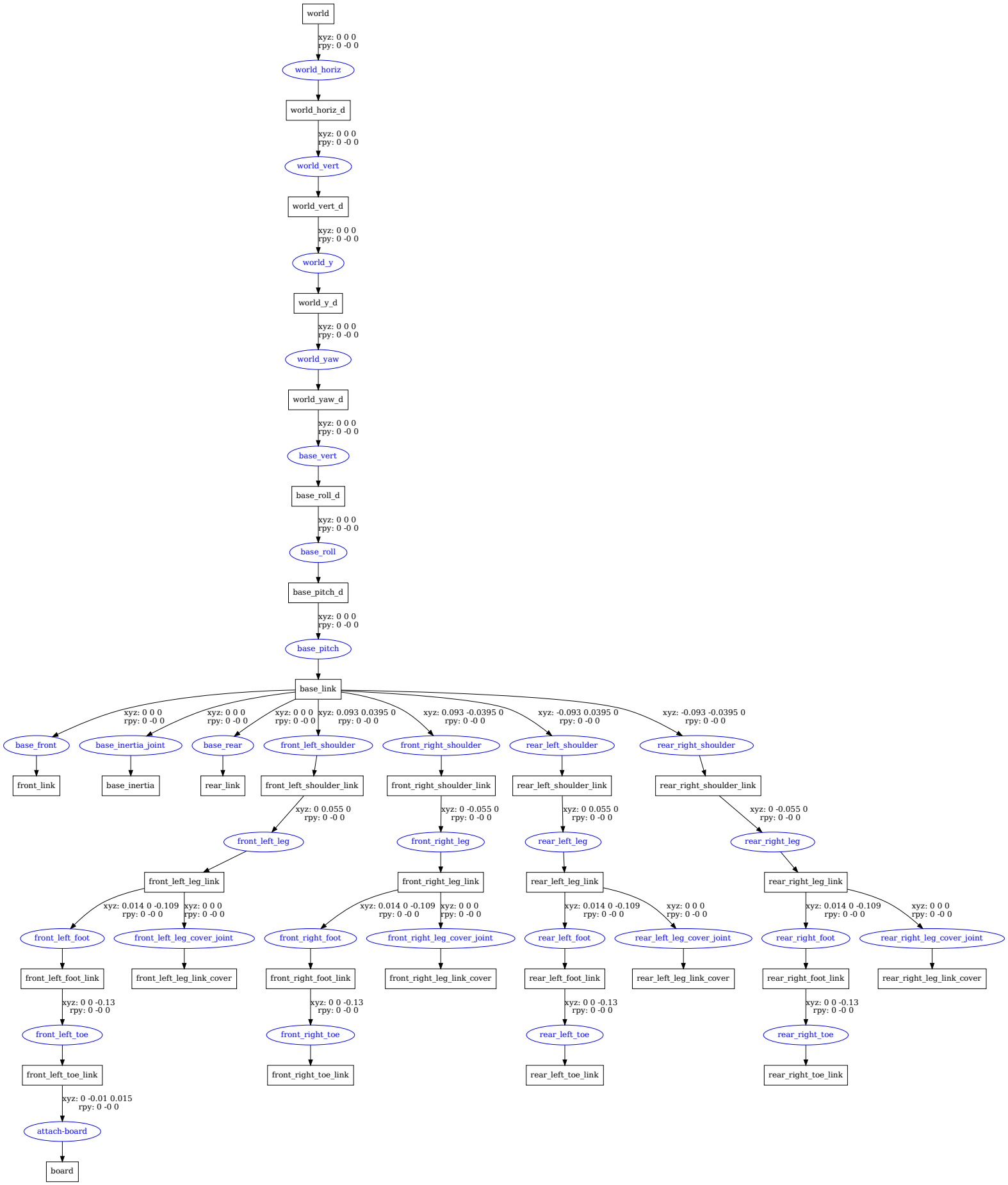


Using the analysis above (plotting joint values with the corresponding condition numbers) we decided to use the values for the gains listed below as we found this to be a good balance achieving our task and preventing a singularity.

$$\lambda_l = \lambda_r = 20$$

$$\lambda_{sl} = \lambda_{sr} = 10$$

$$\gamma_l = \gamma_r = 0.1$$


world
xyz: 0 0 0
rpy: 0 -0 0
world_horiz
world_horiz_d
xyz: 0 0 0
rpy: 0 -0 0
world_vert
world_vert_d
xyz: 0 0 0
rpy: 0 -0 0
world_y
world_y_d
xyz: 0 0 0
rpy: 0 -0 0
world_yaw
world_yaw_d
xyz: 0 0 0
rpy: 0 -0 0
base_vert
base_roll_d
xyz: 0 0 0
rpy: 0 -0 0
base_roll
base_pitch_d
xyz: 0 0 0
rpy: 0 -0 0
base_pitch
base_link
xyz: 0 0 0
rpy: 0 -0 0
base_front
front_link
xyz: 0 0 0
rpy: 0 -0 0
base_inertia_joint
base_inertia
xyz: 0 0 0
rpy: 0 -0 0
base_rear
rear_link
xyz: 0 0 0
rpy: 0 -0 0
xyz: 0.093 0.0395 0
rpy: 0 -0 0
front_left_shoulder
front_left_shoulder_link
xyz: 0 0.055 0
rpy: 0 -0 0
front_left_leg
front_left_leg_link
xyz: 0.014 0 -0.109
rpy: 0 -0 0
front_left_foot
front_left_foot_link
xyz: 0 0 -0.13
rpy: 0 -0 0
front_left_toe
front_left_toe_link
xyz: 0 -0.01 0.015
rpy: 0 -0 0
attach-board
board
xyz: 0 0 0
rpy: 0 -0 0
front_left_leg_cover_joint
front_left_leg_link_cover
xyz: 0.093 -0.0395 0
rpy: 0 -0 0
front_right_shoulder
front_right_shoulder_link
xyz: 0 -0.055 0
rpy: 0 -0 0
front_right_leg
front_right_leg_link
xyz: 0.014 0 -0.109
rpy: 0 -0 0
front_right_foot
front_right_foot_link
xyz: 0 0 -0.13
rpy: 0 -0 0
front_right_toe
front_right_toe_link
xyz: 0 0 0
rpy: 0 -0 0
front_right_leg_cover_joint
front_right_leg_link_cover
xyz: -0.093 0.0395 0
rpy: 0 -0 0
rear_left_shoulder
rear_left_shoulder_link
xyz: 0 0.055 0
rpy: 0 -0 0
rear_left_leg
rear_left_leg_link
xyz: 0.014 0 -0.109
rpy: 0 -0 0
rear_left_foot
rear_left_foot_link
xyz: 0 0 -0.13
rpy: 0 -0 0
rear_left_toe
rear_left_toe_link
xyz: 0 0 0
rpy: 0 -0 0
rear_left_leg_cover_joint
rear_left_leg_link_cover
xyz: -0.093 -0.0395 0
rpy: 0 -0 0
rear_right_shoulder
rear_right_shoulder_link
xyz: 0 -0.055 0
rpy: 0 -0 0
rear_right_leg
rear_right_leg_link
xyz: 0.014 0 -0.109
rpy: 0 -0 0
rear_right_foot
rear_right_foot_link
xyz: 0 0 -0.13
rpy: 0 -0 0
rear_right_toe
rear_right_toe_link
xyz: 0 0 0
rpy: 0 -0 0
rear_right_leg_cover_joint
rear_right_leg_link_cover