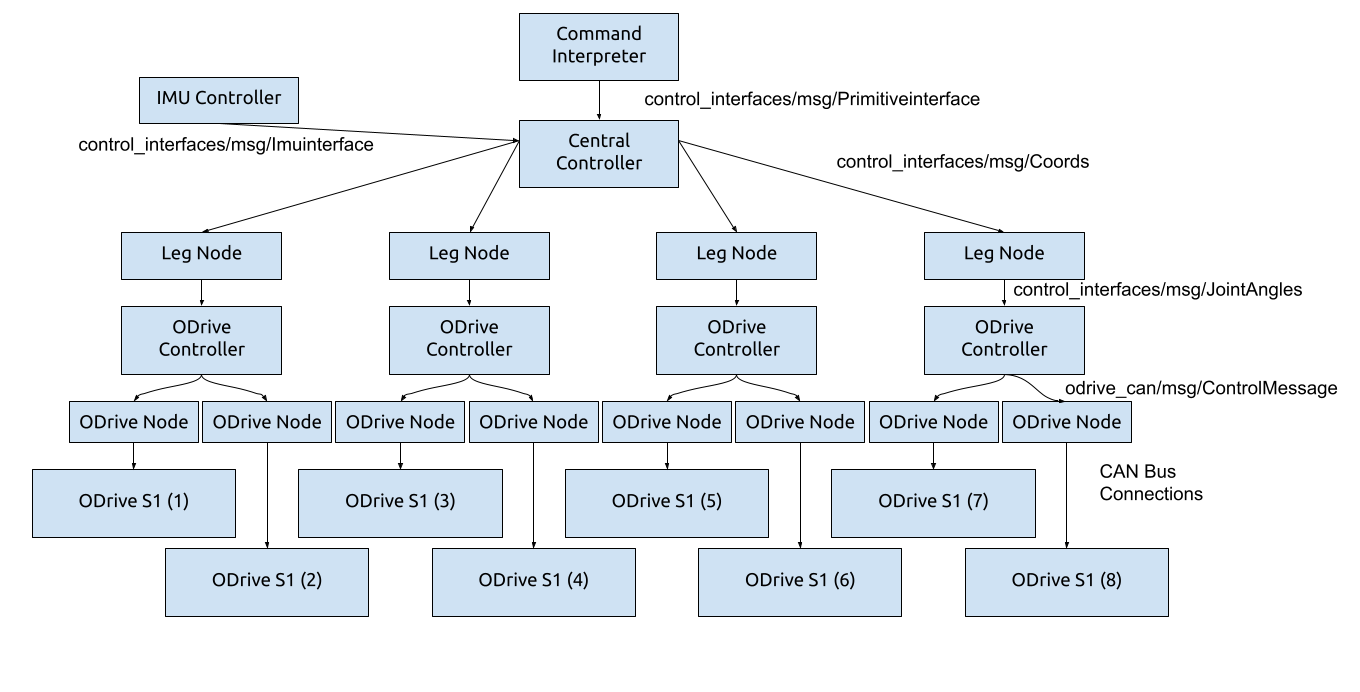
The overall layout of the ROS nodes can be seen in Figure 1. The ODrive Nodes are from the ODrive CAN package, and one is needed for each ODrive. The ODrive controller is a python script that translates between our code and the ODrive nodes, and does so leg by leg. Each leg node executes inverse kinematics to translate a set of desired coordinates (control\_interfaces/msg/Coords) to a pair of joint angles (control\_interfaces/msg/JointAngles) matched to the correct joint. Both nodes are set up to accept an argument from a launch file to control which leg attaches to which ODrive. The Central Controller (cpp\_controller\_node/src/IMU\_node\_cli.cpp) executes the ramps that comprise each motion. It also has the groundwork for listening to the IMU (See README.txt) and adding it into the ramps. The Command Interpreter takes in a message in English to “walk” or “stand” or “sit” and translates and sends it to the Central Controller to execute. It is formatted to be easily integrated into either an Xbox controller node or WASD controller node.

Figure 1. Block diagram of the code driving the robot.

On the robot, the code can be found in the $HOME/Downloads/WalkerFoundation/basewalk\_ws directory. That directory contains a README in addition to the other documentation here. Much of the code contains commentary on ways in which it could be improved and how it was intended to be expanded. It can be convenient to take the SD card from the OrangePi and mount it to your own computer to modify the code in your IDE of choice rather than over SSH.

The packages described above are inside the src directory of the code. The ros\_odrive package is the ODrive Node, leg\_node and odrive\_controller are self-describing, commander is the command interpreter, cpp\_controller\_node is the Central Controller, and control\_interfaces is where all the messages, launch files, and service files are stored. The messages shown in the above block diagram are the only ones currently in use.