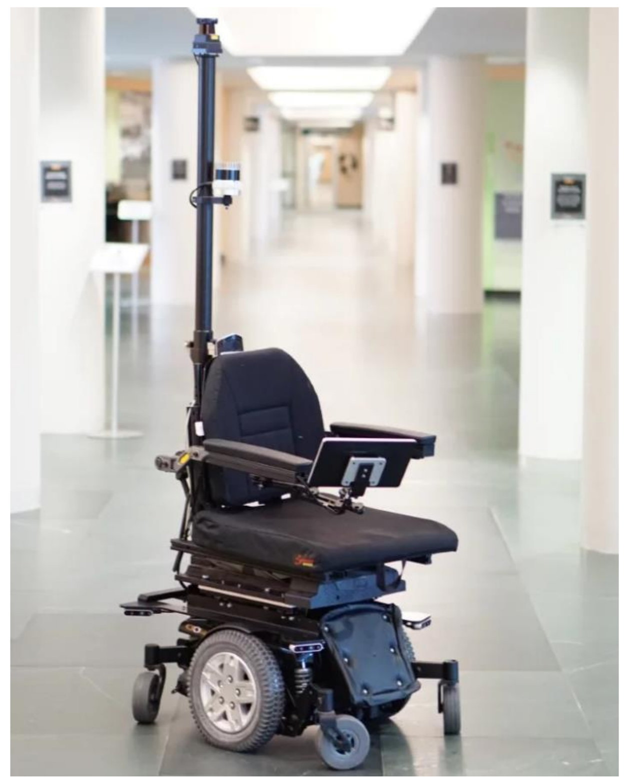
AUTONOMOUS SELF-DRIVING WHEELCHAIR LATERAL DRIFT DETECTION

STUDENTS: JOSHUA MAUER, CHENG ZHU, YESIBAO MUHAMATI

CYBERWORKS ROBOTICS

Purpose and Background

- The Autonomous Wheelchairs increase freedom and ease of mobility for the most vulnerable peoples in society.
- However, some navigation suffers from occasional loss of loop closure and localization. This can be disastrous in the real world either in a hallway or roadway or sidewalk.
- This project is named as Wheelchair Alive and is sponsored by Cyberworks Robotics. It seeks to develop a system that uses camera vision to check for lateral drift of the wheelchair from the desired path.

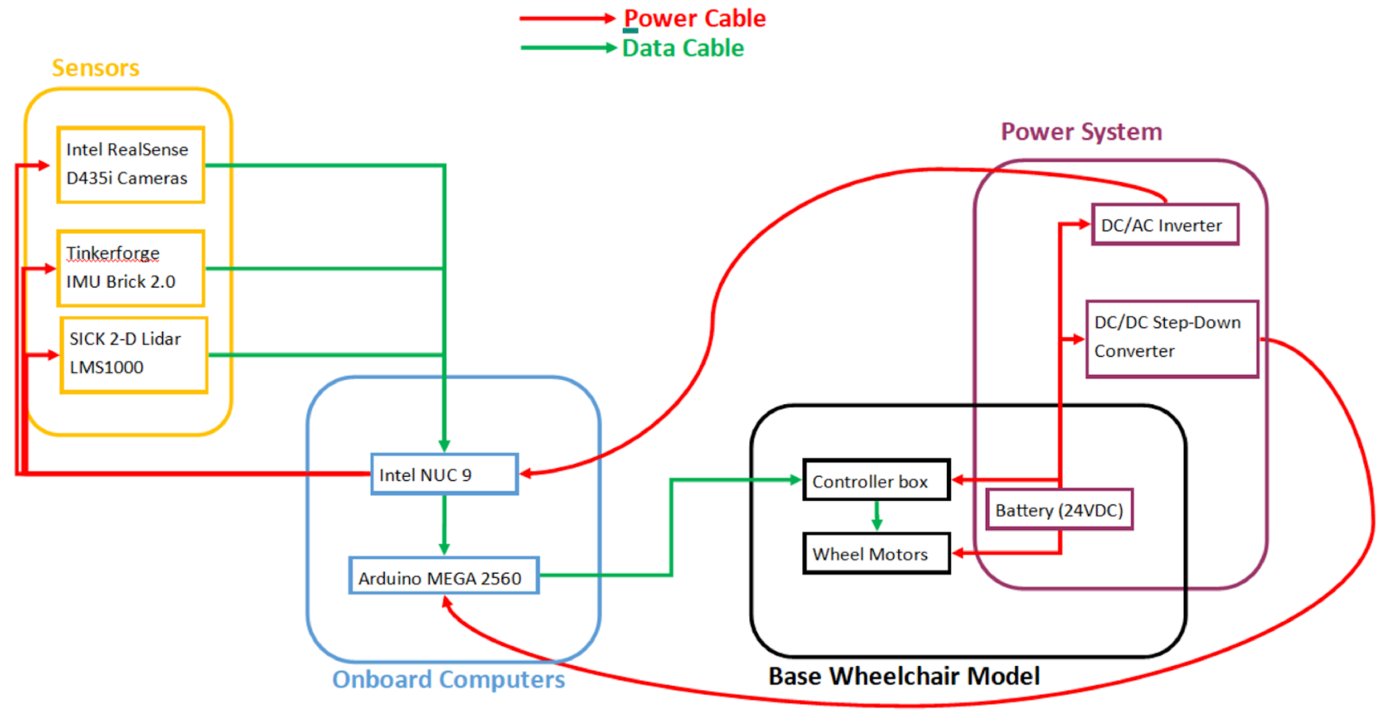


Project Requirements

- The software system of this project needs to detect the lateral drift of the wheelchair and how to compensate it.
- The control system of this project needs to take-in the lateral drift and the. compensate inputs and output a signal to the hardware to adjust the wheelchair to be on the desired path.
- An essential hardware requirement of this project is the Intel RealSense Depth Camera D435i(Showing on the graph on the right).

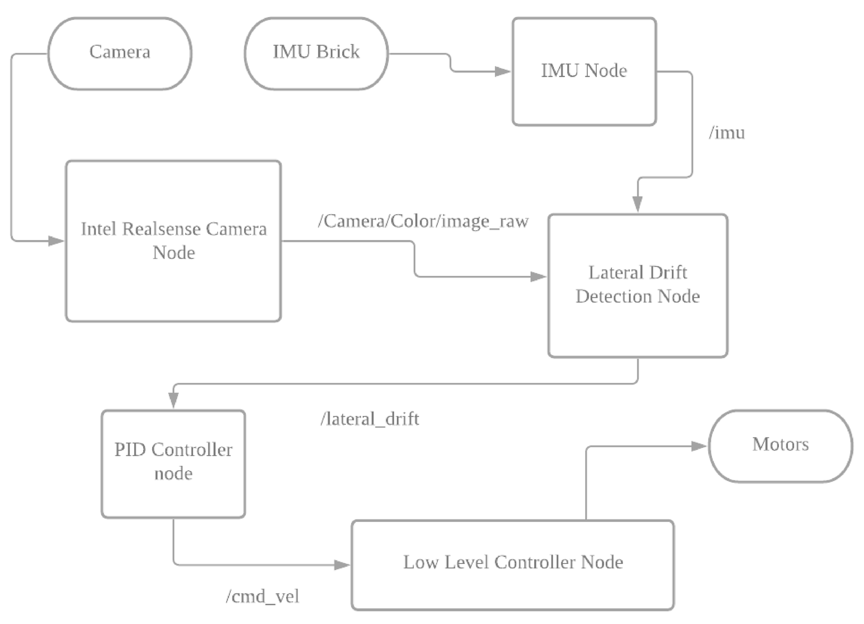
Intel Realsense Camera D435i

Wheelchair Block Diagram



ROS and its Applications

- ROS is an open-source meta operating system used in programming Robots. It consists of packages, software, building tools for distributed computing, architecture for distributed communication between machines and applications. It also provides tools and libraries for obtaining, building, writing, and running code across multiple computers.
- In this project, we are using ROS as a middleware to pass messages between processes and implement functionalities and low-level device control.



Hardware (Mounting Design, Arduino, and Joystick Emulation)

An Arduino 2560 MEGA (Not the exact one used)

An Intel NUC 9 Extreme Edition desktop computer (Not the exact one used)

- The autonomous wheelchair's onboard computing stack consists of two main components—an Intel NUC 9 Extreme Edition small-form-factor desktop computer, and an Arduino 2560 MEGA microcontroller.
- The Intel NUC unit serves as the processing "brain" of the chair and implements the lateral drift detection software that we have designed (software specifics are patent-pending)
- The Arduino 2560 unit receives speed and direction information from the NUC unit via USB and translates the required speed and direction into output pin voltages sent to the PWM pins.
- This allows the Arduino to emulate the joystick output needed to control the chair; the Arduino output voltage pins are connected to the port on the wheelchair's control box that ordinarily accepts input from a joystick.

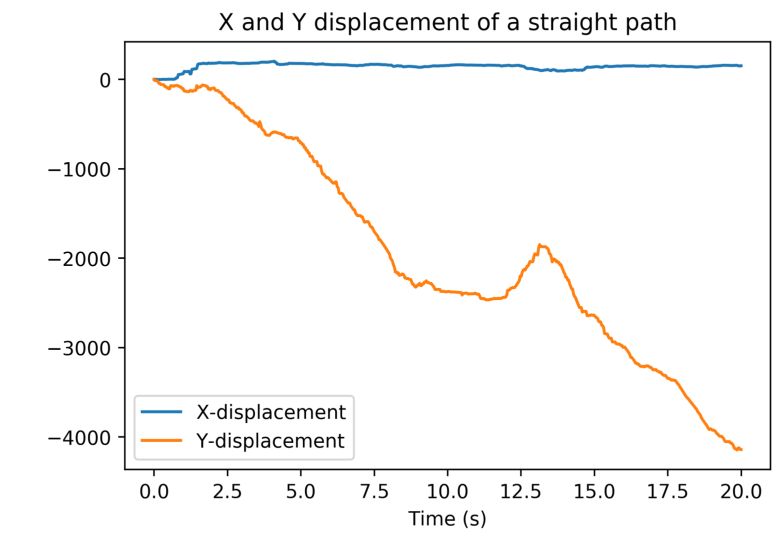
Safety/Ethical Considerations

Safety/Ethics, to avoid some negative effect of this project on safety and ethics, we purpose to do these:

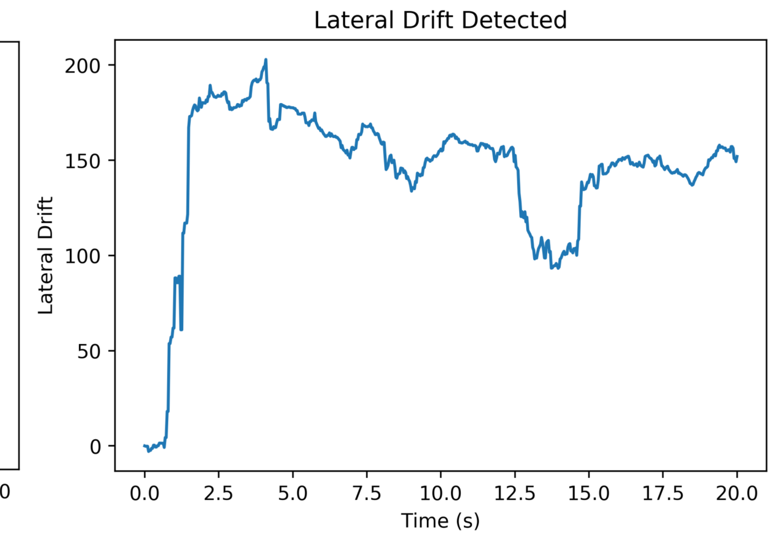
- Test the normal function of our product in normal circumstances and make it as perfect as possible
- Identify as many corner cases as possible
- Test those corner cases and make sure that our product have a solution to these cases
- Have multiple testing periods(to improve our product and identify and provide solution to new corner cases) before we bring our product to use
- Set up a user-feedback system that can collect all the data from the unpredicted situation and make improvements on products after the product has been brought to the public
- Provide a detailed, clear, and useful user manual to let the user to understand our product better so that it can suggest users not to use our product in unsupported situations

Test Results

Our lateral drift detection system is designed to output the lateral drift messages under ROS at a frequency of 5Hz. The results are given by comparing the integral displacements on x-axis and y-axis with our desired path.



Wheelchair's displacement while moving straight



Lateral drift derived from calculated path and desired path

Future Work, References, and Acknowledgments

- Further improvements to detect trivial drift caused by camera mounting error
- Further improvements to deal with drift caused by bumping

Faculty: Blake Hannaford
Graduate Student: Daniel King

References (For Tools Used):
https://www.arduino.cc/
http://wiki.ros.org/ROS/Introduction
https://dev.intelrealsense.com/docs/docs-get-started

ELECTRICAL & COMPUTER ENGINEERING
UNIVERSITY of WASHINGTON

ADVISORS: VIVEK BURHANPURKAR (INDUSTRY MENTOR), BLAKE HANNAFORD (UW)

SPONSORS: CYBERWORKS ROBOTICS, UNIVERSITY OF WASHINGTON