On the Use of Modular Software and Hardware for Designing Wheelchair Robots

Martin Gerdzhev and Joelle Pineau
School of Computer Science
McGill University, Montreal, Canada
martin.gerdzhev@mail.mcgill.ca, jpineau@cs.mcgill.ca

Ian M. Mitchell
Department of Computer Science
University of British Columbia
mitchell@cs.ubc.ca

Pooja Viswanathan and Genevieve Foley
Intelligent Assistive Technology and Systems Lab
University of Toronto
pooja.viswanathan@utoronto.ca, genevieve.foley@mail.utoronto.ca

Abstract

This short paper describes experiences in the development of several smart power wheelchair platforms across three different sites. In the course of the project, we have re-used several of the components (both hardware and software) despite differences in the base platform of the robots. We describe the different platforms, and discuss some of the challenges and results of our work.

Introduction

It is reported that among power wheelchair (PWC) users, about 40% have difficulty with daily maneuvering (Fehr, Langbein, and Skaar 2000). Given that over 4.3 million people in the US are wheelchair users (Simpson, LoPresti, and Cooper 2008) and that the number is only going to increase as the population ages, it makes sense to incorporate smart features into PWCs that will automate challenging tasks. Such “smart” wheelchairs aim to allow more people to be able to safely and effectively use a PWC, by reducing the physical and cognitive load required for operation. A remarkable number of smart wheelchairs have been developed over the years, many of them taking the approach of enhancing commercially available PWC models with a variety of sensors, processors, and custom software. Yet it seems like each new project must “reinvent the wheel” and invest a lot of time and effort in order to develop software/hardware that other groups may have already developed.

This position paper reports on the collaborative design of smart wheelchair platforms at three sites: McGill University, the University of British Columbia and the University of Toronto. An unprecedented sequence of cross-Canada team grants—initially the CanWheel emerging team, then the Canadian Field Robotics Network, and now the AGE-WELL Network of Centres of Excellence—has allowed us over the last five years to jointly develop and share hardware and software components, expertise and personnel while pursuing parallel avenues of investigation in this challenging research space. The results of this collaboration are a collection of wheelchair platforms that share some core components while still providing a diversity of points in the design space to better explore tradeoffs between functionality, flexibility, usability, testability, commercializability and other design dimensions.

Challenges in the Development and Testing of Smart Power Wheelchairs

The team at the University of British Columbia (UBC) began the collaboration using a Quickie Rhythm PWC modified by AT Sciences to include their DriveSafe collision avoidance system (Simpson et al. 2004), custom magnetic rotary encoders for odometry, and a framework of metal tubing around the back and sides of the chair which held a laptop and has supported a variety of sensors including the original DriveSafe sonar and infrared (IR) rangefinders, a bumper skirt, a Bumblebee stereo camera, Hokuyo planar LRFs, Asus Xtion RGB-D cameras, and accelerometers;
Modular Design of Smart Power Wheelchairs

There are many different brands and models of PWCs on the market that can be equipped with a variety of interfaces, seating, drive configurations and other customizations, so it is important to be able to abstract away as many of the hardware differences as possible, and develop add-on modules that are independent of the type of wheelchair used. One major challenge we have encountered (across all platforms) is the interface between our added on-board computer and the original motor controller of each PWC. Somewhat fortunately several brands of PWC use the same kind of motor controller (originally developed by PG Drives and now part of Curtiss-Wright) which allows an external joystick to be connected via an interface variously called the Omni+, R-Net Omni or R-Net Input/Output module. Even with this commonality, the motor controller’s interface remains proprietary and a reverse engineered joystick emulation is our only means of motion control.

The board that we built to interact with the motor controller incorporates a standard off the shelf Arduino with a custom designed shield. The Arduino’s standard USB interface is used to communicate through ROS to a laptop, while the shield has ports for the Curtiss-Wright module and a PWC joystick. This combination allows us to take input from the user, route it through ROS nodes on the laptop to record and potentially modify that input, and then deliver it to the motor controller. Because there is no mechanism for feedback from the PWC base, the shield also provides a port to capture odometry data from custom designed and mounted wheel encoders. Unfortunately, joystick commands do not map well (or even consistently) to the linear and angular velocity commands generated by standard ROS libraries. Where available, feedback from the odometry has enabled design of a PID feedback loop that considerably improves linear and angular velocity control; however, such odometry is not always available and even with it control is imperfect due to the black box nature of the proprietary motor controller into which the joystick signals are fed.

Nevertheless, this board has been successfully incorporated onto six PWCs in the three groups, with only minor tweaks of parameters to account for different electrical and mechanical profiles among the four different commercial models represented. Currently we are working on algorithms that will further refine the joystick control signal with a goal of achieving consistent motion behaviour among the various models of PWCs.

Modular Software Development

With the emergence of ROS the development of modular code that can be ported from one system to another has become significantly easier. There exist drivers and ROS interfaces for the majority of important robotics sensors—such as laser range scanners, 3D cameras, IMUs—that publish standard messages for use in other packages. External libraries like OpenCV and PCL can also be used seamlessly. This effectively allows for building software that is abstracted from the specific hardware, and is only dependent on whether the correct message type is available.

In the course of this project, we have developed several modules that can be used across platforms.

- Module for people detection and tracking from planar laser scans
- Module for detection of tables and ideal docking locations
- Remote control module
- Parallel parking module
- Back-in parking module
- Wall following module
- Module for basic navigational commands with obstacle avoidance (forward/backward motion and turning)
Our Smart Wheelchairs

The McGill SmartWheeler platforms

The team at McGill University in Montreal has built two different smart wheelchairs – a Sunrise Quickie Freestyle (Figure 2) and a Sunrise Quickie S646 (Figure 3). Both models are equipped with laser range-scanners, an Asus Xtion RGB-D Camera, custom magnetic wheel-encoders, IMU, GPS, touch screen, remote control joystick, and WiFi connectivity. The Freestyle model has two SICK LMS100 planar laser range-scanners (one facing forward and one facing backward), while the S646 Model has three Hokuyo UHG planar laser range-scanners (one in the back and two on the sides) as well as multiple sonars (to detect transparent surfaces like glass that are common in malls). Both wheelchairs use the custom Arduino controller board with an Omni+ module in order to send control commands from ROS.

The UBC platform

The UBC platforms consist of a pair of Permobil M300 bases with power tilt seats and width adjustable cushions. We have discarded the IR, sonar and bump sensors from the older platform, and are currently using Asus Xtion Pro RGB-D cameras (instead of stereo) and/or Hokuyo LRFs.

Evaluation of Smart PWCs

With the modular hardware and software components we have developed, we are now able to not only reduce the time it takes to develop new algorithms and functionality by reusing ROS modules, but also replicate our systems across multiple wheelchair bases fast and easily, allowing us to.

Figure 2: The Quickie Freestyle Smart Wheelchair

Figure 3: The Quickie S646 Smart Wheelchair
conduct user trials across sites and at a larger scale. In addition, the compatibility of our hardware components with Curtiss-Wright controllers, which are currently used by several wheelchair bases, opens up the possibility to simply add on our systems to study participants own power wheelchairs that are often customized for them, thus overcoming seating issues. This could potentially reduce the number of PWC bases that would need to be purchased to run larger-scale trials, a savings of $3000 - $30,000 per chair.

However, before we are able to conduct more user testing with the target population, we must overcome several obstacles. First, our interface to the chair electronics emulates a joystick. Consequently, we cannot get any feedback from the black-box controllers (such as a direct measure of odometry, speed profiles, tilt, etc), cannot send any data to the heads-up display, and we do not have a simple mapping from input signals to motion. Third, these off-the-shelf chairs have no mounting points for sensors, so in adding support structures to the system we must balance carefully between the flexibility and convenience desired for engineering prototyping and the aesthetics desired for user studies.

Seating is still an issue with some of the wheelchairs. In the McGill group, the Quickie Freestyle is primarily used by students for internal testing of the algorithms since its seat does not lend itself well to the big modifications that may be needed to accommodate people with disabilities. The Quickie S646 is used for user studies with wheelchair users since it has a more adjustable seat, but still may not be appropriate for users with more complex seating needs.

Despite differences in sensor configurations between PWCs, the data collected by the different teams can be very useful in order to create benchmark datasets for scene and object recognition, and real users’ PWC driving data which is currently difficult to obtain/access. User studies are time-consuming and challenging to run (recruitment, etc.), so sharing existing study data can expedite development, although privacy and ethics issues need to be addressed.

Conclusion

Our experience so far suggests that it is feasible and productive to develop diverse robot platforms across sites while making use of modular software and hardware architectures to share successful component designs.

Nonetheless, it is worth noting that different users may have difficulty with different tasks and may require only certain “smart” features. This situation further emphasizes the desirability of developing modules that can be added or removed as needed, similar to how car manufacturers offer different options, such as adaptive cruise control, backing cameras, pedestrian detection, etc. In the case of our projects, the availability of several distinct platforms at the different sites has allowed us to investigate different configurations without overwhelming development effort, while still pushing us to develop portable solutions which can be shared when appropriate.

The modular systems we have implemented and ongoing efforts to increase compatibility with commercial wheelchairs can thus speed up development efforts and facilitate larger scale trials with our target users in order to validate system usability and effectiveness, which will be an essential step in the eventual commercialization of smart wheelchair technologies.

Acknowledgements

Funding for this project has been provided by the NSERC Canadian Field Robotics Network (NCFRN), CanWheel Emerging Team in Wheeled Mobility for Older Adults (AMG 100925), AGE-WELL Network of Centres of Excellence, and Collaborative Health Research Program (CHRP).

References