Multi-Modal Sensor and HMI Integration with Applications in Personal Robotics

Rommel Alonzo, Sven Cremer, Fahad Mirza, Sandesh Gowda, Larry Mastromoro, Dan O. Popa
[a]Next Generation Systems, Nedderman Hall 250, 416 Yates Street, Arlington, Texas 76010

ABSTRACT

In recent years, advancements in computer vision, motion planning, task-oriented algorithms, and the availability and cost reduction of sensors, have opened the doors to affordable autonomous robots tailored to assist individual humans. One of the main tasks for a personal robot is to provide intuitive and non-intrusive assistance when requested by the user. However, some base robotic platforms can't perform autonomous tasks or allow general users operate them due to complex controls. Most users expect a robot to have an intuitive interface that allows them to directly control the platform as well as give them access to some level of autonomous tasks. We aim to introduce this level of intuitive control and autonomous task into teleoperated robotics.

This paper proposes a simple sensor-based HMI framework in which a base teleoperated robotic platform is sensorized allowing for basic levels of autonomous tasks as well as provides a foundation for the use of new intuitive interfaces. Multiple forms of HMI's (Human-Machine Interfaces) are presented and software architecture is proposed. As test cases for the framework, manipulation experiments were performed on a sensorized KUKA YouBot® platform, mobility experiments were performed on a LABO-3 Neptune platform and Nexus 10 tablet was used with multiple users in order to examine the robots ability to adapt to its environment and to its user.

Keywords: Multimodal, Multi-user, Multi-sensor framework

1. INTRODUCTION

In the field of robotics, sensors are necessary to give a platform a view of the world. Depending on the type of sensor, different details of the environment can be observed. A multi-modal combination of sensors will give the platform a more complete view of its environment. Hence, autonomous robotic platforms accommodate a wide variety of sensors. These sensors can provide the system with 2D and 3D views of the environment using a combination of narrow and wide-view stereo cameras and provide a floor-level view of any obstacles a robot might face during navigation using LIDAR sensors. Robots often include additional sensors for localization and for task-specific applications. For example, the PR2 built by Willow Garage, includes a force sensor, IMU and accelerometers that make it especially unique for application-based robotic research [4]. Many researchers have used the PR2's sensors with the intent of tasking the PR2 with basic server [4] or butler capabilities [5]. Continued research has shown that the PR2 is capable of further developing these basic tasks into more complex operations such as cooking and folding cloths [6]. Other platform's, like the Baxter from Rethink Robotics, have simple sensors like a Sonar and RGB-D sensor [13]. With reduced sensors comes reduced cost and this has allowed the Baxter to be easily adopted by the research and industrial community. Applications include automated baggage checking [7] and dynamic surgical tool tracking [8].

HMI's (Human Machine Interface) allow humans to manipulate and operate robotic platforms and other intractable systems. Most platforms only provide a complex GUI interface on an off-platform computer made by the manufacture or researcher [10,11]. Joysticks provide a simple method for interfacing with platforms and they often only serve as a method for manipulating a robot with simple commands, mostly direct control over a robot. The amount of feedback provided to the user with respect to the current state of the robot is limited. More recently, frameworks for human-robot interaction with multiple sensors and input devices are being proposed [1-3].

In this paper we propose one such framework to handle multi-modal sensors, in particular robot sensor skins, as well as multiple modern interface devices, such as tablet applications. This framework allows the robot to perform autonomous, semi-autonomous, or teleoperated tasks such as navigation, object pick and place, and so on. The framework would also allow the system to quickly evaluate and incorporate user preferences and new forms of system inputs, in order to increase the Human-Robot Interaction productivity. As a test study, we describe the hardware and software

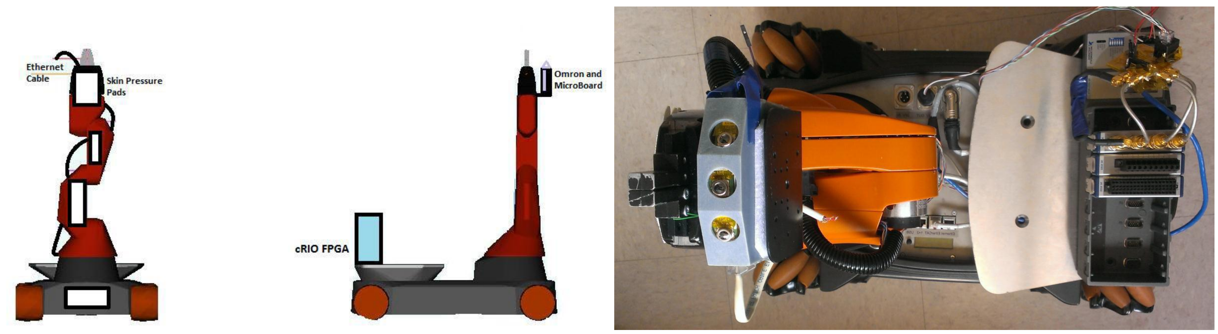
Next-Generation Robotics II; and Machine Intelligence and Bio-inspired Computation: Theory and Applications IX, edited by Dan Popa, Muthu B. J. Wijesundara, Misty Blowers, Proc. of SPIE Vol. 9494, 949409
 · doi: 10.1117/12.2177641

enhancements performed on two robotic platforms in our lab, and report on experimental results with their use. The paper is organized as follows: in Section II describes the Hardware and Software enhancements done to the DR20 system, Section III describes the HMI algorithms, Section IV describes our Experimental Results and Section V concludes the paper and outlines future work.

2. SYSTEM OVERVIEW

Hardware Description

We implemented the proposed framework on the KUKA YouBot® and Neptune platforms available in our lab. The YouBot is a mobile manipulator (Figure 1) consisting of 4 mecanum wheels that allow it to perform holonomic maneuvers in all planar directions. On the robot base, a 5 DOF arm allows the platform to manipulate objects with a two-fingered gripper [14]. The robot arm was outfitted with 16 different pressure sensors embedded in patches of silicone rubber and fitted with plastic buttons for force concentration, while the base was fitted with 10 pressure sensors in the form of skin surfaces. It has a response time of 5-20μsec, with a pressure range between 0 and 1lbs to 0 and 7000lbs with proper circuitry. Mean force a human exerts with one hand is around 45lbs[15]. The interface circuit was set to measure between 0 and 50lbs. Furthermore, IR sensor arrays (4 x 16 sensals total) were installed near the end-effector of the robot (Figure 1b).



(a) (b)

Fig 1. Diagrams of force and pressure sensors mounted on our Kuka Youbot arm®, and top view of robot with pressure, IR sensors and cRIO installed.

We designed a custom data acquisition board (or MicroBoard) to collect and condition sensed pressure data from up to 19 taxels in one centralized location, digitally sample it and transmit this data using SPI and I2C protocols to a NI cRIO real time controller mounted on the robot base [16]. Network cables were snaked through the Kuka Youbot from within the arm to avoid twisting these during operation. Several skin patches consisting of silicone pads with embedded pressure sensors were also installed on the robot to implement physical interaction behaviors (Figure 2).

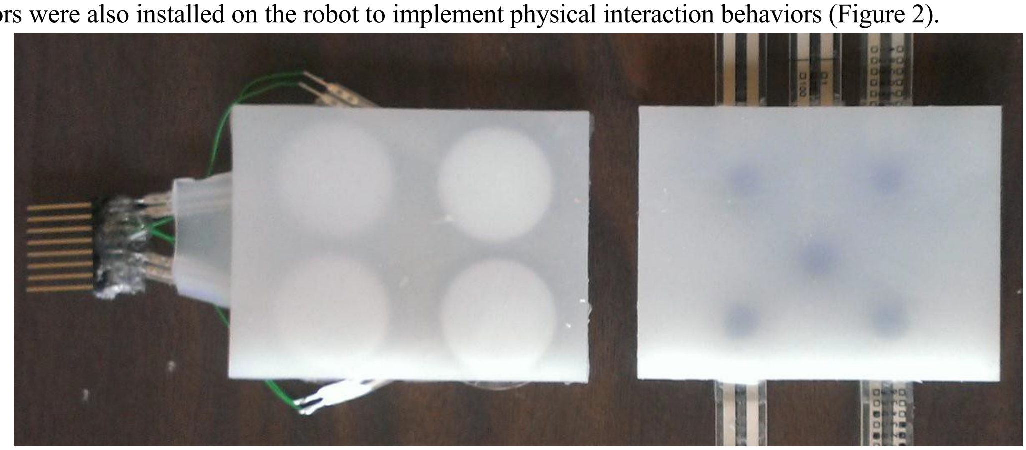


Fig 2. Skin cell pads with embedded pressure sensors and buttons into white silicone P-10 layer

In contrast to the Youbot ®, the Neptune robot (Figure 3a) is a conventional two-wheeled differential drive platform that has been retrofitted with sensors and several HMIs and is used as a testing platform for advanced navigation algorithms. With these additional modifications to its onboard sensors, the Neptune can exhibit autonomous navigation capabilities. The base platform for Neptune is the LABO-3 ® platform from AAI Canada, Inc., equipped with a PC/104 processor from Advanced Digital Logic, Inc., 10 Infrared sensors, one set of bumper sensor at the front with 5 touch sensors and another at the rear with 2 touch sensors. A URG-04LX-UG-01 scanning range finder from HOKUYO has been installed on top of the base IR sensors for the purpose of area mapping and obstacle avoidance. It has a field of view of 240 degrees and the measurement distance of 4m. A single board, an ADL855 PC interface board has been installed to acquire and process data from vision sensors under the Robot Operating System (ROS) framework. A 9 DOF–Razor IMU was also integrated with Neptune incorporating three sensors – a gyro ITG-3200, ADXL345 accelerometer and HMC5883L magnetometer. ATmega328 is used to process the outputs of all the sensors and output over a serial interface. Considering a base IMU which reads out sensor data and would require an additional device for attitude determination which in this case is solved by having an AHRS(Attitude and heading reference system)–onboard processing system solving attitude and heading solutions. These are achieved by a form of non-linear estimation such as an extended Kalman filter. An Asus Xtion PRO LIVE was added to the laser scanner for viewing purposes, and an Asus RT-N65u router was installed for implementing wireless HMI's, collect data and visualization. A tablet app for an Android Nexus 10 was coded to provide a virtual joystick, pitch and roll information, and other button selection functionality.

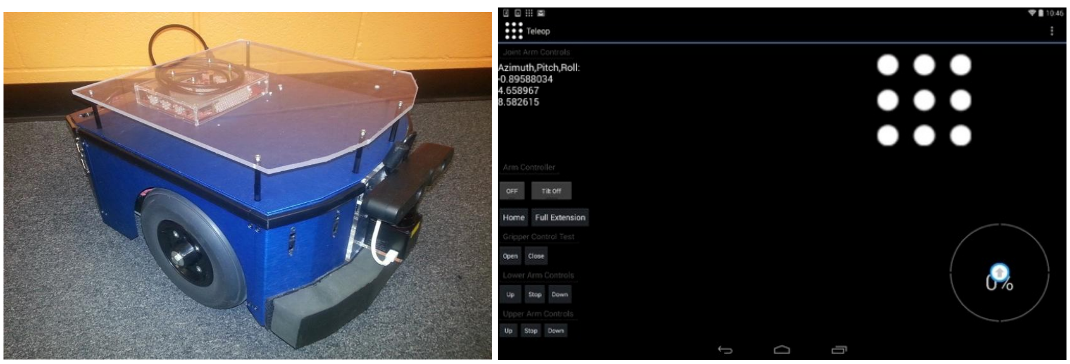


Figure 3: Neptune Robot retrofitted with LIDAR, IMU and 3D depth camera, and an Android tablet app for its teleoperation.

2.2 Software Description

A multi-layered ROS-based framework was formulated to enable the two robotic platforms to act autonomously as well as enable users take control via several HMIs such as tablets or touch. We tested our enhancements by carrying out autonomous and semi-autonomous tasks such as navigation through the environment and object pick-and-place. In order to enable autonomous tasks, several hardware and software enhancements were implemented to take advantage of the underlying robotic framework. The system architecture was built with data flow and data capture in mind. This allows for data to be easily tractable while maintaining minimum bandwidth along communication lines. Software layers can be easily modified in order to allow for basic adjustments to the architecture without interfering with other system services. Figure 4 shows a diagram of the software architecture separated into 3 layers based on providing a unique and important service: sensor, processing and intuitive interface.

The Sensor Layer provides hardware enhancements through sensors that are important for HMI algorithms, and in particular the 3D depth camera, used for object and people detection, the LIDAR, used for collision avoidance and autonomous navigation, and the pressure sensitive skins used for physical HMI. Raw sensor data collected is reformatted, conditioned and re-organized by this layer to be used by the robots. The sensor layer is running on individual Single Board Computers (SBC) on each platform, for instance on the c-RIO for the Youbot platform, or on the quad core SBC retrofitted onto Neptune.

The Interface Layer provides user input from different interfacing devices and communicates back to the Processing Layer. Several different types of interface devices can be used to dictate to the Processing Layer which algorithms and instructions take priority. Today, most people have access to a smartphone, tablet or any touch-screen enabled device. In keeping with this trend, our aim was to develop an intuitive application that can easily control the state of the entire system architecture as well as provide direct control over a robotic arm and/or mobile base.

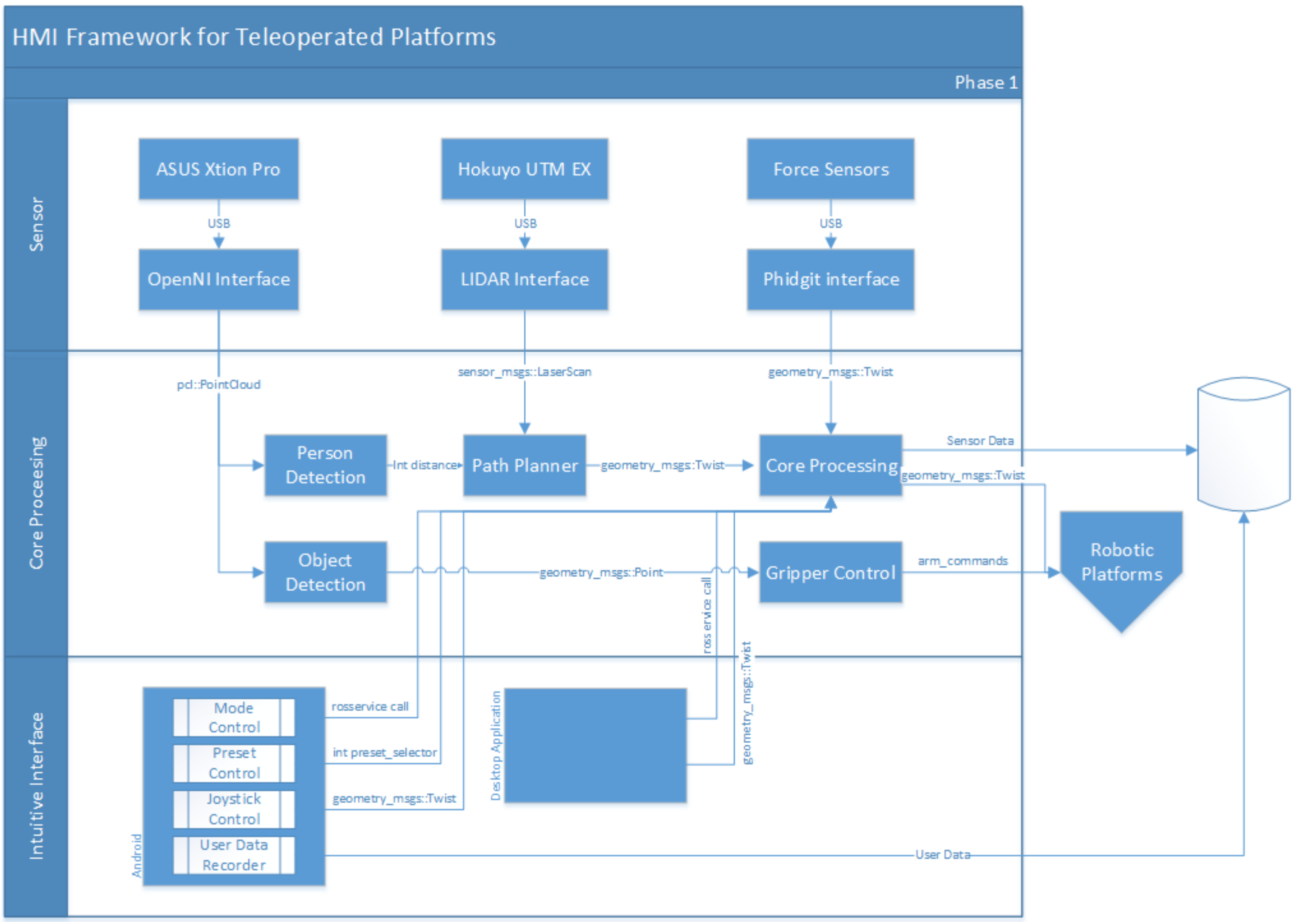


Figure 4: An organizational diagram for the proposed 3-layer software architecture

We use the Robot Operating System as the foundation for our framework. For instance, we can rely on SLAM, Visual Servoing and other advanced autonomous algorithms as the supporting algorithms during implementation. SLAM provides localization and mapping capabilities based on laser scan data from a LIDAR sensor As the Neptune maps out new areas and floors in an unknown building, those maps are saved for later use or for another platform using a similar system framework. Visual Servoing is also employed through the use of 3D depth camera.

3. HMI SCHEMES

In this section we describe several HMI algorithms that have been implemented to connect sensors, operators and robotic platforms.

Joystick Base Control

A virtual joystick control scheme is used for teleoperating the robot bases from the tablet app. The finger position of the user is recorded in terms of (x,y) coordinates relative to the joystick center. By converting these coordinates into polar form, we can calculate a normalized theta value that represents the angular distance between the center points to a user's contact point:

$$q = \begin{bmatrix} \mathrm{x} \\ \mathrm{y} \end{bmatrix} \tag{1}$$

where q is the position vector from the user's finger placement on the tablet screen. We use q coordinates as command vectors by taking θ in polar form and use trigonometric functions to scale θ into stable command instructions between -1 and 1.

$$\theta = \tan^{-1}(\frac{y}{x}) \tag{2}$$

$$\boldsymbol{C} = \begin{bmatrix} c_x \\ c_z \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \tag{3}$$

where $\boldsymbol{C}$ is the end-point or velocity command vector sent to the Neptune or Youbot base platforms. Once the command vector is calculated, we can adjust these values based off of a preset safety vector or adaptive learning parameters $N(q)$ and communicate these normalized values to the teleoperated platform.

$$\boldsymbol{C}'(q) = N(q)\boldsymbol{C}(q) \tag{4}$$

Tilt Controls

Using the Nexus 10's built-in gravity and gyroscope sensors, we can measure the orientation of the tablet. We define the rotation around the x-axis as roll (θ), rotation around the y-axis as pitch (φ) and the rotation around z-axis as yaw (ψ). Depending on the angle of rotation, we find the difference between the current angular values and initial angular values recorded at the system startup. Since we know the range of these values (0-9.8) based on the gravity sensor properties, we re-normalize these values between 0 and 100. These values are then sent to the robotic manipulator (YouBot). The normalized pitch values are applied as angular velocity values for some of the robot. The normalized azimuth values are applied as angular velocity values for the first joint of the robot arm.

$$\boldsymbol{\Theta}_n(q) = N(q)(\boldsymbol{\Theta}_c(q) - \boldsymbol{\Theta}_i(q)) \tag{5}$$

where $q = [\theta \quad \phi \quad \psi]^T$, $\boldsymbol{\Theta}_n(q)$ is a normalized angular vector, $\boldsymbol{\Theta}_c(q)$ is the current current angular vector, $\boldsymbol{\Theta}_i(q)$ is the initial angular vector and N(q) is a velocity gain. An adaptive gain value can be introduced that can help new users adapt to different platforms while keeping the platform in a safe state. We implemented this algorithm as a joint controller for the YouBot's arm manipulator.

Robot Arm Control

Based on the amount of force applied to individual flexiforce sensors, a robotic motion can be generated in either the robotic arm or mobile base. This algorithm was already described in our previous work [12]. Additionally, as the robotic arm changes orientation, we keep track of the new positions of each individual force sensor, and therefore recompute the effect of physical pushing onto skin sensors. Based on the amount of force applied to each individual sensor, the corresponding force value is directly applied to control velocity along individual joints or in Cartesian space, through a set of gains:

$$\boldsymbol{Q}(q) = N(q)\boldsymbol{F}(q) \tag{6}$$

where $q = [\theta \quad \phi \quad \psi]^T$, is a assigned joint as well as assigned selected sensor, $\boldsymbol{F}(q)$ is the force applied onto the sensor, $N(q)$ is a normalizing vector and $\boldsymbol{Q}(q)$ is the expected velocity applied to every individual joint.

4. EXPERIMENTAL RESULTS

Mobile manipulator platforms like the KUKA YouBot allow for high-dexterity movement of its robotic arm. Without an intuitive interface, an average user wouldn't be able to control the platform to its full potential. On the other hand, mobile robots such as the Neptune can navigate in-door environments in direct teleoperation mode, semi-autonomous, or autonomous mode. Our proposed software framework, allows us to evaluate the performance of HMIs by allowing users to control the pose of the robot arm using the Nexus 10 tablet's orientation sensors, on-screen joystick and visual feedback from onboard cameras. The goal is to quantify the performance and set user preferences for a variety of controllers that can be implemented in joint space as described below. User times were recorded and are shown in Table 1.

Base Movement

In this experiment, we tested a users' ability to move the Neptune and YouBot between several predetermined points on the lab floor map via the Nexus 10 Tablet. Using a small predetermined path, shown in Figure 5, we tested different HMI's by recording the completion time of each experiment and calculate the standard deviation time to determine each interface's performance. Users were asked to navigate the robot forward along a divider wall for a distance of 9 ft, then take a right turn for a distance of 6 ft. Navigation can be performed by the users in two modes: autonomous, in which the user dictates the final destination to the platform, it computes a path based on an available map and executes the motion and teleoperation in Joint space, in which the user sets base velocities in a moving frame tied to the robot. Using linear and angular velocity commands calculated from the joystick control scheme (Figure 5c, 5d), we manipulate the Neptune towards following the path dictated in Figure 5a.

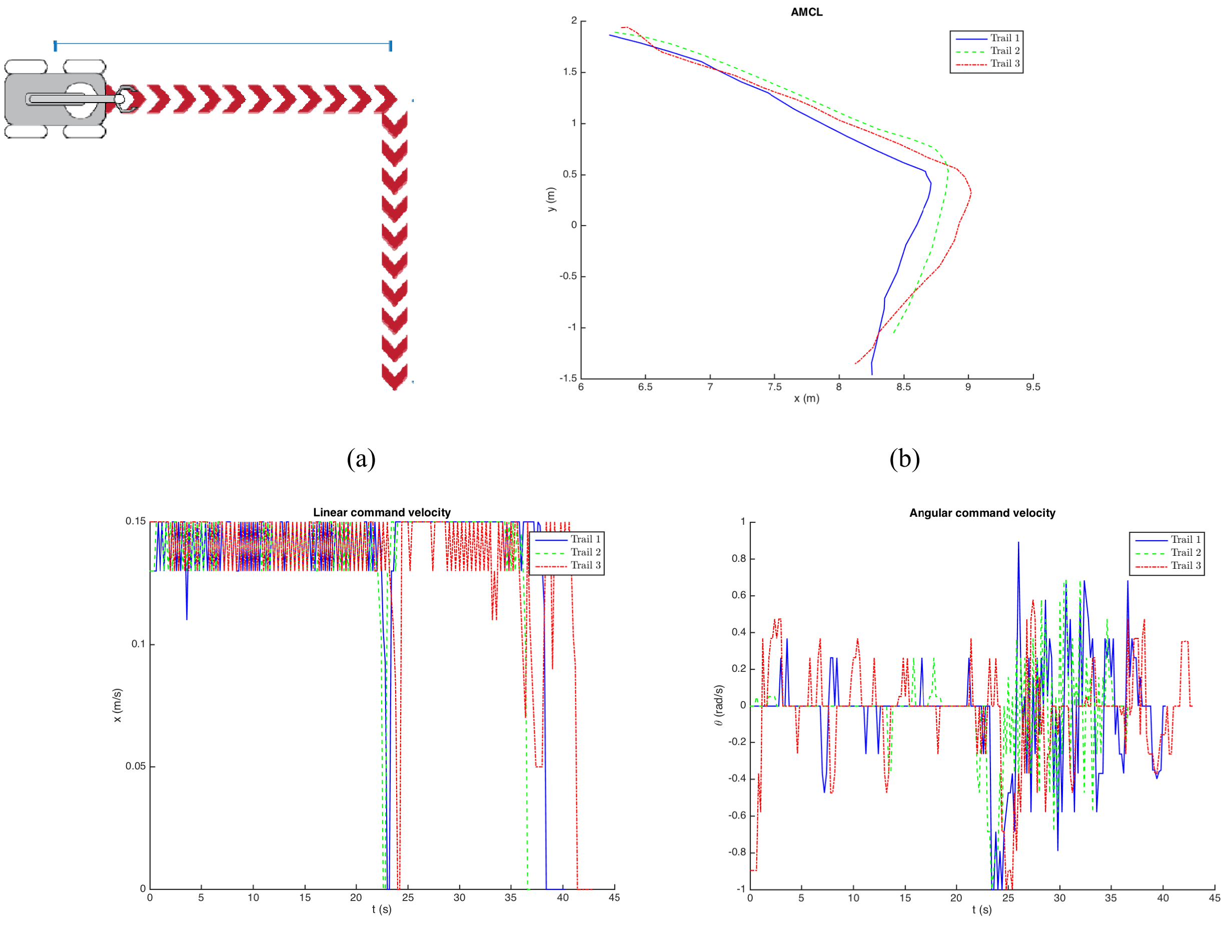
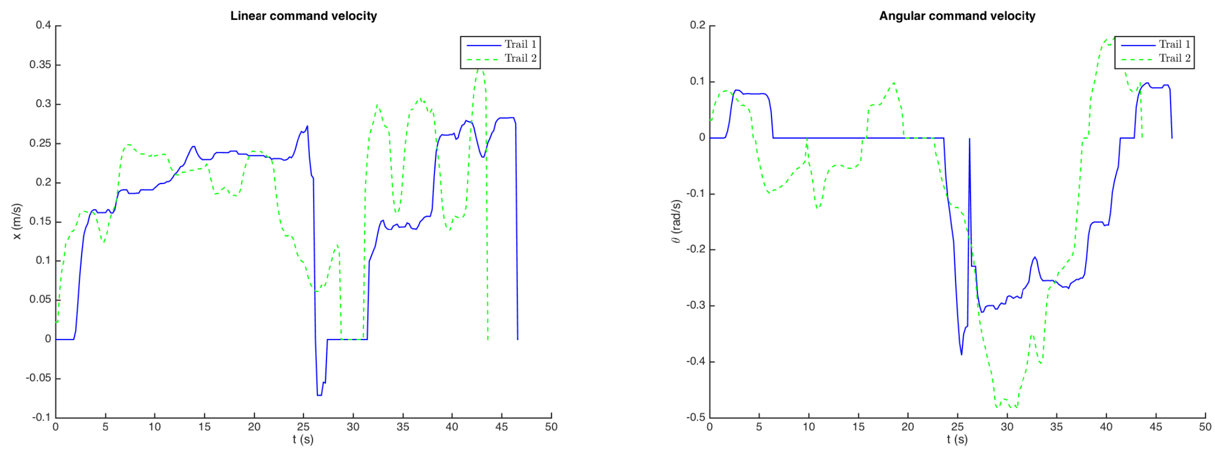


Figure 5: Designated Navigation track (a), Actual path followed by the Neptune (b), Command vectors sent to Neptune(c,d)

Mobile Manipulation

Manipulation with a multi-DOF arm can be a complex task for robots with many joints, such as the YouBot. Using an intuitive interface supported through a joint controller, we expect that users should be able an easier time. In this experiment, the users were tasked to position the YouBot's arm into an optimal position to pick up a soda bottle placed 1 foot away from the platform and place it on it's base plate located behind the YouBot's robotic arm. The task can be completed by controlling individual joints and the base in Joint space via the Tablet as well as the pressure sensors mounted on the arm. With the joint space scheme, the user can select between individual joints to be controlled through the Nexus tablet application.

While not statistically significant due to the limited numbers of trials and users, these experiments show that implementing YouBot arm orientation controls in Joint space through different interface schemes has a significant impact on the pick-and-place completion time.



(a) (b)

Figure 6: Linear and angular command velocity vectors sent to the YouBot using two different methods

Table 1: Experimental results, showing completion time in seconds.

Trial	Neptune mobility (Base Movement)	YouBot Pushing via Pressure (Base Movement)	YouBot Tablet Joystick (Base Movement)	YouBot Tilt Control in Joint Space (Mobile Manipulation)	YouBot Tilt Control in Cartesian Space (Mobile Manipulation)
1	45	22.1	16.1	78.2	40.0
2	46	17.9	18.3	87.6	19.3
3	42	18.2	15.9	80.7	36.6
Average (s)	44.3	19.4	16.8	82.2	32.0

5. CONCLUSION

In this paper we proposed a HMI framework for controlling robotic platforms using visual, as well as contact touch sensors. The framework is formulated in ROS using a three layered design that is scalable and can accommodate a variety of robotic platforms, sensors and interfaces. To demonstrate the effectiveness of our methodology, we conducted navigation and pick and place experiments with robotic platforms Neptune and YouBot. Results show that by providing more sensor modalities, adding autonomy and new types of interfaces, we could extend the platforms usability and features to included sophisticated autonomous tasks as well as provide easy-to-use interface such as a tablet. Our future goals are to study user preferences and correlate them to task completion times so that the user is provided with intuitive controls without being overwhelmed. We will also complete user experiments and tailor the interface gains in an adaptive fashion to preferences on the fly.

6. ACKNOWLEDGMENT

This work was supported by National Science Foundation NRI Grant #IIS-1208623. We wish to thank Joe Sanford, Isura Ranatunga, Joshua Baptist, J Paul Carpenter, and Emrys Maier and UTA Research Institute Staff for their help with the experimental work presented in this paper.

REFERENCES

[1] Iba, S.; Paredis, C. J J; Khosla, P.K., "Intention aware interactive multi- modal robot programming," Proc IROS 2003, 27-31 (2003)

[2] Gorostiza, J.F. ,Barber, R., Khamis A.M., Pacheco, M.M.R., Rivas, R., Corrales, A, Delgado, E., Salichs, M.A, "Multimodal Human-Robot Interaction Framework for a Personal Robot," Robot and Human Interactive Communication, 39-44 (2006)

[3] Iba, S., Paredis, C. J J, Khosla, P.K., "Interactive multi-modal robot programming," Robotics and Automation, 161-168 (2002)

[4] Cremer, S., Ranatunga, I., Popa, D.O., "Robotic waiter with physical co-manipulation capabilities," CASE2014, 1153-1158 (2014)

[5] Bohren, J., Rusu, R.B., Jones, E.G., Marder-Eppstein, E., Pantofaru, C., Wise, M., Mosenlechner, L., Meeussen, Wim; Holzer, S., "Towards autonomous robotic butlers: Lessons learned with the PR2," ICRA2011, 9-13 (2011)

[6] Maitin-Shepard, J., Cusumano-Towner, M., Lei, J., Abbeel, P., "Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding," ICRA2010, 2308-2315 (2010)

[7] Parida, S., Cabrera M.E., Wachs J.P., “Dynamic Surgical Tool Tracking and Delivery System using Baxter Robot,” SURF Symposium, 51 (2014)

[8] DeDonato M.P.; Dimitrov V.; Padir T., “Towards an Automated Checked Baggage Inspection System Augmented with Robots, ”SPIE DSS, 90740N-90740N-11(2014)

[9] Daniel Stonier, “RosJava – ROS Wiki”, http://wiki.ros.org/rosjava

[10] , “Hardware Specs | Willow Garage”, https://www.willowgarage.com/pages/pr2/specs

[11] Estremera, J.; Garcia, E.; Gonzalez de Santos, P., “A Multi-Modal and Collaborative Human–Machine Interface for a Walking Robot”, Journal of Intelligent and Robotic Systems, Paper 35(4), 397-425(2002)

[12] Sanford J., Ranatunga I., Popa D., “Physical human-robot interaction with a mobile manipulator through pressure sensitive robot skin”, PETRA2013, Article 60 (2013)

[13] , “Baxter Research Robot - Technical Specification Datasheet & Hardware Architecture Overview”, http://sdk.rethinkrobotics.com

[14] , "KUKA youBot store," 18 December 2012, www.youBot-store.com

[15] Chaffin, D., Andres, R., Garg A., “Volatile Postures during Maximal Push/Pull Exertions in the Sagittal Plane”, Human Factors, Paper 25(5), 541-550(1983)

[16] , “What is CompactRIO – National Instruments” http://www.ni.com/compactrio/whatis/