A Low-Cost Service Robot for Everyone

Marc Howard*, Anton Mayr* and Alexandros Sigaras*

***Abstract*— This paper presents a low-cost, full-size service robot that can be constructed in two days for under $1500 USD. Built with common hardware, such as the iRobot Create and the Kinect, this robot is able to use many existing libraries for vision and navigation. This paper explains how a body can be easily constructed and how arms can be attached and implemented, so that the robot can perform simple tasks in a human environment. The design and hardware choices when developing such a robot are discussed in order to show the capabilities and limitations of a low-cost robot. The flexibility of the robot is emphasized, as the robot can run under Linux/ROS or Windows and is easily adaptable to work with new hardware and software. The interaction capabilities of the robot are explained, showing how it can take input from voice commands, remote control, and a BCI device. Finally, the paper showcases already implemented applications to give the reader an overview of how the robot can be used, and easy ways to expand upon its functionality.**

I. INTRODUCTION

Over the last five years, service robots have become more and more popular. Existing robotics companies have opened divisions to produce mobile robots for service and assistance. Such robots have been put to use for jobs such as delivering hardware to production lines or painting aircraft [1]. Because of the many applications for service robots, and the subsequent demand, more than 50 service robot manufacturers already exist [2].

However, service robots are still not advanced enough to perform a large variety of tasks in a standard human-occupied environment. They are limited by unsolved problems in fields such as object recognition and grasping, and such limitations lead to a specific and restricted usage [3]. Very advanced robots such as Honda's Asimo or the Willow Garage PR2 provide a good indicator of what is already possible, displaying the capability to complete complex tasks like fetching objects from a refrigerator or folding towels.

The biggest disadvantage of such advanced robots is their cost. Despite their limitations, their cost remains extremely high, making them unaffordable for a common household or a research lab. Luckily, more affordable robots exist, such as the TurtleBot or the KUKA youBot.

TurtleBot is offered at a low price and provides a mobile platform (iRobot Create) and vision system (Microsoft Kinect). It also offers an SDK and various libraries for visualization, planning, perception, control, and error handling. Unfortunately, it does not provide enough capabilities to operate as a service robot in a human environment. It does not come with an arm by default, and it is simply not big enough for many applications, since its height does not allow it to recognize or grasp objects that lie on a table.

*M. Howard, A. Mayr and A. Sigaras are with the Robotic Lab, Department of Computer Science, Columbia University, New York, NY, 10027, USA, (email: {mh3203, ajm2217, as4161}@columbia.edu).
*All authors contributed equally

The youBot comes with excellent hardware (KUKA arms and omnidirectional wheels) and also runs on ROS, but it has a price-tag of over $25,000 USD. Unfortunately, the youBot is also not big enough to fetch an object from a table.

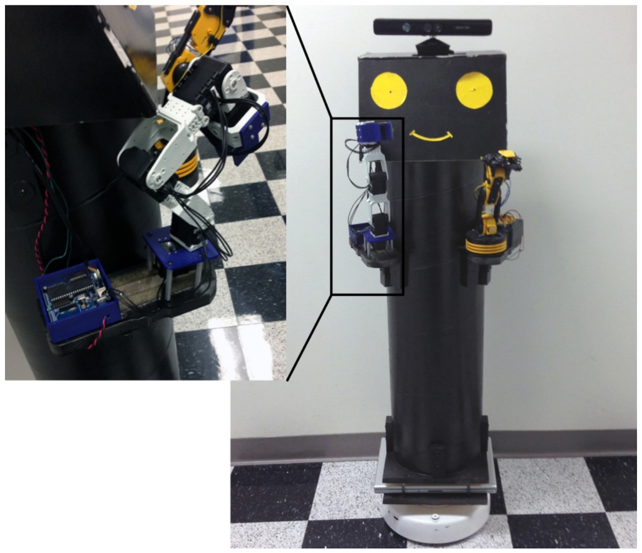


Fig. 1. The Service Robot built at Columbia University, showcasing the ability to use different models of arm. The expanded image shows the Turtlebot Arm mounted on the robot, providing excellent gripping capability.

The absence of a low-price service robot that can interact in a human environment led us to the creation of a new robot (Figure 1). Our robot's hardware supports navigation, vision, and grasping while costing under $1,500 USD. It builds upon the design of the TurtleBot and enhances it with the incorporation of a human-sized body and dual arms mounted at table height. The robot uses the Kinect because of its low price and excellent quality [4], although any vision system can be mounted on the robot. Our work enables interested roboticists to easily replicate our design and build their own service robot, able to perform simple tasks in a household or laboratory environment. Furthermore, we provide software to run the robot either on Linux using ROS or on a Windows system. This allows a broader use of SDKs that are only targeted for a specific operating system, such as the Intel Creative Interactive Gesture Camera [5] [6], or the Kinect Fusion [7] [8].

In this paper we present a complete robot design which can be implemented by a non-expert, and discuss the software

Materials	Price (USD)
iRobot Create	$130
Microsoft Kinect	$90
Two TurtleBot Arms	$500
Arbotix Microcontroller	$60
Construction Materials	$120
TurtleBot Power Board	$80
Total for Components	$980

TABLE I
BILL OF MATERIALS LIST

and user interfaces which make such a robot functional. The main emphasis is on affordable materials that can be easily manufactured at home or in a lab, and on modularity which allows new vision systems or arms to be easily be mounted. We also showcase possible applications for the robot.

The goal of this paper is to give guidance to small labs and interested persons to build their own low-cost robot for the purposes of research and development.

II. DESIGN

This section discusses the process of designing this robot and outlines specific decisions and the reasonings behind them. With this information, skilled replicators should be able to either produce an exact duplicate or construct a robot similar to ours with modifications that better suit their needs. It also provides information about how to use a Windows or Linux/ROS platform on this robot.

A. HARDWARE

The design goal was to keep the cost of the service robot under $1,500 USD without compromising key capabilities such as computer vision, voice recognition, grasping, and manipulation. Table I shows the list of materials that were used to build the service robot. Additionally, the service robot requires an onboard computer to function. A computer should be chosen which matches the level of processing desired from the robot. A $100 USD netbook is sufficient for basic navigation and vision, but a more powerful computer may be desired for more complex tasks. The robot was designed to fit computers with a screen size up to 15", so that most laptops can be used comfortably with the robot. Buying a computer specifically for the robot should add between $100 and $500 USD to the total cost, keeping the total under $1,500 USD.

The construction materials consisted of wood planks, cardboard sheets, metal fasteners, metal angle brackets and metal standoffs, all of which were obtained easily at a local hardware store. The focus was on using light materials that would not add significant weight to the robot, while keeping stability and the limits of the iRobot Create in mind. Wood and cardboard were chosen whenever possible due to their light weight and ease of manipulation. This allowed easy design changes to the robot without having to obtain new materials, and guaranteed that the design could be replicated with basic tools. These pieces could be replaced with metal or plastic if the replicator has a facility that allows manufacture of such materials, though metal will add significant weight to the robot and may overload the Create base.

The robot is wired so that all hardware components draw their power from the rechargable battery in the iRobot Create. This can be recharged by either plugging the Create into the wall, or allowing the Create to dock at an iRobot charger station. This facilitates easy charging, and allows the robot to recharge itself by identifying its charger station and navigating to it. With all systems running, the battery drains in about 30 minutes, although that can be extended if minimal motion is required. The onboard computer runs off its own battery, and does not power any other devices.

Included here is an overview of the robot's construction (Figure 2), including a schematic of the overall layout, size, and basic construction. Additional details and schematics are available online, in a link at the end of the paper, and provide precise building instructions and available code.

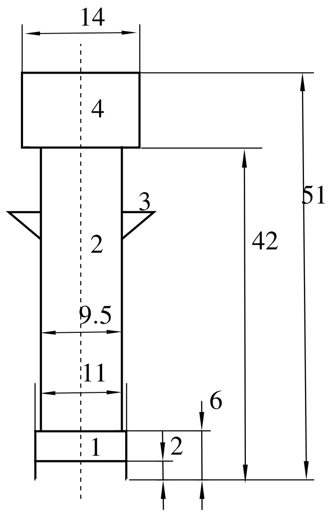


Fig. 2. Measurements of Service Robot (inches). This is only a rough schematic. More detailed schematics are provided as links at the end of the paper.

1) The Base: The base of the robot consists of two wooden squares, separated by a few inches (Figure 3). This provides a safe compartment for a laptop, and is mounted on the iRobot Create by using metal standoffs and a wooden block for stability. These standoffs provide a gap between the laptop and the Create that is big enough to allow a user to reach in and access the Create buttons and add/remove a power board or the serial interface of the Create. The compartment features small stands for the laptop to allow proper airflow underneath for cooling purposes, and the standoffs that separate the top and bottom of the compartment were placed specifically to allow laptops of various sizes to fit comfortably in the compartment. The design places the laptop in the base so that the laptop weight is close to the ground for stability reasons and to ensure that the laptop is protected if the robot is knocked over. The top of the compartment is attached directly to the body using wood screws.

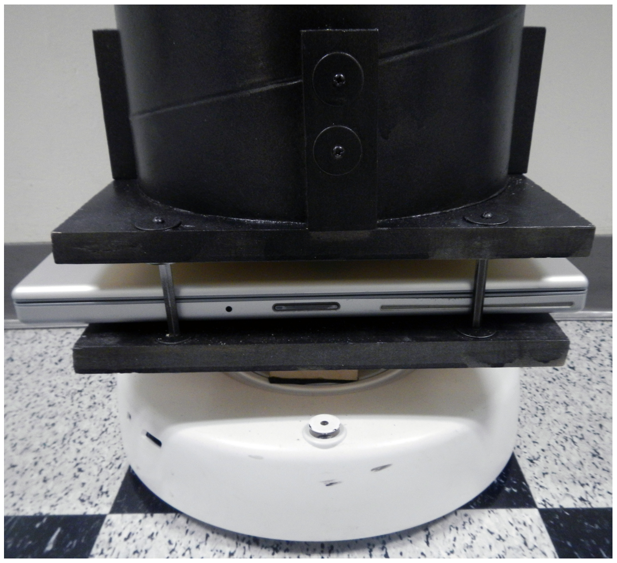


Fig. 3. The Base of the Service Robot. A laptop rests in a secure compartment on top of an iRobot Create. The body of the robot is securely attached to the top of the base.

2) The Body: The body was made from a reinforced cardboard tube, which is strong enough without incurring the cost of custom fabricated metal or plastic. Cardboard also offers more flexibility for later adaptations. The tube provides excellent resistance against low speed impact and provides enough vertical strength to comfortably support a large amount of hardware without any problems. The tube is hollow, allowing the wiring from the arms and head to connect with the laptop while safely encased inside the robot. Since the body has no sharp edges, it poses no threat to wiring inside or humans outside. The body is attached to the head, arms and base with wooden clamps consisting of two pieces of wood screwed together through the tube wall, pinching the cardboard and holding it in place. This method of attachment allows the cardboard to be firmly attached to any wooden surface in a strong and durable fashion. Additionally, the connection between the body and the base is reinforced with steel angle brackets to help alleviate the torque that will be put on that particular joint during normal operation.

3) The Arms: Arms can be mounted in any position on the body or head that is convenient for the environment in which the robot will be operating. The original design placed the arms slightly above counter height to allow easy manipulation of objects on tables and to give the robot a humanoid appearance. The arms are mounted on a horizontal wooden platform which is attached to the body using the wooden clamp method. They are attached to the platforms in a secure and unmoving fashion, which allows the arms to interact with the vision system after calibration. The arms are positioned symmetrically on either side of the body, but are mounted closer to the front to allow manipulation of objects in front of the robot and interaction between the arms if desired. The provided design does not place the arms close enough to each other that they could collide, but a modified design could allow the robot to hand things from one arm to another. The arms are connected to the laptop via wires that run through the body of the robot. This design uses the TurtleBot Arm, designed by Willow Garage, which was built in the lab using a 3D printer and parts ordered online. This arm is easy to build and provides accurate control, but any arm can be mounted on the robot as long as it can interact with the laptop. The TurtleBot arm uses an ArbotiX RoboController to interface between the servos and the laptop, and can be controlled from ROS using packages freely available from Willow Garage.

4) The Head: The head is a trapezoidal prism made of foamboard, which is primarily used by architects to build models. The head has a vertical back and a sloped front with room for an optional face. The head and face were designed to be simple and friendly, so that people would want to interact with the robot. The design was inspired by MIT's Boxie [9] and adapted to its final version after conducting a user study.

The foamboard head is attached to a wooden frame with a hook and loop fastener. This makes the head easily interchangeable, keeps the weight down, provides an easy way to access the interior of the robot in order to change or repair cables, and offers an easy and stable way to attach the Kinect or any other vision system to the head of the robot. Having a sloped head not only makes the robot look nicer, but also guarantees that the Kinect's field of vision is not blocked when looking down. The eyes of the robot are big circles with an LED mounted in the center of each, which allows the eyes to light up. Lighting the eyes is an simple way of making the robot's face look more expressive. In the provided packages, the eyes light up when the robot is using the Kinect for a vision task, when the robot is talking to a person, and when the robot is dancing. Through combinations of winking and blinking, they can be used to express emotions, making it easier for the user to engage with the robot.

B. OPERATING SYSTEMS

As with the provided hardware, the goal of the provided software was to create a robotic platform available to any researcher. To this end, a multi-platform approach was chosen. As a result, the service robot can run under either Linux/ROS or Windows.

1) Running on ROS: Because of the robot's configuration, many ROS packages will work smoothly on this robot. This provides an immediate wealth of functional packages and a large support community to developers using ROS. Modifications to some packages will be needed to utilize the second arm, and calibration will be needed to adjust to the new shape. All of the software required to use existing ROS packages can be found on the TurtleBot website [10], along with installation instructions. Additionally, any packages that use only the Create or the Kinect should work with no adjustments. Due to the nature of ROS, it should be easy to create new applications for this robot by building upon existing ROS packages.

2) Running on Windows:

a) Drivers and Libraries: Providing power to a vision system on a mobile robot is often a challenge. Although drivers and libraries exist in Windows for both Kinect and the iRobot Create, there was no existing support powering the Kinect from the Create. Therefore, a new library for the iRobot Create was developed to support powering the Kinect. Utilizing the Create's open interface [11], commands are sent to the powerboard via its cargo bay parallel connector. An additional library was written to allow manipulation of the TurtleBot Arms under Windows. This was achieved by interfacing through the Arbotix RoboController using serial commands.

b) Creating a Service Robot SDK: With the core libraries and drivers completed, the next step was to create an SDK for the service robot that integrated all the libraries. The SDK was designed to handle concurrency and coordination, to support multithreading, and to treat every component as a service. Such an approach conforms with Microsoft's Robotics Developer Studio Approach CCR and DSS frameworks, and allows the researcher to develop using a higher-level programming approach. With this setup, a developer can simply instantiate a robot object and add functionality without worrying about deadlocks and resource handling. This simplifes the coordination of multiple tasks and provides a global "kill switch" for the robot. Furthermore, since it utilizes the Kinect SDK, the service robot SDK supports control through predetermined voice commands [12]. The SDK was built using C#, so the resulting library is accessible to any programming language of the .NET family.

c) Interfacing and communicating with the service robot: The service robot can be programmed to run autonomously under either operating system. In addition, a system was added to the SDK which allows the secure exposure of its services. This allows data exchange with an external system, providing the basis for remote control. The system was built using Windows Communication Foundation (WCF), a technology based on a service oriented architecture that supports SOAP, AJAX and REST messages. This approach allows universal interaction with the service robot, independent of platform or programming language. This allows the service robot to integrate with other research projects and to communicate with computers, tablets, smartphones and websites of any kind. Because it is running a web application server, the SDK provides the framework to expose any added features with very little additional coding. This system allows the service robot to act autonomously, and to establish a two-way link with any number of external devices. This can be used to command the robot remotely, or to allow the robot to control other devices.

III. HUMAN-ROBOT INTERACTION POSSIBILITIES

The robot was designed to interact with humans in multiple ways (Figure 4). Applications were developed to showcase some of these modes of interaction, including graphical user interfaces, voice control, remote control via mobile devices, and direct control via brain-computer interfaces. Windows was chosen primarily in order to obtain access to Microsoft Kinect libraries. This section will present all the human robot applications which have been implemented. Section V will discuss incorporating SLAM and grasping libraries.

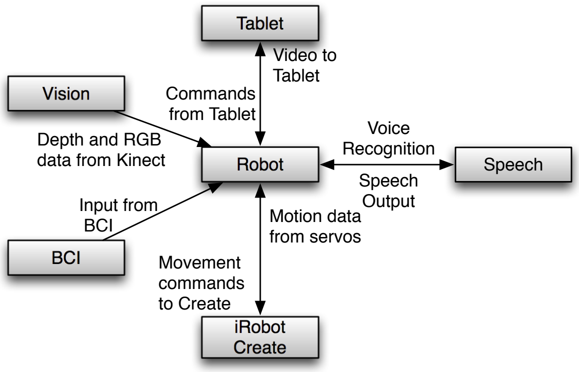


Fig. 4. The I/O Architecture of the Robot. The Robot acts as a hub between many different input sources, and reacts using event handling.

A. Voice Control

The robot comes with basic preprogrammed voice control using the Kinect SDK voice recognition system. The decision was made to use only preprogrammed commands so as to limit the set of possible matches. This allows the robot to obtain better results with accented or poorly enunciated speech. In a user study, people with different native tongues were invited to give speech commands to the robot, and the robot was able to successfully interpret the vast majority of commands. Using this approach allows developers to associate voice commands with each application, which then can be used successfully by many different people without any training. This system performs with a much better success rate than a universal speech recognition system, since it can discard any input that does not match a known command.

Furthermore, the robot is capable of performing speaker recognition once trained (Figure 5). The robot can be trained by saying a specific phrase five to ten times. The frequency domain of the speech signal will then be smoothed and sampled. The system only samples every fourth value, enabling fast processing without compromising the result. The data is then added to a database, along with the phrase and a username, to enable future identification of the speaker. A normalized cross-correlation [13] and the difference between the aligned areas (squared-difference) [14] are used as similarity features. To recognize the speaker, a simple heuristic is used. It either results in confident recognition, partial recognition or no recognition. This allows it to react by greeting the user, attempting to verify the user's identity, or introducing itself to a new user (respectively). This algorithm allows the robot to keep track of users, and treat each with a unique and customizable manner.

The voice system capabilities were used to interpret specific motion commands ("Turn Right"), to initiate a follow algorithm, and in an introduction algorithm. When asked who he is, the robot will offer a handshake and introduce itself to the new person.

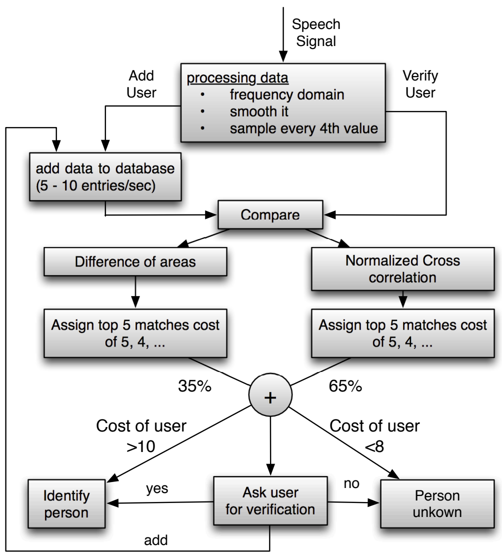


Fig. 5. Voice Control training algorithm. The voice control system is one of the core methods of interacting with the robot. The ability to recognize a user's voice and respond in an individual fashion greatly helps improve user experience.

B. Following a User

The follow algorithm uses skeleton tracking to follow a user around (Figure 6). The robot reacts to voice commands and keeps a certain distance from a user depending on which person gave the voice command. A processing rate of 10 frames per second was chosen to guarantee smooth motion while not dominating the computational resources. For this algorithm the acceleration and top speed were set at values that would allow smooth following without instability or sluggishness. These values can be seen in section IV. The tolerances were chosen to emphasize smooth robot motion and to avoid small corrections. The algorithm uses multi-threading to enable the robot to process other commands while it is following a user.

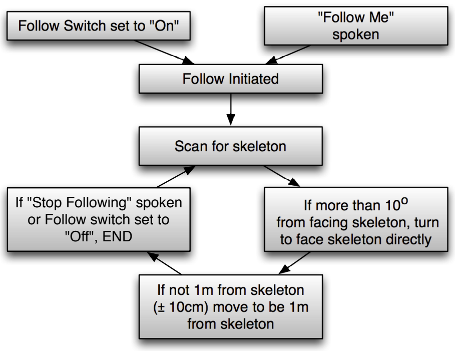

Follow Switch set to "On"
"Follow Me" spoken
Follow Initiated
Scan for skeleton
If "Stop Following" spoken or Follow switch set to "Off", END
If more than 10° from facing skeleton, turn to face skeleton directly
If not 1m from skeleton (± 10cm) move to be 1m from skeleton

Fig. 6. The Follow Algorithm allows the robot to follow a user, remaining one meter away at all times. This is useful for keeping the robot close at hand while changing areas.

C. Control through a BCI device

Brain-Computer Interfaces for human robot interaction [15] [16] [17] [18] [19] are becoming increasingly popular. Consumer devices such as the Emotiv EPOC [20] and the Neurosky MindSet [21] are revolutionizing the market by enabling control of a robot via EMG and EEG signals. Because the robot proposed in this paper uses a service-oriented architecture, it supports control from any BCI devices which can execute commands. Using an Emotiv BCI EEG headset, an application was built to control the robot through EMG signals. The headset can detect when a user performs different facial expressions, such as smiling, clenching their teeth, or raising their eyebrows. These motions can be mapped to different operations which the service robot can execute (Figure 7). The benefits of using EMG signals are that no prior training is required and that the signals are strong enough to provide high accuracy in the command execution.

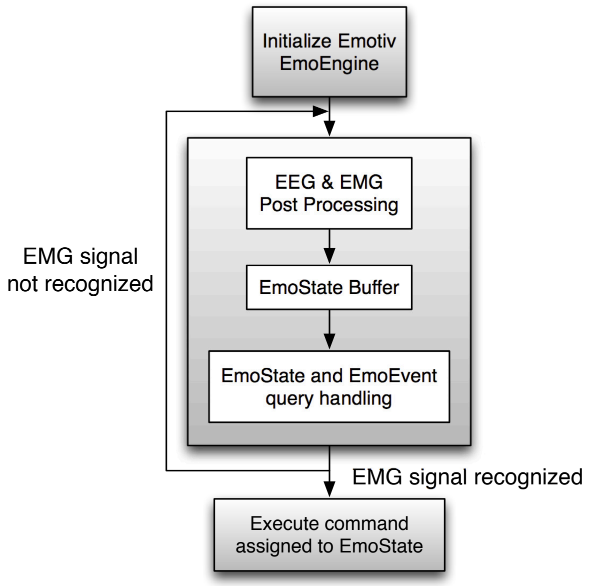

Initialize Emotiv EmoEngine
EEG & EMG Post Processing
EmoState Buffer
EmoState and EmoEvent query handling
EMG signal not recognized
EMG signal recognized
Execute command assigned to EmoState

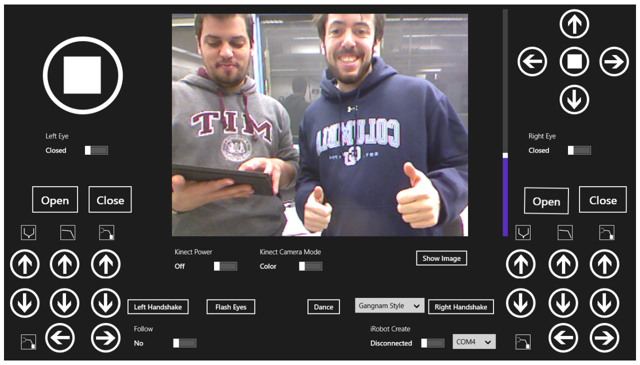
Fig. 7. The BCI interaction allows the robot to interact with disabled persons who are not able to speak or use handheld devices.

D. Remote Control

Another way of interacting with the robot is through remote control. The remote control was tested with a Microsoft Surface tablet; the multi-touch interface allows intuitive and precise control. The use of a web application server with web services as SOAP, AJAX, or REST messages also allows control from an iOS, Windows Phone or Android device. This significantly expands the developer audience for the service robot and provides a more abstract layer of communication.

Since the service robot can expose its services securely, a user can interact with the robot during the performance of autonomous actions. This allows the user to issue an override command when direct control is required. This is very useful for situations where the robot may become trapped or is operating near unaware human beings.

To showcase an application utilizing all of the software components, a teleoperation application was built targeting the Microsoft Surface tablet running on an ARM computer architecture. The user interface developed (Figure 8) was designed with regards to Nielsen's 10 usability heuristics [22] and Microsoft's design guidelines, providing an easy-to-learn interface with a native look. The chosen device orientation was landscape, as it positioned the Kinect video and depth stream in the center of the user's view while placing the controls on the edges for easier touch interaction. While under development, the tablet was given to 27 people with different backgrounds and ages for testing. Their feedback helped to make the interface as intuitive and easy to use as possible. The tablet was successfully used to navigate the robot around the campus of Columbia University and to accomplish simple grasping tasks, such as getting small objects from the office next door.



Left Eye: Closed
Right Eye: Closed
Kinect Power: Off
Kinect Camera Mode: Color
Gangnam Style
Follow: No
iRobot Create: Disconnected
COM4

Fig. 8. The Tablet Interface allows convenient remote control of the robot, allows the robot to act as a telepresence device, and acts as a manual override if required.

The remote control application can be used both as a way of testing the hardware and its limitations, and as a programming example of interaction with the robot's web services. Furthermore, this provides a convenient way to control the robot for a demo or use it as a telepresence device.

These ways of interacting with the robot can be used as a starting point for many other projects discussed in Section V.

IV. RESULTS

Using this hardware, tests of the robot's limitations produced the results in Table II.

Top speed was measured on level ground, and the maximum angles of stable ascent and descent were measured on a ramp. It is highly recommended to reduce the top speed of the robot before driving on a ramp with a slope close to the maximal value. Maximum acceleration and deceleration represent the highest values which do not jeopardize robot stability. To maintain optimal stability with minimal oscillation of the body, it is recommended that the robot only be used at or below the optimal acceleration proposed in this paper. These values are made possible by minimizing the weight of the body and putting most of the weight in the lower part of the robot. This lowered the center of mass and reduced the torque that can cause the robot to fall over.

Top Speed	1 m/s
Maximum Angle of Ascent	5 degrees
Maximum Angle of Descent	5 degrees
Maximum Acceleration	0.5 m/s^2
Maximum Deceleration	1 m/s^2
Optimal Acceleration	0.25 m/s^2

TABLE II
MOBILITY LIMITS

Next, the capabilities of the arms were tested, which produced the results shown in Table III.

Maximum Lifting Capacity	500 grams
Response Time	0.25 seconds
Time to adjust position	1 second (fast) 2 seconds (smooth)

TABLE III
ARM LIMITS

The arms are configured so that the servos can be controlled independently or simultaneously. This allows easy control for a user who wants to issue single commands, such as "Left Elbow Up," but also gives fast response time to automated algorithms which can move all the servos at once. This allows the arm to quickly reach a specific configuration.

When the provided software is used, the robot will accelerate and decelerate at a rate that is fast enough to allow useful functionality, but will not operate in an unstable manner. Furthermore, the software allows smooth motion by controlling its acceleration autonomously, which can be useful when a user gives commands to the robot.

Through all of these actions, the robot limits itself to stable motions, and prevents motion of the arms which would collide with the robot.

V. DISCUSSION

The robot we describe excels in affordability, modularity, ease of construction, and ability to accomplish simple tasks. The proposed robot does not purport to compete with the excellent hardware or capabilities of expensive modern robots, such as the Asimo, Hubo or PR2, but testing has shown that this robot can be used and adapted to perform many useful tasks. This robot fills an important niche as a low-cost, full-size research platform capable of incorporating different hardware and software that specific users may require. This robot can be duplicated exactly to provide an adult-sized robot with accurate vision and dual arms which is capable of performing basic tasks and conducting basic experiments.

One possible use of the robot is as a telepresence device, controlled by a computer or tablet from anywhere with

Internet connectivity. By adding a simple monitor, the service robot can perform two-way video and audio streaming. This allows the user to see and speak to anyone in the vicinity of the robot. Implementing one of the many available SLAM algorithms can make the robot more autonomous, enabling navigation to any known location. With the built-in remote grasping capabilities, this service robot could be used for a variety of tasks, such as retrieving objects for disabled users.

Future work could include implementing an object recognition and grasping algorithm. This would allow the robot to fulfill simple tasks in the household, such as finding glasses and bringing them to the kitchen sink. It is important that the lifting and grasping capabilities of the arms are taken into consideration when selecting objects to grasp.

Because of the modularity of the software, existing packages can be imported and built upon to produce desired behavior. The robot can also be used in any number of fields, such as the diagnosis and therapy of social deficits[23]. Since the robot costs under $1,500 USD to produce and does not require any significant tools other than a saw and drill to manufacture, this platform will make these previously expensive capabilities accessible to small labs and hobby roboticists.

VI. CONCLUSION

This paper shows a low-cost robot that is able to accomplish a large variety of different tasks in a real environment. This paper provides implementations of algorithms to interact with the robot via voice or remote control, offers a wide variety of ready-to-go ROS packages for a number of functions, and proposes different applications that can and will be implemented in the future. Due to low price, simple manufacturing, and platform versatility, this design gives research labs and interested persons the possibility of building their own robot for immediate use or further development. We hope to motivate people from different fields and with different interests to use our guidelines, to adapt them for their own purposes, and to bring service robots into our daily lives.

VII. RESOURCES

Detailed schematics and building instructions can be found at www.auxy-robot.com/hardware .

Software Packages and installation instructions can be found at www.auxy-robot.com/software .

VIII. ACKNOWLEDGEMENTS

The authors would like to thank Professor Peter Allen for his support and funding. Additonal thanks to Dr. Austin Reiter and Dr. Hao Dang for their valuable advice in this endeavor.

REFERENCES

[1] Bennet Brumson. New applications for mobile robots. `http://www.robotics.org/content-detail.cfm/Industrial-Robotics-Feature-Article/New-Applications-for-Mobile-Robots/content_id/3362`, April 2012.

[2] The Robot Report. Service robots for personal and private use directory. `http://www.therobotreport.com/index.php/service_robots_personal_private_use`, 2013.

[3] P. McGuire, J. Fritsch, J.J. Steil, F. Rothling, G.A. Fink, S. Wachsmuth, G. Sagerer, and H. Ritter. Multi-modal human-machine communication for instructing robot grasping tasks. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 2, pages 1082–1088 vol.2.

[4] M. Van den Bergh, D. Carton, R. de Nijs, N. Mitsou, C. Landsiedel, K. Kuehnlenz, D. Wollherr, L. Van Gool, and M. Buss. Real-time 3d hand gesture interaction with a robot for understanding directions from humans. In *RO-MAN, 2011 IEEE*, pages 357–362, 31 2011-Aug. 3.

[5] G.R. Bradski and V. Pisarevsky. Intel's computer vision library: applications in calibration, stereo segmentation, tracking, gesture, face and object recognition. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 796–797 vol.2.

[6] Stefan Waldherr, Roseli Romero, and Sebastian Thrun. A gesture based interface for human-robot interaction. *Autonomous Robots*, 9:151–173, 2000.

[7] Richard A Newcombe, Andrew J Davison, Shahram Izadi, Pushmeet Kohli, Otmar Hilliges, Jamie Shotton, David Molyneaux, Steve Hodges, David Kim, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*, pages 127–136. IEEE, 2011.

[8] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM, 2011.

[9] Alexander James Reben. Interactive Physical Agents for Story Gathering. Master's thesis, Massachusetts Institute of Technology, Cambridge, 2010.

[10] Willow Garage. Turtlebot. `http://ros.org/wiki/Robots/TurtleBot`, 2013.

[11] iRobot. Open interface manual for irobot create. `http://www.irobot.com/filelibrary/pdfs/hrd/create/Create%20Open%20Interface_v2.pdf`.

[12] Zhengyou Zhang. Microsoft kinect sensor and its effect. *MultiMedia, IEEE*, 19(2):4–10, Feb.

[13] Hamici Zoubir. Speaker recognition using spectral cross-correlation: A fast algorithm. *Journal of Computer Science*, (Special Issue): 84-88.

[14] Beigi Homayoon. *Fundamentals of Speaker Recognition*. Springer, New York, 1st. ed. edition, 2005.

[15] Christian J Bell, Pradeep Shenoy, Rawichote Chalodhorn, and Rajesh PN Rao. Control of a humanoid robot by a noninvasive brain–computer interface in humans. *Journal of Neural Engineering*, 5(2):214, 2008.

[16] Jörn Vogel, Sami Haddadin, John D Simeral, Sergej D Stavisky, Dirk Bacher, Leigh R Hochberg, John P Donoghue, and Patrick van der Smagt. Continuous control of the dlr light-weight robot iii by a human with tetraplegia using the braingate2 neural interface system. In *International Symposium on Experimental Robotics (ISER2010), Dehli, India*, 2010.

[17] J Weisz, B Shababo, L Dong, and P Allen. Grasping with your face. In *13th International Symposium on Experimental Robotics (ISER2012), Berlin, Germany*, 2012.

[18] Yongwook Chae, Jaeseung Jeong, and Sungho Jo. Toward brain-actuated humanoid robots: Asynchronous direct control using an eeg-based bci.

[19] Dennis J Mcfarland and Jonathan R Wolpaw. Brain–computer interfaces for the operation of robotic and prosthetic devices. *Advances in Computers*, 79:169–187, 2010.

[20] Emotiv. Epoc. `http://www.emotiv.com`.

[21] Neurosky. Mindset. `http://www.neurosky.com/Products/MindSet.aspx`.

[22] J.Nielsen. 10 usability heuristics. `http://www.nngroup.com/articles/ten-usability-heuristics/`.

[23] M. Mataric B. Scassellati, H. Admoni. Robots for use in autism research. *Annual Review of Biomedical Engineering*, 14.