Nelson: A low-cost social robot for research and education

Article · January 2011

DOI: 10.1145/1953163.1953230

Nick John Webb
Union College

Tomek Strzalkowski
University at Albany, The State University of New York

Nelson: A Low-Cost Social Robot for Research and Education

Michael Ferguson
ILS Institute, SUNY Albany
1400 Washington Avenue
Albany, New York, USA
ferguson@cs.albany.edu

Nick Webb
ILS Institute, SUNY Albany
1400 Washington Avenue
Albany, New York, USA
nwebb@albany.edu

Tomek Strzalkowski
ILS Institute, SUNY Albany
1400 Washington Avenue
Albany, New York, USA
tomek@albany.edu

ABSTRACT

A social robot is a robotic platform that supports natural interaction with people in a human-scale environment. Such a platform allows interesting opportunities for both traditional Computer Science students and students from other disciplines, such as psychology, philosophy, design and communications. In this paper, we describe a new social robotic platform for educational uses that is equipped with a social face, arms for gesturing, advanced sensory, mobile base, and ROS integration. By using off-the-shelf and rapidly prototyped components, together with open source software, this platform is low-cost, easy to use, and easy to reproduce.

Categories and Subject Descriptors

K3.2 [**Computer and Information Science Education**]: Computer science education; I.2.9 [**Robotics**]: Commercial robots and applications.

General Terms

Algorithms, Design, Experimentation

Keywords

Robotics, ROS, Social Robotics

1. INTRODUCTION

While a number of schools and classrooms are exploring the use of robotic platforms, the focus so far has been on providing students with platforms to explore core programming skills, or the fundamentals of robotics or cognitive science (cf. [3, 7, 5, 11]).

As interest in robots and robotics increases, there is a parallel interest in the *social* component of robotics. The goal for us is to create robots that can work alongside, and collaborate with, humans in real world environments. In order to achieve this successfully, such robots need to be able to interpret and produce a wide range of social functions. This includes principles of human-robot interaction, communication, philosophy, psychology and design.

To explore such human-robot interaction issues, research groups typically create their own 'social robot', robots which have some easily recognizable social characteristics, such as a face which can produce expressions. The drawback with such platforms is that they are often expensive, and the designs are not open source.

With Nelson, our goal is to create a socially capable robotic platform, allowing students to study issues such as human-robot interaction or the communicative requirements for a robot to be deemed socially acceptable. For instance, students might study the social cues that a robot could generate (robot specific communication research) or recognize how the robot's navigation impacts people's impression of it (which we refer to as socially-aware navigation).

In developing Nelson, we set forward a series of design requirements that would be necessary to adequately support the social robotics program we envision. First and foremost, both hardware and software should be open source where possible, and easily modifiable to allow students to expand the platform as they develop new experiments.

Second, the platform should be low cost, so that it is accessible to a wide audience. We used off-the-shelf components whenever possible.

Third, we wanted a platform that would be socially acceptable to a large audience, and so we selected a small number of parameters we believed would fulfill this requirement; for example, we created a robot that was closer to human size, but not so large as to be intimidating, and we required that the robot have some sort of face.

Finally, because we believe that social robots must interact with people in human-scale environments, the robot needed robust sensory to deal with the complex environment of the real world. Additionally, because of the robust sensory, the robot should carry at least its basic processing power onboard.

For those parts which could not be obtained off-the-shelf, we tried to design parts which could easily be manufactured or replicated using known processes, such as 3D printing, laser cutting, or sheet metal fabrication. As different sites would have access to different methods of fabrication, each of Nelson's parts were designed to be made in several ways.

2. RELATED WORK

When talking about robotic platforms in an educational environment, we have to distinguish between hardware platforms and software platforms. The choice of both of these

Figure 1: Nelson, a social robot.

elements has a direct correlation with the usability, cost and extensibility of the project in question, and must be considered carefully.

From an educational perspective, there are a number of robot hardware platforms available, each aimed at a particular facet of the educational market. For example, the Scribbler robot[1] is designed as a low cost entry level robot, used by many to teach both fundamental robotics, and as a platform for introducing programming to a wide audience. The LEGO NXT[2] has many of the same principles, but with a larger array of sensing. The iRobot Create[3] is a robust, industrially produced platform with a more limited sensor array but which has a high payload capacity and can easily be extended.

Recently, Aldebaran robotics has produced the Nao robot[4], a humanoid replacement for the now-defunct SONY Aibo in robotic soccer competitions. These platforms are more socially capable than the Scribbler, Create or NXT, but are still fairly small, and very expensive.

The CMU Chiara[5], is a multi-legged open source platform, which is a more capable research platform. However, mobility with the Chiara is still limited due to battery life and weight, and it is primarily designed as a platform to research issues in cognitive science and machine vision.

The RobotCub project[6] is one of the most exciting robotics projects in Europe. Tasked to design and build a robotic platform for future research and education, the consortium has built iCub - at 37" high, the iCub is equivalent in size to a three year old child. It has 53 degrees of freedom, has fully working, grasping hands, as well as visual, auditory and haptic sensory capabilities. The entire design of the iCub, both hardware and software, is freely available under the Free Software Foundation GNU license, however a complete, assembled iCub costs around $200,000.

From a software perspective, there are several packages which attempt to abstract robotic concepts, in a way that makes interfacing with a range of hardware possible. This is attractive on at least two levels. First, students can switch platforms, based on experimental need, e.g. requiring a specific set of sensors or effectors. Second, a user community can create libraries of useful packages and plugins, which can then be exploited or expanded by students, relieving the need to reinvent the wheel at each iteration.

Two of the most successful open-source robot platforms currently available for education are Myro and Tekkotsu. Myro is a Python-based programming environment for introducing programming in a unique and personal way [2]. The software uses the Parallax Scribbler robot with a custom add on board that communicates with as student's laptop over a wireless connection. The Scribbler robot has limited sensory, which includes several floor sensors and a few IR obstacle avoidance sensors. The add-on board has a small camera, however the frame rate is limited. The Tekkotsu system is designed for higher level robotics education, with extensive support for cognitive vision, walking robots, and kinematics [12]. Students use C++ and a custom state machine language to create Tekkotsu programs. Tekkotsu supports only a small set of hardware, currently the iRobot Create, Dynamixel servos, webcams and several custom walking robots.

While each of these platforms offer support for their intended audience, we deemed that neither was the preferred platform for a new robot intended for social robotics as neither addressed the issues of robots operating in real human-scale environments. Instead, we chose to build on the Robot Operating System (ROS), a meta-operating system for robotics [10]. ROS is in use at over 30 institutions, with support for over 25 robotic platforms ranging from the Lego NXT up to advanced mobile manipulators such as Willow Garage's PR2 robot[7]. While its robustness, ease of use, and real-world focus has gained ROS a foothold in research in both University and Industry settings, ROS has not been widely applied to educational settings.

The positive here is that there is a large amount of open-source software available under ROS for interfacing with hardware such as scanning laser range finders and stereo cameras. There is also a large amount of software for advanced tasks such as navigation in human-scale environments. ROS allows the student to choose between several supported programming languages. ROS has an extensive, community-generated wiki, which also has a number of tutorials. In choosing ROS for Nelson's software platform, we accepted that generating some custom code to make ROS more accessible for students was worth the resultant speed up in development time for creating such a platform.

3. INTRODUCING NELSON: HARDWARE

We specifically wanted to design a robot with social capability. For us, this means that it should have a face (and implied through this, a voice, although this is not covered

[1] http://www.parallax.com
[2] http://mindstorms.lego.com
[3] http://www.irobot.com/create
[4] http://www.aldebaran-robotics.com/en/
[5] http://www.chiara-robot.org
[6] http://www.robotcub.org
[7] http://www.willowgarage.com

in this paper), and arms, enough to create expressions and gestures, as well as pick up and put down small objects. Also, it would need sufficient sensory ability to navigate and negotiate small populated environments, such as our labs or office suites. In this section, we detail our approach to these requirements.

3.1 A Low-Cost Face for HRI

One of the most important elements of this robot is the face and head. Some of the more interesting robotic faces, such as that found on the Mertz robot[1] can cost thousands of dollars to build, well beyond the scope of our budget. Instead, we designed a face that can be constructed for around $150 using either basic hand tools, or a 3D printer.

Nelson's head consists of a 3 degree of freedom neck, and an 7-degree of freedom face. The neck is built out of Dynamixel AX-12 servos, a low cost, strong, and robust actuator that includes easy to assemble brackets. Of the three neck servos, one is devoted to panning the head left or right. The remaining two servos are used to move the head up or down, and forward or backward. This simple neck allows a wide range of motion, and Nelson can appear to be inquisitive by leaning forward towards an item, or afraid by pulling his head back. The range of motion of his head is such that he can easily view people standing a reasonable distance (around 3 feet) in front of him.

The face servos are moving lightweight components and are therefore barely under load allowing us to use tiny micro servos which cost only a few dollars a piece. A number of very small parts had to be produced, however all are designed to be produced on a low-cost 3D printer[8]. Alternate part designs are available that can be created using a laser cutter, or hand cut from sheet material using other tools. The servos are installed such that each eye can pan independently, and a tray holding the eyes can tilt them up and down as a pair. A pair of eyebrows can each be moved up or down independently, and each lip can move up or down independently. This small number of movements can generate a vast array of possible facial expressions, as shown in Figure 2. One actuator that was planned but not yet implemented activates a set of eyelids. Eyelids would likely improve the range of emotions that Nelson can express, however our early approaches were fragile.

As we intended Nelson to be socially acceptable to a wide audience, we paid special attention to his level of realism. In developing Nelson's face, we attempted to keep the features cartoon-like, so as to avoid the uncanny valley effect[4]. Unfortunately, as the robot came together and the software became more sophisticated, we received feedback that we were rapidly approaching that point. Future versions of Nelson may heed some of the other recommendations from[4], such as making a robotic face wider than it is tall so as to make the robot appear less human. Such approaches have worked well on robots such as Willow Garage's PR2, but the results are unclear for robot's with active faces.

3.2 Torso and Arms

We believe that a key requirement for continued interaction is that a robot be closer to human scale. While a tall robot is more susceptible to toppling over, we have designed Nelson with a lightweight upper body and torso, keeping heavy components such as batteries and computers close to the ground. Nelson's torso raises his head to more reasonable height, close to that of a child, and approximately the same size as the iCub robot. This is also an important consideration for sensors, as a camera mounted too close to the ground will wash out when attempting to look up at a person.

A pair of simple arms are attached to the torso. These arms are primarily to aid in gesturing, such as pointing at objects, and conveying expressions although they can be used for simple object manipulation. Each arm includes three servos for moving the arm and an additional servo to create a small gripper. The arms, like the neck, are constructed out of Dynamixel servos, making them inexpensive yet robust.

[8]http://www.makerbot.com

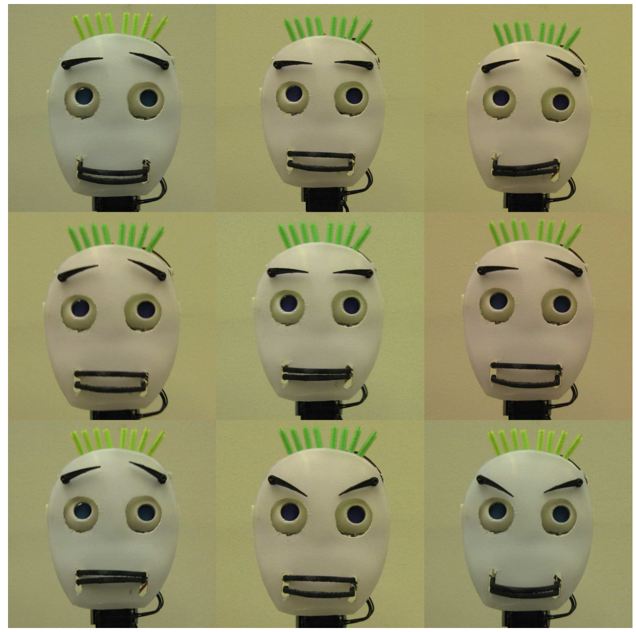

Figure 2: The many faces of Nelson.

3.3 Mobility

Because social robots must interact with people in human-scale environments, Nelson has a mobile base, capable of achieving approximately a walking speed. While there are a number of mobile bases with support for ROS, an iRobot Create base was chosen, due to its low-cost, robust operation, and hefty payload capability. The Create is also already widely used in existing educational programs, making it a logical choice.

A custom column mount is built on top of the Create, to which the torso and head are attached. The column mount is built out of low cost electrical conduit, a locally available and inexpensive alternative to the more common but costly aluminum channel found on higher end robots.

3.4 Sensors

Operation in a complex environment requires advanced sensory capabilities. Nelson is currently outfitted with a single webcam above his face, and a unique scanning range

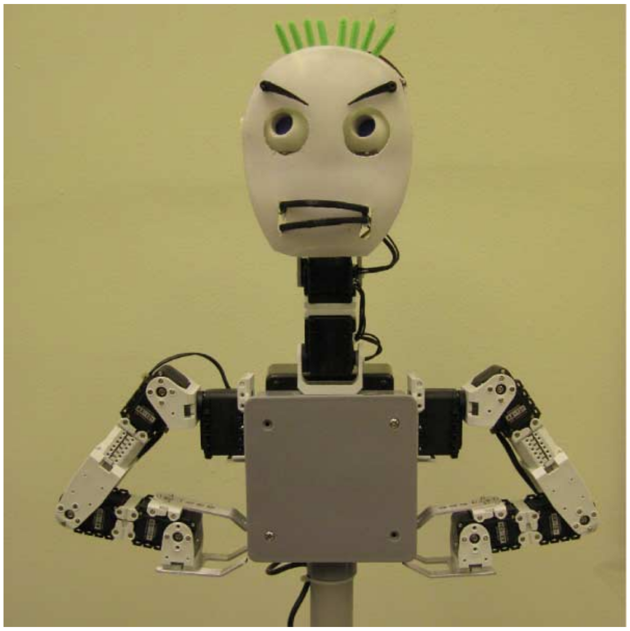

Figure 3: View of the Nelson's arm and chest components.

finder. As ROS is fully integrated with OpenCV[9], creating visual routines is a relatively simple task.

Navigating in human-scale spaces requires that a robot avoid obstacles, especially dynamic ones. Typically, this is done using a laser range finder, however such devices are extremely expensive, typically several thousand dollars. Nelson, uses a lower cost replacement constructed from a $20 infrared range finder and a Dynamixel AX-12, which we call the Planar Meta-Laser (PML). The PML pans the servo, taking readings with the range finder and broadcasting the data in the same format as laser range finders. By broadcasting the data in the same format, we can use existing tools, such as ROS's Navigation Stack [8]. Further details on the usage of the PML are provided later in this paper.

3.5 Controller

All of the servos are connected to a Netbook using a Vanadium Labs ServoStiK. The ServoStiK is a USB interface device that passes serial commands between the PC and the servos, in addition to controlling the PML.

4. INTRODUCING NELSON: SOFTWARE

Having constructed the hardware, we need to implement software for robot control. Building Nelson's software on ROS means that we have easy access to a wide variety of open-source libraries and tools. Reducing the amount of custom code needed to run Nelson makes the platform both more robust and maintainable.

ROS currently supports Python, C++, and Lisp, and there is experimental support for Java and other languages. The ability to allow students a wide choice of languages means that students spend their time developing interesting robotics applications and learning about robotics, rather than debugging the nuances of a language they are not entirely familiar with.

ROS includes a federated model of code distribution, using a single, publicly accessible wiki hosted at www.ros.org, but allowing sites to host their own code repositories. Software is organized into packages, small portions of code that do a specific function, and stacks, which are a collection of related packages. A good example is the *Navigation* stack which provides autonomous navigation capabilities to a mobile base. The stack includes packages such as *costmap 2d* which handles the creation of 2D occupancy grids from sensor data, a separate *base local planner* package then uses this 'costmap' to safely send trajectory commands to the mobile base. ROS uses a distributed computing approach to robots, so each package typically consists of one or more programs that become separate nodes in the runtime graph. This distributed approach, and a large collection of open source software, means that students can often prototype a high level idea in a very small amount of code.

The distributed approach of ROS, and the easy division and communication between different nodes within the runtime graph also promotes collaboration. Robotics is a diverse field, and we expect that each student will choose to focus on one aspect of a social robotic application. ROS makes it easier then for teams of students to collaborate in this manner.

4.1 Servo Control

Vanadium Labs provides an ROS library for interfacing to the Dynamixel servos using a ServoStiK. On top of this library we have built a series of controllers and tools which provide easy access to the face servos and arms.

4.2 Virtual Face

In addition to an embodied version of Nelson's face, we have created a virtual version that interfaces with ROS. The GUI version of Nelson interacts with ROS in exactly the same way as the real face, allowing students to test code in a simulated environment.

5. A COURSE IN SOCIAL ROBOTICS

One principal goal with this platform is to introduce the issues of Human-Robot Interaction to a wide audience, including both students in traditional Computer Science and inviting participation from students from other disciplines. A first iteration of this combined class happened in the spring semester of 2010.

In addition to introducing the technical topics involved with using ROS, and situating our overall human-robot interaction objectives within the field, a major portion of the course is a final social robotics application project. For this final project, CS and non-CS students form teams to create an application, guided by both the technical reality of the robotic hardware and software capabilities and the understanding of social issues in creating such robots. For the first iteration, we had students with psychology and communication majors.

Planning for this application begins with round-table discussions between the two sets of students, with literature reviews on, for example, the importance of social cues, being presented by non-CS students, arriving at a sub-set of behaviors and actions the final robot should perform in order to be socially successful or accepted. Ultimately, the

[9]http://opencv.willowgarage.com

majority of the programming is then left to the CS students, although non-programmers are encouraged to write simple programs for Nelson using the easy to learn Python language.

Our intent is that all students leave a course in Social Robotics with the following:

- A working knowledge of the major issues confronting Human-Robot Interaction and related fields.
- An understanding of the kind of technology currently available for platforms that can operate in human environments. In particular, a solid understanding of the current sensor suites, and their limitations.
- Experience working in a truly interdisciplinary team, especially in the area of creating software requirements when not all members of the team are expert programmers.

5.1 Example Social Robotics Projects

Our social robotic platform provides a number of opportunities for students in both traditional Computer Science and other disciplines to explore interesting topics in Human-Robot Interaction. The next few sections of this paper outline several example projects that have been implemented using this platform, which give a feeling for the wide range of topics that can be explored.

5.1.1 People Detection and Following

People following is an interesting application and active research area which combines mathematical techniques, sensory and perception, and the understanding of human interaction. Robust algorithms for people following typically build on both or either of visual techniques or laser-based detection of legs. Such algorithms also require the use of filters on the candidate points that may be people. Students of Social Sciences can provide valuable input and feedback in this area in how people move and the size and effect of social space (personal versus public space), thus improving such filters.

Nelson's standard applications include a simple person detector which was implemented using the base scanning sensor to find possible candidates, and then a color tracking filter to verify the candidate, such as found in [6]. Even with great sensory, robotic applications can break down in complex environments. In particular, the presence of many people in an environment can cause people tracking to fail. The "Walk With Me" application allows a person to lead Nelson around by the hand, helping him to move in, through, and away from large crowds.

5.1.2 Navigation in Human-Scale Environments

As robots move out of the lab and into the real world, there is an immediate requirement for autonomous robust navigation. Paramount is that the robot be able to deal with obstacles, especially dynamic ones. Since the Planar Meta-Laser (PML) broadcasts its data in the same format as a much more expensive scanning laser, we are able to use ROS's standard costmap modules for local planning and obstacle avoidance.

However, the PML does not provide enough high quality data to construct area maps or to localize the robot. As a semester project several students created a localization

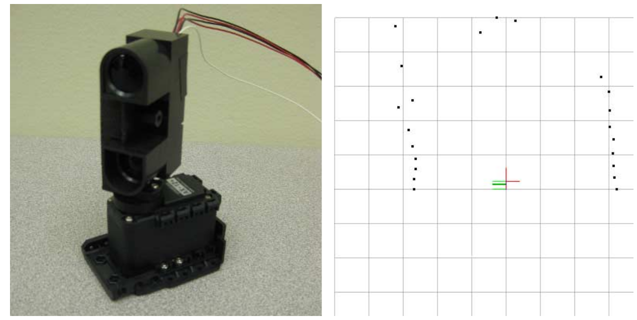

Figure 4: The PML, and a planar scan created with it.

routine to supplement the existing ROS navigation software. This localization system uses markers from ARToolkit, placed around the lab. Upon seeing a marker the robot can automatically align itself within a larger map. Students have already applied social robotics constructs in deciding how fast the robot should move, so as not to be a danger to humans in the navigation space, and attempting to make the navigation more fluid and natural, ideally observing social conventions, such as moving to one side in a corridor.

Future work on this low cost navigation system might include refining the global planner, creating a more ROS-like map server, and expanding the local planner to be more people friendly.

5.1.3 Non-Verbal Communications

Several communications majors who participated in Nelson's first deployment immediately wished to address the issue of non-verbal communications. An 'emotional engine' was created for Nelson using emotional dichotomies based on the Circumplex Model of Affect [9]. The emotional engine allows Nelson to automatically react based on his level of stimulation, and his belief of how accurately he was accomplishing his task, base on metrics such as the number of failed attempts at speech recognition or the tone of a person's voice. A graphical-based controller was also implemented to control Nelson's emotional state directly.

Future work in this area could include addressing issues such as response time and eye contact, as well as developing additional metrics for more robust feedback to the emotional engine.

5.2 Discussion

It is hard to formally evaluate the outcome of a single course, held in a single semester. However, it is the impression of the educators involved, and the anecdotal feedback of the student participants, that the course was a great success. The initial round table process produced fascinating interaction between groups of students who would not normally have cause to interact during the course of their academic studies. Students from outside of Computer Science gained access to, and direct experience with, complex computer-based systems, in a way usually out of reach to all but those that major in a computer or engineering field. The course will be offered again in the Spring of 2011, and Nelson will be a major component as the focus for social robotic applications.

6. FUTURE WORK

From a hardware standpoint, while Nelson provides an excellent set of capabilities, there is still more we would like to implement and refine. First and foremost is reducing the number of parts required for the face, and making parts simpler to produce.

We would also like to improve Nelson's sensory. A redesign of the head to better accept a pair of cameras would allow the creation of a low-resolution 3D depth map, which could be very useful for supplementing the PML during navigation, as well as improving people perception. Reduction in the noise of the PML's servo motor is also a top priority, as the noise is currently a distraction when interacting with people. Our current solution has been to throttle back the rate of scanning during interactions, so that the servo becomes quieter.

7. CONCLUSIONS

We have presented a low cost and robust robot that is capable of navigating human-scale environments and interacting with people. This platform presents interesting opportunities to a wide range of interdisciplinary students, and we have started to investigate this through an 'Introduction to Social Robotics' course.

Our wiki (http://robotics.ils.albany.edu) contains videos of Nelson in action, as well as software, build instructions, and part layouts.

8. ACKNOWLEDGMENTS

This work is funded in part by the NSF CPATH program (award number IIS 0722277). We would like to thank our students, who patiently dealt with debugging issues in early versions of Nelson and who implemented the example projects discussed in this paper.

9. REFERENCES

[1] L. Aryananda and J. Weber. Mertz: A quest for a robust and scalable active vision humanoid head robot. In *IEEE-RAS/RSJ International Conference on Humanoid Robots*, November 2004.

[2] T. Balch, D. Blank, M. Guzdial, and D. Kumar. IPRE: Institute for personal robotics in education, 2007. (www.roboteducation.org).

[3] R. D. Beer, H. J. Chiel, and R. F. Drushel. Using autonomous robotics to teach science and engineering. *Communications of the ACM*, 42(6), 1999.

[4] C. F. DiSalvo, F. Gemperle, J. Forlizzi, and S. Kiesler. All robots are not created equal: the design and perception of humanoid robot heads. In *DIS '02: Proceedings of the 4th conference on Designing interactive systems*, pages 321–326, New York, NY, USA, 2002. ACM.

[5] Z. Dodds, L. Greenwald, A. Howard, S. Tejada, and J. Weinberg. Components, Curriculum, and Community: Robots and Robotics in Undergraduate AI Education. *AI Magazine*, 27(1):11–22, 2006.

[6] M. Kleinehagenbrock, S. Lang, J. Fritsch, F. Lomker, G. A. Fink, and G. Sagerer. Person tracking with a mobile robot based on multi-modal anchoring, 2002.

[7] A. N. Kumar. Three years of using robots in an artificial intelligence course: Lessons learned. *Journal on Educational Resources in Computing: Special Issue on Robotics in Undergraduate Education*, 4(3), 2004.

[8] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige. The office marathon: Robust navigation in an indoor office environment. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 300–307, 2010.

[9] J. Posner, J. A. Russell, and B. S. Peterson. The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology. *Development & Psychopathology*, 17:715–734, 2005.

[10] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng. ROS: an open-source robot operating system. In *Open-Source Software workshop of the International Conference on Robotics and Automation (ICRA)*, 2009.

[11] E. Sklar, S. Parsons, and M. Q. Azhar. Robotics across the curriculum. In *Proceedings of the 2007 AAAI Spring Symposium on Robots and Robot Venues: Resources for AI Education*, 2007.

[12] E. J. Tira-Thompson. Tekkotsu: A rapid development framework for robotics. Master's thesis, Carnegie Mellon University Robotics Institute, 2004. Available online at http://www.tekkotsu.org.