Maxwell: Developing a Low-Cost Mobile Manipulator

Michael Ferguson
Vanadium Labs LLC
Averill Park NY, 12018 USA
ferguson@vanadiumlabs.com

Abstract

In this paper we describe our recent work developing an inexpensive mobile manipulator, capable of operating in human-scale environments. Our robot (Maxwell) has a 5-degree of freedom arm, pan and tilt head with depth sensor, and differential drive mobile base. The robot is fully integrated with the open-source ROS framework, easing software development and allowing interoperability with a large amount of existing software.

Keywords

Keywords go here.

ACM Classification Keywords

I.2.9 Robotics – Commercial robots and applications.

Introduction

Recently there has been great interest in the development of mobile manipulators, robots which are capable of both navigating dynamic environments and manipulating objects. A number of advanced mobile manipulators have recently been introduced to the research community, such as the Willow Garage PR2 [1] and Meka Robotics M1 [2], however these platforms cost hundreds of thousands of dollars making them out of reach for all but the best funded labratories. In this paper we present recent work developing a low-cost mobile manipulator which is capable of operating in human-scale environments, and leverages off-the-shelf components and the open source Robot Operating System (ROS) [13]. The current iteration of robot, called 'Maxwell', was built for under $4,000, however we envision future iterations to cost between $2,000 and $2,500, a reasonable cost for a platform that offers many capabilities and opportunites in studying and researching the numerous aspects of navigation in human-scale and dynamic environments, object recognition and manipulation, and human-robot interaction.

In developing our robot we set forth a series of design requirements. First and foremost, the platform should be low cost, so that it is accessible to a wide audience. We used off-the-shelf components whenever possible. For those parts which could not be obtained off-the-shelf, we designed simple and cost effective parts that could be easily laser cut from plastic sheet stock.

Second, we wanted a platform that could operate in human-scale environments. We created a robot that was closer to human size in height, but not so large as to be intimidating. While this requirement would normally result in a robot which is large and bulky to move about, we designed Maxwell such that he could be quickly broken down into several sections for easy transport. This is especially important should the robot be entered in any of the many competitions and benchmarks which require travel, such as the ICRA or AAAI mobile manipulation challenges, or the international RoboCup@Home contest.

Figure 1: Maxwell, a human-scale mobile manipulator.

Finally, because we believe that 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 as the amount of data generated by these sensors is not easily transported over typical wireless networks.

To realize these design goals, we constructed Maxwell, a mobile manipulator consisting of a PC and co-processor for motion control and planning, a mobile base, head, and arm. We have chosen to implement the control structure of our robot using the Robot Operating System (ROS) [13]. The ROS framework functions as a set of processes, potentially on different hosts, which are connected at runtime in a peer-to-peer network. Since individual tasks can each be handled by a separate process, it greatly encourages and promotes code reuse. By using ROS, we gain a significant head start on many of the challenges of mobile manipulation, such as kinematics, navigation, and 3d perception. We describe the major elements of Maxwell in the following sections, while an overview of component costs is found in Table 1.

ArbotiX2 Co-Processor

Maxwell is designed to carry a typical full-size laptop (14.1" screen) on his rear deck. Since typical computers do not have direct access to external IO, our laptop connects to external robot hardware through a Vanadium Labs ArbotiX2 RoboController [3]. The ArbotiX2 is an open-source robot controller with a 16MHz AVR, digital and analog input and output channels, ability to control AX, RX, and EX series Dynamixel servos, and available dual-channel 30 amp motor drivers with encoder capture. In Maxwell, the ArbotiX2 is used as co-processor, connected over USB, which handles all real-time operations and passes commands between the PC and the servos, as well as implementing closed-loop PID control of the differential drive mobile base.

We have developed a set of ROS bindings for the ArbotiX2. These bindings allow access to all aspects of the hardware. We can query and set the position of individual or chains of joints. Each joint can also be queried for temperature and voltage, allowing system monitoring. The ROS bindings connect the mobile base control in a standard ROS way, allowing Maxwell to access many tele-operation and navigation programs which have been written for other ROS-enabled robots. Finally, the ArbotiX2 has sixteen input and output ports, including eight analog inputs, which allow the easy addition of new sensors or actuators as needed. The ROS bindings and firmware for the ArbotiX2 is released under a BSD License, and documentation can be found at http://www.ros.org/wiki/arbotix.

Table 1: Overview of Costs

Component	Cost
Mobile Base (Drive Motors, Wheels, etc)	$400.00
ArbotiX2, Motor Drivers, Wiring, Battery	$275.00
Upright Column and Hardware	$40.00
Pan and Tilt Neck, Kinect	$250.00
Arm (Servos, Brackets, Wiring)	$1700.00
Hokuyo URG-04LX-UG01 Laser Range Finder	$1175.00

Mobile Base

Mobility is provided by a differential drive base with two independently controlled wheels, capable of achieving approximately 0.6 m/sec with a 20 pound payload. While there are a number of mobile bases available, some already supported by the ROS software platform, a custom laser-cut design was implemented to offset Maxwell's considerable height and payload requirements, without exceeding our overall budget requirements. A closed loop PID speed control is implemented on the co-processor.

An upright column, made of 8020 aluminum framing, is mounted to the top of the mobile base. This column raises the head to human-height and brings the arm within reach of objects on table tops. The column is constructed of three individual pieces and is easily removed from the base, allowing Maxwell to be disassembled into pieces with no dimension larger than 20 inches, allowing for easy shipment or transport in a protective case as seen in Figure 2. Disassembly of the column requires removing only 10 screws, allowing Maxwell to be disassembled or reassembled in minutes.

Sensory

Primary sensory is from a Microsoft Kinect sensor [4] which is mounted on a pan and tilt neck, which provides a wider field of view. The Kinect generates both traditional RGB images and depth images which are used to construct a point cloud representation of the environment. Maxwell is currently fitted with a lower-cost Hokuyo laser scanner on his base, however, it is hoped that future advances in mapping and localization technologies for the Kinect will render this somewhat expensive device unneeded.

In addition to ranging sensors, Maxwell has also been fitted with a shotgun microphone for understanding natural language, although use of a bluetooth headset is also common.

Figure 2: Maxwell, disassembled for shipment and packed in a Pelican case.

A Low-Cost Overhead Arm

Maxwell's arm consists of a 5 degree of freedom (DOF) manipulator with a 2-servo gripper. The arm is constructed of various-sized Dynamixel servos, which have a robust and easy to use serial bus architecture. The arm has approximately a 50 ounce payload when fully extended, however the current gripper design currently limits overall payload ability to 20 ounces for most object shapes. An overhead grasp configuration affords a number of grasping possibilities, while minimizing the number of joints in the arm.

Perception

3-D perception is achieved through a Kinect, which provides both a traditional RGB image as well as a depth image. The depth image is then converted into a point cloud. ROS is fully integrated with OpenCV [9] and the Point Cloud Library (PCL) [5], allowing students to quickly dig into advanced 2-D and 3-D perception.

With 3-D perception, coupled with audio input, we have a number of options for Human-Robot Interaction. We have integrated the Pocketsphinx [6] speech recognizer with ROS. Pocketsphinx can easily be reconfigured for new language models for specific, student-led HRI projects, using a series of web based language modeling tools developed by the Sphinx team.

Manipulation

While we intend for Maxwell to use the very capable arm navigation capabilities found in ROS [7], we have for the time being implemented our own elementary manipulation pipeline. This pipeline offers capabilities for solving forward and inverse kinematics and moving the arm through a trajectory of poses, which is sufficient for our current generation of student projects. As the ROS arm navigation stack approaches a version 1.0 release, and our own configuration files for using it have matured, we will have collision-free arm motion capabilities.

We use the existing ROS arm_kinematics package to compute inverse kinematics (IK) solutions. However, as our manipulator has only 5 degrees of freedom, we cannot reach all 6-DOF poses. The X, Y, Z coordinates of the piece are therefore converted into a 6-DOF pose by several heuristics:

- The yaw angle of the grasper is determined uniquely from X and Y coordinates of the grasper.
- The roll angle of the grasper can be specified to maximize the grasp ability.
- The optimal pitch angle of the grasper is an overhead grasp, however we can search the IK configuration space within a small region around overhead grasping to find a suitable solution.

Navigation

We have integrated the ROS navigation stack [11], allowing Maxwell to build maps of his surroundings using gmapping [10] and navigate throughout the environment autonomously. An example map is shown in Figure 3.

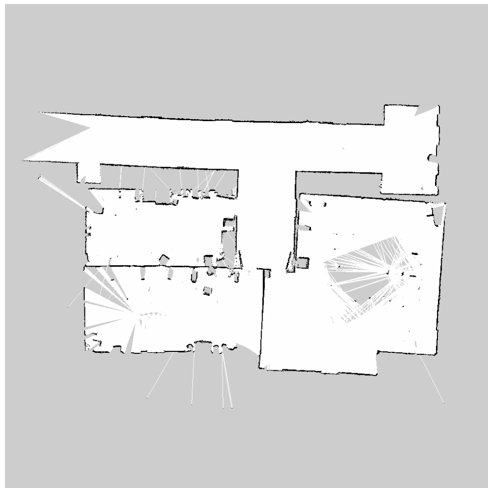

Figure 3: A map of our lab, generated by Maxwell using gmapping.

Having a working implementation of a complete navigation system is a huge head start for students. Additionally, the ROS navigation stack has been designed in a convenient way that allows individual components, such as the local trajectory planner, to be replaced with new implementations. This allows students to try out new algorithms quickly, for instance, they could implement a human-aware trajectory planner which follows social norms.

Diagnostics and Calibration

One of the major aspects that must be addressed in any robust platform is diagnostics and calibration. Maxwell employs the ROS diagnostics system to monitor and log all aspects of the robot, such as joint positions, temperatures and voltage, encoder values, and the status of each process in the runtime ROS network. Having logs available allows users to more easily debug issues that might arise in a complex manipulator such as Maxwell.

With any multi-joint, multi-sensor robot comes the issue of calibration of all joints and sensors such that hand-eye coordination can be accomplished. Maxwell is especially in need of automatic calibration due to the use of low-cost, and thus low-tolerance, parts used. To calibrate Maxwell we are working on an approach similar to [12], but adapted to lower-cost components and their inherit issues.

Example Applications

Maxwell has spent the majority of a semester deployed in the Social Robotics Lab at University of Albany. A group of undergraduate and first-year graduate students enrolled in our Robotics Seminar at the University at Albany, SUNY are currently preparing Maxwell for entry to the AAAI Small-Scale Manipulation Challenge, which focuses on robotic chess for 2011. In addition to assisting in implementing the arm navigation capabilities for Maxwell, this project team is constructing a perception pipeline for recognizing and understanding chess boards, pieces, and the game in play using them, relying on the Kinect sensor.

A second group of students are studying algorithms for the detection, tracking, and following people using Maxwell. Using the depth image from the Kinect they are developing a robust 'upper body detector', which will be combined with speech and face detection. The task is based on the 'Follow Me' challenge from RoboCup@Home [8], an international showcase of service and assistive robotics.

Conclusion and Future Works

We have presented here our work on Maxwell, a low-cost, human-scale mobile manipulator suitable for educational and research uses. Maxwell uses off-the-shelf components, the ROS framework, and is open-source wherever possible.

Future work with Maxwell will focus on improving automatic calibration of intrinsic parameters, adding a vertical trolley for the arm, and adding self-recharging capabilities to allow extended usage. The addition of a vertical lift for the arm will allow Maxwell to manipulate items on the floor in addition to those found on table tops, greatly improving the abilities of the platform.

References

[1] http://www.willowgarage.com/pages/pr2/overview.

[2] http://mekabot.com/.

[3] http://www.vanadiumlabs.com/arbotix.

[4] http://www.xbox.com/en-US/kinect.

[5] http://pointclouds.org.

[6] http://cmusphinx.sourceforge.net/wiki/.

[7] http://www.ros.org/wiki/move_arm.

[8] http://www.ai.rug.nl/robocupathome/.

[9] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[10] G. Grisetti, C. Stachniss, and W. Burgard. Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2005.

[11] 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.

[12] V. Pradeep, K. Konolige, and E. Berger. Calibrating a multi-arm multi-sensor robot: A bundle adjustment approach. In *International Symposium on Experimental Robotics (ISER)*, 2010.

[13] 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.