A Modular and Flexible Bimanipulation System for Space-Analogue Experiments

Alejandro Hernandez Herdocia, Azad Shademan, Martin Jägersand

Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada.
{alejandv, azad, jag} at cs.ualberta.ca

Abstract

This paper presents a system developed with off-the shelf components which will be used to study human-robot cooperative tasks from tele-manipulation to supervisory control. The approach enables the design to be modular and flexible, thus presenting an upgradable laboratory and test bed for prototyping autonomy-enhanced and supervised-control activities for Space-Analogue Mobile Manipulation.

1 Introduction

Mobile manipulators are relatively complex to design and build because of the big share of temporal, economic, technical and human resources employed. This paper presents a system developed with off-the shelf components. The goal of this project is provide a platform to study human-robot cooperative tasks from tele-manipulation to supervisory control for space applications. Usually the robot development is targeted towards the application and deployment of the robot itself while leaving the design and validation of tasks and activities to be done in simulations or in exact replicas of the robot.

While an accurate simulation is a good tool to verify some functionalities, in most cases the algorithms need to be also verified on the prototype. Working with exact replicas is resource intensive. One way to avoid these shortcomings is to use a general approach when designing the tasks and activities, such that they can be tested in a generic test-bed.

It is important to *design* and not merely put together available modules in the same way a plan is needed to tackle a problem or develop a project. By putting together hardware without having a design, the risk of stumbling into problems in the future increases. Most of the major problems can be avoided since the beginning by considering critical and desirable features and their trade-offs. This pays-off later in the course of the project when the goals become more complex and the design is robust and tolerant because there is a clear notion of the advantages and limitations of the developed system.

In order to build a platform that is accessible to many research institutes, we propose using off-the-shelf components. Building an off-the-shelf component-based system enables the design to be modular, flexible and upgradable while not requiring particular technical skills in specific domains. The resulting test-bed can be used to prototype autonomy-enhanced and supervised-control activities for Space-Analogue Mobile Manipulation (*e.g.,*planetary exploration and on-orbit servicing). Moreover, the approach enables universities to work on the challenging problems and develop algorithms which can later be transferred to space applications. The developed solutions are generic enough to be used then on any platform with minor transferring and translation work.

The system was built without the use of a specialized engineering facility; it is integrated by off-the-shelf components for the mobile base, arms, computation, sensing and power systems. A distributed software system was designed and implemented. Most of the design freedom is among the multitude of choices for computer components and various battery types and chemistries. There are some choices for mobile bases and a few different commercial robot arms that would fit. With a proper design for power and hardware components, many issues that cannot be addressed in simulations, are addressed.

Because of the complex nature of some tasks, full autonomy is work in progress. Some of the tasks are difficult enough that require supervised control from a user despite of its potential automation. This can be achieved by working with an accessible platform instead of deploying a solution directly in the field. The gulf between autonomous robot control and master-slave tele-operation is yet to be overcome by easing the task design and simulation of the autonomous control and by reducing the costs of communication delays and interaction [26] over the operators [28].

Following this introduction, Section 2 presents a short survey of systems similar or related to the one presented. Section 3 outlines the design. The system integration and preliminary tests are presented in Section 4 to finally conclude in Section 5.

2 Related Work

Mobile manipulators appear in several research areas including, but not limited to, humanoid robots, human-robot interaction, rehabilitation, assistance and space exploration. Although these areas are not directly related to space robotics, they are relevant to our paper for their mechanical design insights.

Robonaut [3][6][14], Rollin' Justin [7][8][25] and EUROBOT [13] are three dual arm manipulators targeted for space applications. All are custom engineered, and even EUROBOT has both of its versions (wet and space) customized. While Robonaut and Rollin' Justin have mobile rolling bases, EUROBOT features a leg as a means of locomotion. All of them have been engineered through a long lapse of time. All of them are designed in general for space applications. Justin and Robonaut are more focused towards planetary exploration while EUROBOT is towards on orbit servicing.

Another group of robots that are relevant to this work comprise Golem Krang [31], uBot [11][12][32], PR2 [33], ARMAR [2][4], WENDY [23], and ISAC [18]. They are all dual arm mobile manipulators like Robonaut, Rollin' Justin, and EUROBOT, but they have different applications such as caretaking, home assistance or general dual arm manipulation research. These projects provide some substantial research on dual arm mobile manipulation that can be taken into consideration when generating a general framework for the system.

Finally, a third group encloses DEXTER [24], DOMO [15], Cardea [9], HERB [30], EL-E [19], UMAN [20] and PowerBot [1]. All these robots are relevant because of either their contribution or their characteristics and main research focus. Some of these projects center on bimanual manipulation such as DEXTER and DOMO, while others center in mobile manipulation (CARDEA, HERB, EL-E, UMAN and PowerBot). In particular, HERB provides insight on how to integrate off-the-shelf components to achieve a working system.

3 Design

Nimble human-like dimensions and capabilities are desirable for interacting with humans. It is easier to design a task considering how a person would do it. Moreover, it is desirable to have an agile mobile dual whole-arm manipulator capable of contact manipulation, in order to design tasks taking advantage of its dynamic capabilities.

There are several arms and mobile platforms commercially available, but few are compatible and suitable to combine. Usually, arms have their power electronics detached from the main body and in general for robots, the control software is sometimes proprietary and closed. It is desirable then to consider this aspects when selecting the modules. It is even possible to take the off-the-shelf component integration to a lower level and build the mobile base or the manipulator with commercial modules [29]. As in every design task, there are several trade-offs and decisions to be made regarding the capabilities and constraints when selecting them. It is important to differentiate which constraints are critical and which are desirable then find hardware that can meet these constraints.

Off-the-shelf component integration has both advantages and disadvantages compared to custom design and out-of-the-box complete platforms. Some advantages are as follows:

- It saves time.
- It has state-of-the-art components without possessing the engineering expertise to create every detail.
- The performance of each purchased subsystem is well defined at the outset of integration.
- The capabilities of the system can be personalized, *w.r.t.*out-of-the-box systems.

Some drawbacks are as follows:

- Subsystems may not be combined in an optimal way.
- Detailed information and source code may be proprietary and not available.
- Opportunities for developing new state-of the art parts are missed.
- Potential difficulty exists in maintaining complex modules and parts.

Because off-the-shelf design considers hardware that might not fulfill the exact need, there are also several variables to take into consideration when making a choice about which option to use (some of the features may even conflict with the objective). In case of the arms, the characteristics that vary include payload, sensing, geometry and workspace; in case of the mobile base, the payload, autonomy and control features; for on-board computers, there are even more alternatives.

The following subsections outline the chosen hardware and some of the modules that will be included, power requirements for the systems on-board the robot, computational hardware, and the software approaches and framework.

3.1 Hardware and Spatial Configuration

In this case, the desired outcome is a platform for evaluating and prototyping semi-autonomous tele-operated manipulation tasks. A two arm system is considered because some of the applications of the robot require to explore bimanual and whole-arm/body manipulation. Another important characteristic is that the arm should have no or minimal external components (*e.g.,* amplifiers), *i.e.,* it should be self contained. It is also important that the arms are backdrivable as some applications involve human interaction. Regarding the mobile base, it should be sturdy and have a high payload capacity, and we prefer a compact size for increased maneuverability. Moreover, the arms should be preferably mounted at a height making the workspace of the arms reside at the level of a common office desk (approximately 75 cm from ground level).

Currently, the design integrates two Barrett WAM (Whole Arm Manipulator) arms atop a Segway RMP mobile platform; see Fig. 1. The WAM is appealing due to its low total and moving mass, backdrivable joints, high-efficiency cable transmission (in turn, it may raise the need to exchange worn cables periodically), integrated motor controllers and amplifiers into the joint motors, and open interface and source code for control. The Segway RMP base is appealing for its significant payload-to-weight ratio, moderate dimensions, and market-proven design.

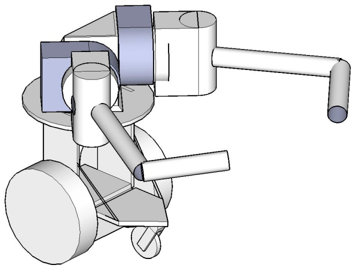

Figure 1. CAD model of assembly

The WAM, unlike conventional arms, is designed to exert contact with the end effector and also with any surface of the links to grasp objects, for instance, for greater force and contact area. Example of this is between the forearms (Fig. 2). By mounting the arms vertically at a 90°offset, the workspace resembles that of a human [21], *i.e.,* there is no overlapping space on the back. There is a substantial overlapping workspace for two-arm interaction in the front (shown in Fig. 1). A lightweight mount was built from plywood for this purpose.

The arm assembly can be set either at the height of the standard Segway platform to optimize reach at a height of 70 cm above the ground level. This configuration allows the robot to lift some large objects from the floor, as shown in Fig. 6. Alternatively the arm assembly can be set just above the Segway fenders on a shortened mount to improve the reach for manipulation of objects on floor level. The overall dimensions of the robot are comparably compact. However, the arm configuration makes the shoulders slightly wider than a standard single door frame.

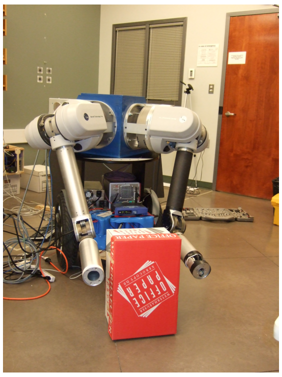

Figure 2. WAM whole arm contact manipulation

The computers and batteries are placed in the space between the wheels on the Segway. The current configuration consists of two PC computers for vision and sensing and one pc104 computer integrated into the arm for motion control. A 4-port wireless router connects the on-board computing with a local ethernet and allows user-control via wireless ethernet. Fig. 4 illustrates a hardware diagram for all the subsystems of the robot (On-board on the right, and User/Remote on the left). Fig. 3 shows the spatial arrangement of the modules within the robot.

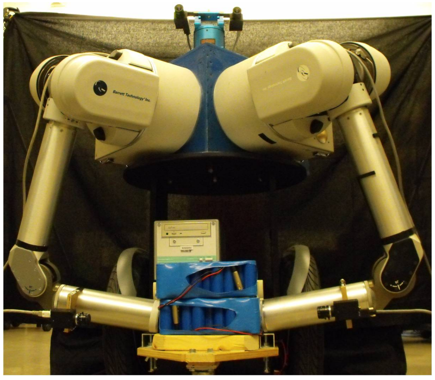

Figure 3. 2WAM arms, forearm carmeras, pan-tilt unit with stereovision, computers and power integrated on Segway RMP

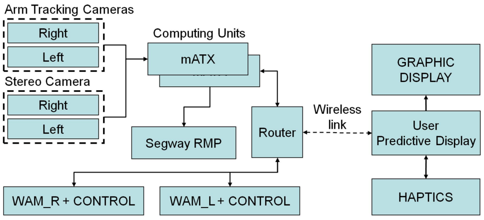

Figure 4. Hardware setup.

Table 1. Power Specification for the system

Subsystem	Idle	Typical	Peak
WAM 4 DOF	18W	28W	600W
2 Computers	70W	150W	190W
Cameras etc	$\leq$10W	$<$10W	$\leq$ 10W
Segway RMP	Integrated battery		
TOTAL	$\leq$ 125W	$<$ 250W	$\approx$ 1400W

Table 2. Specification of the power source for the robot.

Characteristic	Needed	Achieved
Running time idle	5 hours	9 hours
Running time typ load	3 hours	$\approx$ 4.5 hours
Nominal Voltage	$\geq$ 100V	144V
Peak Discharge Current	11A	25A
Peak Load	1.5KW	3.6KW
Total Power (batteries)	100V × 6Ah= 600 Wh	144V × 8Ah= 1152 Wh

The robot will support research in visual and force-based tele-operation. The system considers two sets of cameras, namely Arm Tracking and Stereo Camera. The former group are those cameras that will be mounted on the fore-arms while the latter are the ones are mounted on the pan-tilt turret atop of the two arm configuration. Both sets of cameras can be used for visual servoing (Eye-in-Hand configuration with the Arm Tracking set and Eye-to-Hand with the Stereo Camera set). Additionally the cameras will also be used to feed the visual information to the 3D modeling of the environment for navigation and manipulation purposes.

3.2 Power

A mobile manipulator contains numerous subsystems that need to be powered. For a less complex system, the most convenient solution is often to buy independently battery-powered subsystems, *e.g.,* laptops, powered sensors, etc. However, for more complex systems, it is better to minimize the number of systems that need to be separately charged and to equalize run-time of the subsystems. This favors a single-power-source system and power bus configuration. The modules in this design use universal AC/DC line voltage power supply units. This adds the convenience of direct switching between AC line voltage, and battery power. The purpose of having batteries is to make the robot self-contained and investigate prioritization of power use, *i.e.,* evaluate the cases when the power on-board the robot is limited. This puts the system's conditions closer to an actual deployed robot.

The main power needs are listed in Table 2 and the power consumption for idle, engaged and peak stages is shown in Table 1.

Table 3. Power consumption of different computer solutions evaluated. The two frequencies indicate the min (idle) and max frequencies (engaged) in Intel "Speedstep" automatic CPU frequency adjustment. The Shuttle Zen lacks Speedstep.

Machine	Shuttle Zen	Thinkpad X61	mATX G33	ATX I975
CPU freq GHz	NA/3.2	0.8/2.4	2/2.67	1.6/2.4
Etching	90nm	45nm	45nm	65nm
Cores	1	2	4	4
Peak Std Volt	110W	53W	115W	120W
Peak UnderVolt	90W	48W	96W	NA
Idle	57W	16W	35W	68W
Efficiency MOPS/W	127	346	346	314

Lead Acid, NiCd, NiMH, and LiION batteries were evaluated considering capacity, energy density, depth of discharge, inner resistance, maximum charging cycles, price and robustness (tolerance to overcharge, deep discharge, mechanical shock and impacts). NiCd were selected because they combine robustness and low price. The power source uses four 36 V 8 Ah High Capacity NiCd packs in series. This powers the entire system excluding the Segway RMP. The system has been tested to run until deep discharge of the batteries.

3.3 Computing Hardware

The computing power on-board of lab robots ranges from specialized embedded computers to standard laptops. One popular choice of the former are the PC104 type because of its expandability options and generic x86 architecture. Although appealing, the tradeoff is that these boards are not designed for high-end computing and do not feature the most powerful recent processors. This is inconvenient when processing high bandwidth data (such as vision or multiple sensor data). Laptops on the other hand, have processors almost as powerful as workstations and are already integrated into small packages. The tradeoff is expandability because of the high level of integration.

The middle ground is selecting a suitable desktop Mother Board and configuring a computer with a high-end processor. The main advantages of selecting such solution are expandability and the option of tuning the computing system for optimizing power consumption. Moreover, desktop processors usually feature more cores (most high-end laptop processors feature 4 cores; desktop computers feature 4, 6, and even 8 cores). Table 3 shows a comparison of the tested systems.

The on-board computing system will consist of two (currently only one) mATX PC based Foxconn G33M-S Mother Boards featuring Intel XEON processors and low-power integrated graphics. These are fit into standard 13” cases. If higher performance is needed, an Intel Core i7 solution could be used. These high-performance PCs are dedicated to information processing and running the high-level control of the mobile base. In addition to the two high-performance PCs, there are two (desirably just one) computers (a Shuttle ZEN and a PC104) running a real-time operative system to control the arms.

Two IEEE1394b boards were incorporated for image acquisition. The images are received from four high-frame-rate PtGrey Grasshopper tracking cameras (MONO8, image size of 640×480 pixels captured at 60 Hz) mounted on the pan-tilt unit and the forearms.

3.4 System Software

It is almost inevitable that different system components use different computational architectures when using the off-the-shelf approach. One way to cope with these differences is to employ a solution that takes care of unifying the computational power. One such solution is the Parallel, Virtual Machine (PVM) [16]. PVM unifies different computers on a network into one logic processing entity. This both solves the problem of differences in architecture and enables to distribute the video processing task over the available computing resources. This is important in particular for vision, as processing video on a single computer would be too taxing.

Software should be developed within a framework to guarantee robustness, flexibility, maintainability and platform portability. The software for this system is built upon a component-based approach [10] not only in hardware but in software as well. Using this framework ensures tolerance to hardware (configuration changes, new sensors, . . .) and software (new drivers, applications, . . .) updates.

Fig. 5 shows the software architecture. The principal building blocks are shown as solid boxes, and they are grouped into semantic groups. These groups are defined in terms of the functions they cover. This grouping gives a better understanding of the information flow in the system (a more complex diagram organized as in [10] is possible but it will be more complex as each module includes from the driver to the application). In this diagram the safety daemons are omitted for better illustration purposes. By including the approach in [5], robustness and safety of the system is ensured. This is an extensible framework (using reuse and modularity) to systematically test software.

The overall architecture of the software follows a master-worker architecture. One master process controls several worker processes, each of them operating a different component of the robot. In general, we use libraries that are written in C/C++ (such as XVision [17] and ViSP [27] and accessed with a scripting interface. At the moment, MATLAB and its mex scripting are used for convenient prototyping. Other scripting languages such as Python will be used in the future as an interface to the underlying C/C++ libraries. Using a scripting language, ensures that the functional modules run efficiently in C/C++, while scripting enables fast and high-level prototype development.

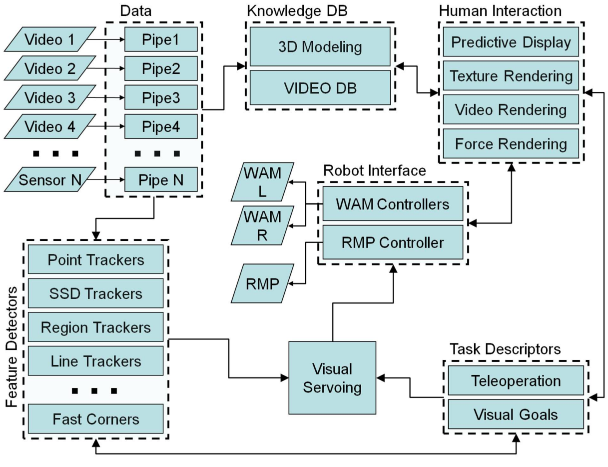

Figure 5. Software setup.

Processes are usually run in different places depending on the nature of the process itself. Vision-related and high-demanding processes are run on the PCs while time-critical control of the arms runs on the PC104 or the Shuttle computers. Safety daemons, will be run in each subsystem to monitor the conditions of operation and prevent undesired behaviors.

4 System Integration and Test

In this system, all the major computation is done onboard the robot, whereas the visualization and task specification is processed on the user side. The commands are then sent over to the robot through the wireless link. All the computations that are critical to ensure a transition to a stable state are run on-board. Visualization is done on the user side as it makes more sense to run the process as close as possible to the output screen to avoid lag in the visualization. In fact, this platform will be used for predictive display research.

The robot was programmed to pick up a box off the floor as shown in Fig. 2. First, taught trajectories were recorded. These trajectories were later replicated to lift the box. The WAM arms are *taught* through direct human contact interaction with the arms. To the moment, this test was performed in open loop. In future work the task will be performed using visual servoing to close the loop.

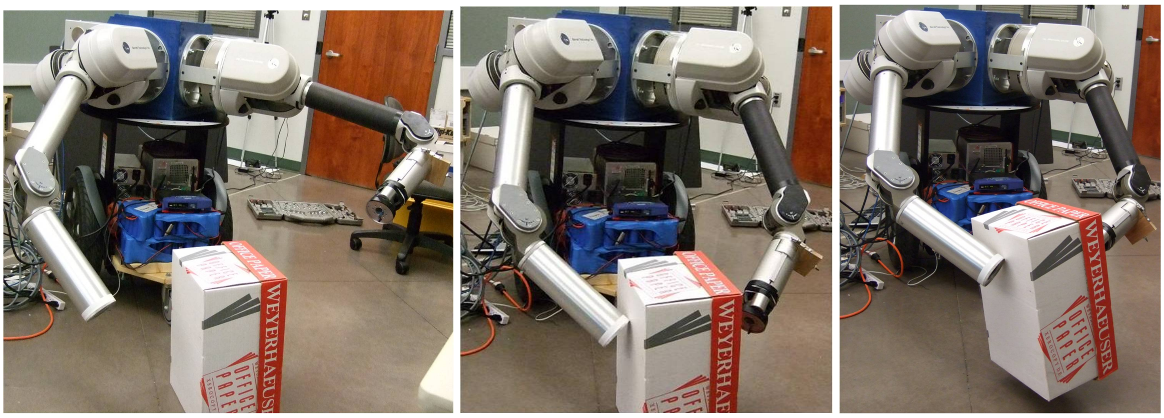

Figure 6. Two arm manipulation: Lifting a box

In this test, the computation was distributed across the on-board computing systems. One of the PCs ran a worker process to control the mobile base and a vision worker. The PC104 and the Shuttle were each running a control worker process for each WAM arm. On the user side, the control of the system is generated on a laptop connecting to the robot's wireless network.

5 Conclusions

Previously, only those with access to engineering facilities and expertise were the only ones able to build a mobile manipulator. The availability of lightweight self-contained robot arms, such as the WAM, makes it possible to integrate such a system using commercially available components (computing systems, batteries, distributed computing software, . . .).

Only a small number of dual-arm mobile manipulators have been built. However they could reach a much wider audience if built from off-the-shelf components. This enables a broader audience to do research, perform tests, and validate task designs. The objective of this system is to provide a means to design autonomous tasks for robots working in remote environments such as in planetary exploration or in on-orbit-servicing.

In future work the mobile manipulator will be used to research semi-autonomous vision and contact-sensing behaviors in tele-robotics. The main idea is to use higher level routines to relieve the tele-operator from tedious detailed motion control(moving from motion level to task level). Furthermore this approach is likely to be less sensitive to the inevitable time delays in tele-manipulation and at the same time it becomes more robust to variations in the task. This latter poses new challenges which need to be addressed to ensure tolerance and robustness of the overall system.

A Phantom Omni haptic device will be used to facilitate tele-robotic control of the WAMs. This has been already tested on one arm but is yet to be extended to both arms in order to provide straight-forward tele-operation. Using the Phantom, we can directly tele-manipulate the joint angles of the WAM or use it as a teaching input to create primitive motions. Our last objective is to issue deictic commands, using a visual interface in a hand-held device or a workstation. This is the same as instructing a peer to do certain task only by a minimal, high level description.

Including new items such as sensors, processing units, actuators or behaviors (in software) is greatly eased by the modular approach in software and hardware. Developing software or modifying the configuration of the robot becomes a fast and simple task.

Acknowledgments

We would like to thank a laboratory colleague Simon Leonard for the use of OpenMAN [22] software for trajectory planning, and the Segway group at the University of Alberta for the use of their Segway code. This project is partially funded by iCORE through ICT Student Scholarships.

References

[1] ActivMediaRobotics. Manipulators, September 2008. Brochure from ActiveMedia robotics.

[2] A. Albers, S. Brudniok, J. Ottnad, C. Sauter, and K. Sedchaicharn. Armar iii design of the upper body. Technical report, Institute of Product Development University of Karlsruhe (TH), 2006.

[3] R. Ambrose, R. T. Savely, S. M. Goza, P. Strawser, M. A. Difiler, I. Spain, and N. Radford. Mobile manipulation using nasas robonaut. *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, 2:2104– 2109, 2004.

[4] T. Asfour, K. Berns, and R. Dillmann. The humanoid robot armar: Design and control. In *Humanoids*, page 6, 2000.

[5] S. Bensalem, M. Gallien, F. Ingrand, I. Kahloul, and N. Thanh-Hung. Designing autonomous robots. *IEEE Robotics Automation Magazine*, 16:67–77, 2009.

[6] W. Bluethmann, R. . Ambrose, M. A. Difiler, S. Askew, E. Huber, S. M. Goza, F. Rehnmark, C. Lovchik, and D. Magruder. Robonaut: A robot designed to work with humans in space. *Autonomous Robots*, 14:179197, 2003.

[7] C. Borst, C. Ott, T. Wimbock, B. Brunner, F. Zacharias, B. Bauml, U. Hillenbrand, S. Haddadin, A. Albu-Schaffer, and G. Hirzinger. A humanoid upper body system for two-handed manipulation. In *ICRA*, 2007.

[8] C. Borst, T. Wimbock, F. Schmidt, M. Fuchs, B. Brunner, F. Zacharias, P. R. Giordano, R. Konietschke, W. Sepp, S. Fuchs, C. Rink, A. Albu-Schaffer, and G. Hirzinger. Rollin justin - mobile platform with variable base. In *ICRA*, 2009.

[9] R. Brooks, L. Aryananda, A. Edsinger, P. Fitzpatrick, C. Kemp, U.-M. OReilly, E. Torres-Jara, P. Varshavskaya, and J. Weber. Sensing and manipulating built-for-humanenvironments. *International Journal of Humanoid Robotics*, 1(1):1–28, 2004.

[10] D. Brugali and P. Scandurra. Component-based robotic engineering (part i) [tutorial]. *IEEE Robotics Automation Magazine*, 16(4):84–96, 2009.

[11] P. Deegan, R. Grupen, A. Hanson, E. Horrell, S. Ou, E. Riseman, S. Sen, B. Thibodeau, A. Williams, and D. Xie. Mobile manipulators for assisted living in residential settings. *Autonomous Robots, Special Issue on Socially Assistive Robotics*, 24(2):14, 2008.

[12] P. Deegan, B. J. Thibodeau, and R. Grupen. Designing a self-stabilizing robot for dynamic mobile manipulation. In *RSS*, 2006.

[13] F. Didot, P. Schoonejans, E. Pensavalle, G. Battistoni, S. Ferraris, S. Estable, T. Huesling, and EADS-Astrium. Eurobot underwater model: System overview, test results & outlook. In *i-SAIRAS*, Hollywood, USA, February 2008.

[14] M. A. Diftler, J. R. Platt, C. J. Culbert, R. Ambrose, and W. J. Bluethmann. Evolution of the nasa/darpa robonaut control system. In *ICRA*, volume 2, 2003.

[15] A. Edsinger-Gonzales and J. Weber. Domo: A force sensing humanoid robot for manipulation research. In *Humanoids*, volume 1, 2004.

[16] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam. Parallel virtual machine.

[17] G. Hager, S. Puri, K. Toyama, and D. Burschka. A brief tour of xvision. —http://www.cs.jhu.edu/CIPS/xvision/index.html, 2010.

[18] R. A. P. II, K. Kawamura, D. M. Wilkes, K. A. Hambuchen, T. E. Rogers, and A. Alford. Isac humanoid: An architecture for learning and emotion. In *Humanoids*, 2001.

[19] A. Jain and C. C. Kemp. El-e: an assistive mobile manipulator that autonomously fetches objects from flat surfaces. *Autonomous Robots*, 28(1):45–64, 2009.

[20] D. Katz, E. Horrell, Y. Yang, B. Burns, T. Buckley, A. Grishkan, V. Zhylkovskyy, O. Brock, and E. Learned-Miller. The umass mobile manipulator uman: An experimental platform for autonomous mobile manipulation. In *In Workshop on Manipulation in Human Environments, at Robotics: Science and Systems*, 2006.

[21] N. Klopcar and J. Lenarcic. Kinematic model for determination of human arm reachable workspace. *Meccanica*, 40(2):203–219, 2005.

[22] S. Leonard. openman: An open source C++ toolbox for control and simulations of manipulators.

[23] T. Morita, H. Iwata, and S. Sugano. Human symbiotic robot design based on division and unification of functional requirements. In *ICRA*, volume 3, 2000.

[24] U. of Massachussetts at Amherst. Robotics at umass amherst - robots - dexter. http://www-robotics.cs.umass.edu/Robots/Dexter, September 2008. The information was taken from the UMASS at Amherst Lab of Perceptual Robotics Webpage.

[25] C. Ott, O. Eiberger, W. Friedl, B. Bauml, U. Hillenbrand, C. Borst, A. Albu-Schaffer, B. Brunner, H. Hirschmuller, S. Kielhofer, R. Konietschke, M. Suppa, T. Wimbock, F. Zacharias, and G. Hirzinger. A humanoid two-arm system for dexterous manipulation. In *Humanoids*, 2006.

[26] P. K. Pook and D. H. Ballard. Deictic teleassistance. In *Proc. IEEE/RSJ/GI Int. Conf. Intelligent Robots and Systems '94. 'Advanced Robotic Systems and the Real World' IROS '94*, volume 1, pages 245–252, 1994.

[27] ric Marchand, F. Spindler, and F. Chaumette. Visp for visual servoing: a generic software platform with a wide class of robot control skills. *Robotics & Automation Magazine, IEEE*, 12(4):40–52, 2006.

[28] T. B. Sheridan. Space teleoperation through time delay: review and prognosis. *IEEE Transactions on Robotics and Automation*, 9(5):592–606, 1993.

[29] C. Smith and H. Christensen. Robot manipulators. *IEEE Robotics Automation Magazine*, 16:75 – 83, 2009.

[30] S. Srinivasa, D. Ferguson, C. Helfrich, D. Berenson, A. Collet, R. Diankov, G. Gallagher, G. Hollinger, J. Kuffner, and M. VandeWeghe. Herb: A home exploring robotic butler. *Autonomous Robots*, 28:5 – 20, 2009.

[31] M. Stilman, J. Olson, and W. Gloss. Golem krang: Dynamically stable humanoid robot for mobile manipulation. In *ICRA*, 2010.

[32] B. J. Thibodeau, P. Deegan, and R. Grupen. Static analysis of contact forces with a mobile manipulator. In *ICRA*, 2006.

[33] Willow-Garage. Technical specifications. http://www.willowgarage.com/pages/robots/technical-specs, January 2010.