

Building a Mobile Manipulator from Off-the-Shelf Components

Alejandro Hernandez-Herdocia, Azad Shademan, Martin Jägersand

Abstract—A number of mobile manipulators have been custom engineered and built in labs worldwide. Mobile manipulators could become much more prevalent if they were assembled from commercially available components instead of being locally engineered and built. In this paper, we describe some tests and the integration steps in putting together power source, computers, and distributed software systems for integrating two Barrett WAM arms on top of a Segway RMP mobile base. The description is meant to be accessible so that researchers without engineering lab access or experience can make use of them. We conclude by showing initial experiments with two arm manipulation.

I. INTRODUCTION

Building a mobile manipulator, *i.e.*, a mobile robot with one, two or more arms, has traditionally been a major engineering project due to the complexity of designing the robot and its capabilities. In the case of mobile robotics it is common to buy a commercial mobile base and just put a laptop for control and sensing, and one will have a functional system. This paper covers design choices in building a two-arm mobile manipulator. Although the target applications of our design are in visual servoing and tele-robotics research, it would be suitable for various human-scale robotics projects.

Our system is built without the use of any specialized shop or engineering facilities; we integrate off-the-shelf components for the mobile base, arms, computation, sensing and power systems. We also design and implement a distributed software system. While there are several choices for mobile bases, and a few different commercial robot arms that would fit, most of the design freedom is among the multitude of choices for computer components and various battery types and chemistries.

Following this introduction, Section II presents a short survey of systems that are similar to ours. Section III outlines the subsystem design and the basic hardware building blocks for our system. In Section IV, we outline the power requirements for the systems on-board the robot. The computational hardware, and the software approaches and framework are detailed in Sections V and VI respectively, which are the main focus of the paper. The system integration and preliminary tests are presented in Section VII to finally conclude in Section VIII.

II. RELATED WORK

In the literature mobile manipulators are described in works about humanoid robots, human-robot interaction, rehabilitation, assistance and space exploration. We divide related works into two main groups. The first group contains those

which have a two arm manipulator on a dynamically balanced mobile base, the second one includes all those projects that have different hardware configuration.

Robots having two arms and a mobile base using dynamic balancing are NASA's Robonaut, Georgia Tech's Golem Krang and UMASS' uBot. Although the platforms are comprised of two arms on top of a dynamic balancing base, they have different purposes. Robonaut was mainly designed for Extravehicular Activity in space robotics. In this mode it is not using a mobile platform for displacing the robot's torso but a long positioning system [3]; the use of an RMP mobility platform is more for demo purposes. Georgia Tech's Golem Krang is mainly designed to study potential and kinematic energy use for handling human tasks [22]. University of Massachusetts' uBot is a smaller mobile manipulator designed for human interaction [8]. While Robonaut and Golem Krang feature degrees of freedom in the torso, uBot and our robot do not. It is worth noting that our system uses off-the-shelf components, while the others are custom built.

In a second group we include robots that are related to our research but differ on their features, either by the number of arms or by they type of mobile base. In this group we include several robots, which we will classify in three subgroups.

The first subgroup includes robots featuring two-arm manipulators atop of a statically balanced mobile base. DLR's Rollin' Justin [5] is being developed for research on dexterous manipulation using two coordinated manipulators atop of an adjustable moving base. ARMAR [2] is aimed towards dexterous manipulation as well. ISAC [11] and WENDY [17] were designed to assist individuals and they both emphasize the need of security mechanisms in human-robot interaction. All these systems were specifically engineered and are not available commercially. Recently, Willow Garage has developed the PR2 [23]. PR2 is an integrated mobile two-arm manipulator offering an open software architecture. What makes PR2 different is that it is a commercially available system.

We can observe that all of these systems are custom engineered and focus mainly in the problem of dexterous manipulation considering also a mobile base, which will be our first research approach but with different application.

The second subgroup includes fixed two-arm manipulators. UMASS' Dexter [18] uses Barrett WAM manipulators in the same way we are using them and MIT's DOMO [9] is all custom made. These two robots are platforms used specifically for dual-arm manipulation which in turn is relevant due to the applications.

The third subgroup contains basically single-arm mobile manipulators. MIT's Cardea, currently with only one

arm, was designed for navigation of unstructured interior spaces [6]. HERB [21] features the same configuration as our robot, yet with only one arm, and its application is more for day to day living assistance. HERB and Cardea are both built using a Segway RMP as a mobile base, like our system. ELE [12] is also for assistance, and features a statically balanced mobile base and a gripper for picking up objects from flat surfaces using a very user friendly interface. SAMM [14], UMAN [13] and a few commercial mobile manipulators, like the ones offered by ActivMedia Robotics [1] (Pioneer 3 with Pioneer Arm, the PowerBot with the Power/PTRV Arm), are statically balanced. Most of the research done on mobile manipulation is done with this setting, thus a milestone to our development.

These various related projects differ in the design of the parts, their development, the system integration and the architecture. They range from complete custom designs, such as Golem Krang to mostly using off-the-shelf components such as HERB.

III. DESIGN

For interacting with humans and solving human-scale tasks, nimble human-scale arms which are suitable for contact manipulation are desirable. As mentioned, a few such systems have been custom designed, engineered and built in various labs. However, mobile manipulators integrated from off-the-shelf components could reach a much wider audience. Of course off-the-shelf component integration has both advantages and disadvantages compared to custom design and out-of-the-box complete platforms. Using off-the-shelf components saves time, allows the use of state-of-the-art components without possessing the engineering expertise to create every detail, and (usually) means that the performance of each purchased subsystem is well defined at the outset of integration. Moreover it allows personalization of the capabilities of the system, *w.r.t.* out-of-the-box systems. Drawbacks are that subsystems may not combine in an optimal way, detailed subsystems' information and source code may be proprietary and not available. Opportunities for developing new state-of-the-art parts are missed and potential difficulty exists in maintaining the complex parts while lacking detailed knowledge of their functioning.

While there are both numerous arms and mobile platforms available, few seem to be suitable to combine. A main problem with robot arms is the commonly separate amplifier cabinet. Another is that manufacturer control software is proprietary and not modifiable. A possibility is to build the arm itself from available components [20]. On the mobile base side, few have the carrying capacity for even one, let alone two arms.

Then what components are suitable to consider and which can be integrated? As in every design task, there are several questions one has to answer regarding the capabilities and constraints when selecting them. One has to take into account which of these constraints are critical and which are desirable then find hardware that can meet these constraints or change the design. Because off-the-shelf design considers

hardware that might not fit exactly to our need, there are also several variables to take into consideration when making a choice about which option to use. 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.

In our case, we want to obtain a platform for evaluating and prototyping semi-autonomous teleoperated manipulation tasks. We choose a two arm system because we want to explore applications of 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.*, self contained, as they will consume space on-board and will reduce the potential payload the robot can carry. We also want the arms to be back driveable as we would like to conduct tests with human interaction. Regarding the mobile base, we want it to be sturdy with a high payload capacity, and we prefer a compact size. Moreover, we want to place the arms at a height such that the workspace of the arms resides at the level of a common office desk (approximately 75 cm).

We choose to integrate two Barrett WAM arms on a Segway RMP mobile platform; see Fig. 2. The WAM is appealing due to its low total and moving mass, back drivable joints, and high-efficiency cable transmission. Some drawbacks are the need to periodically exchange worn cables and the need to (sometimes) use an external control computer. Important for mobile applications is that the motor controllers and amplifiers are integrated into the joint motors, doing away with external amplifier cabinets altogether. One of the most appealing, and as far as we know unique aspects is that the manufacturer provides an 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. It is unclear how the built-in balancing controller could be used together with contact manipulation (and unfortunately, unlike the WAM, it is not open source and user modifiable¹), so for the moment we use it in tractor mode with a caster wheel and use the RMP statically balanced. The caster wheel is front-mounted just inside of the reach of the WAMs to give a stiff forward support point when lifting objects.

The WAM acronym stands for "Whole Arm Manipulator." Unlike conventional arms, it is designed to exert contact not only with the end effector, but also with any link surface to grasp objects, for instance, for greater force and contact area between the forearms (Fig. 2). This is possible due to the low transmission losses, where motor currents can be used to calculate external forces without the need to put force sensors everywhere on the arm. We performed kinematics simulation to study the resulting common workspace and

¹the WAM provides full access to the control source code for the robot which makes it possible to write your own controller without having another control loop running under or parallel to it. The Segway RMP, however has a control loop always running behind any custom controller

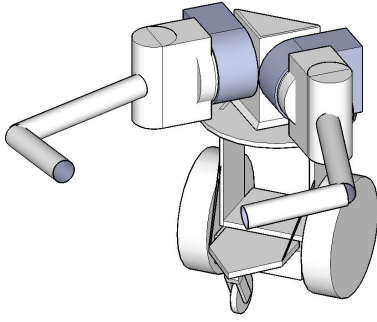


Fig. 1. CAD model of assembly

possible motions for different angular and displacement offsets between the two WAM arms. The simulation data was visualized in more detail with a solid CAD model. We found that mounting the arms vertically at a 90 degree offset, as shown in Fig. 1, gives a substantial overlapping workspace for two-arm interaction. This configuration also makes this workspace resemble that of a human [15], where there is no overlapping space on the back, but a great deal of it in the front. A lightweight mount was built from plywood. The arm assembly can be set either at the height of the standard Segway platform to optimize reach on standard 70 cm table tops, or alternatively set just above the Segway fenders on a shortened mount to improve the reach for manipulation of objects on the floor. This configuration allows the robot to pick up large objects from the floor, as shown in Fig. 5. The overall dimension of the robot is comparably compact. However the arm mounting makes the shoulders slightly wider than a standard single door frame.



Fig. 2. WAM whole arm contact manipulation

We integrate the computers and batteries as low as possible, in the space between the wheels on the Segway. The current configuration consists of two PC computers for vision and sensing; one pc104 computer is integrated into the arm for motion control, and a 4-port wireless router connects the on-board computing with a local ethernet and allows user-control from a wireless laptop or tablet. The platform supports different sensing configurations. For visual-servo motion control, cameras can be mounted on the forearm.

TABLE I
POWER CONSUMPTION SPECIFICATION FOR THE SYSTEM

Subsystem	Idle	Typ Load	Peak
WAM 4 DOF	18W	28W	600W
WAM 7 DOF	27W	45W	800W
2 Computers	70W	150W	190W
Cameras etc	$\leq 10W$	$< 10W$	$\leq 10W$
Segway RMP	Integrated battery		
TOTAL	$\leq 125W$	$< 250W$	$\approx 1600W$

In Fig. 2, one mount is visible on the right arm (without a camera attached). A pan-tilt unit with a stereo camera head can fit on top of the arm platform. The robot will support research in visual and force-based tele-operation. A hardware diagram is shown in Fig. 3.

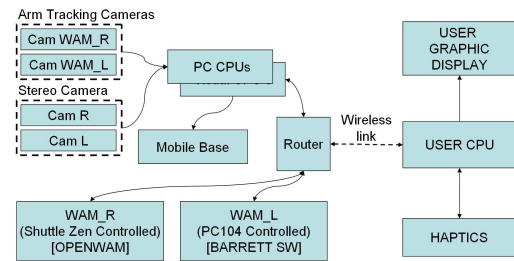


Fig. 3. Hardware setup.

IV. POWER

A mobile manipulator contains numerous subsystems that need to be powered. For less complex systems, the most convenient solution is often to buy battery-powered subsystems, e.g. laptops, powered sensors, etc. In our case, we would like to minimize the number of systems that need to be separately charged, and thus equalize run time of the subsystems. This favors a single-power-source system and power bus.

We use universal AC/DC line voltage power supply units, which adds the convenience of direct switching between AC line voltage when in lab use, and battery power when going beyond cord reach. By testing and examining the circuitry we found that most power supplies would run also on DC, in particular the ones featuring rectification up front of the circuit. We tested the rectification diodes in the most critical PSUs (e.g., WAM arms and computers) for overheating. The diodes were working within the safe temperature range specified for the components. We evaluated the use of a DC-DC converter, but it would have introduced the need for an AC-AC conditioning and rectification stage, which would imply the need of more components to factor into the design.

The main power needs are depicted in Table II and the power consumption for idle, engaged and peak stages is shown in Table I.

Different battery chemistries, namely Lead Acid, NiCd, NiMH, and LiION, were evaluated considering capacity, energy density, depth of discharge, inner resistance, recharging cycles and price. We define robustness on the batteries as their tolerance to overcharge, deep discharge, mechanical

TABLE II
SPECIFICATION OF THE POWER SOURCE FOR THE ROBOT.

Characteristic	Needed	Achieved
Running time idle	5 hours	9 hours
Running time typ load	3 hours	≈ 4.5 hours
Weight	≤ 30kg	24Kg
Nominal Voltage	≥ 100V	144V
Max. Discharge Current	11A	25A
Max Peak Load	1.5KW	3.6KW

shock and impacts. We chose NiCd for the combination of robustness and price. In our configuration we are using four 36 V battery packs, each composed of 30 1.2 V-8 Ah High Capacity NiCd. This powers the entire system excluding the RMP Segway Platform. The total nominal capacity of the power supply is 144 V * 8 Ah=1152 Wh. The system has been tested to run until deep discharge of the batteries, having 96 V of output voltage².

V. COMPUTING HARDWARE

Computing hardware found in lab robots range from specialized embedded solutions to standard laptops. A popular choice for the former would be the pc104 series, which has a compact size and lots of expandability options, while retaining the standard PC x86 architecture. However, pc104 boards do not feature the most powerful recent processors. While fine for most motor control applications, processing high bandwidth data, such as from vision and other sensing, would be too taxing. High-end laptops come with processors nearly as powerful as desktops, and are conveniently already integrated with batteries into a small package. The main drawback is that expandability options are limited to one or at most two cardbus or PCI-E1 cards.

For our mobile manipulator, we chose to custom build PC workstations for a variety of reasons. In recent years, Intel and AMD have unified processor lineups making PC and workstations basically the same. In different applications, they will be tuned to different power consumption by varying voltages and frequencies (CPU, FSB, etc), however a suitable motherboard (MOBO) will allow adjustment of most parameters. A main advantage of a PC-based solution is the expandability. Moreover, desktop processors are available as quad-core and even hex-core, while the great majority of laptops contain dual-cores, and only a small number of them are quad-core. Table III shows a comparison of the tested systems.

The final computing solution was built from standard mATX PC components, as it is possible to find 13- and 14-inch wide standard cases, making the building and later access to the boards much more convenient. We chose the Foxconn G33M-S for our final build, with low-power integrated graphics. In addition to using one or two high

²Protection circuitry in PSUs prevents the unit from starting if the initial voltage is below 100V, but allows the unit to remain functional well below this limit until the power demanded from the PSU exceeds its nominal output, this is why we were able to test the system when the power bus was below 100V DC.

TABLE III
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 1975
CPU freq GHz	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

performance PC's for data processing (currently only one), a low-power PC104 board is dedicated to the real-time motion control of one of the arms, and the other arm is controlled by a Shuttle Zen³. For higher performance, an Intel i7 or i5 solution could be used. However we have yet to find motherboards with integrated graphics for these, and a discrete graphics card would add significant power consumption. Among PC MOBOs, we found that few would boot without graphics installed. Server boards often will, and we ran the 1975 boards headless for some time. However, we found it useful to have the ability to hook up a monitor for debugging, so having built in graphics is an asset. For image acquisition, we incorporated two IEEE1394b boards on the computer to receive images from high frame rate (200 Hz VGA) PtGrey Dragonfly express tracking cameras.

VI. SYSTEM SOFTWARE

When using off-the-shelf components, it is inevitable that different system components use different computational architectures. The computing loads required for vision tasks and robotic control tasks are also too great for a single computer. We use Parallel Virtual Machine (PVM) to distribute computing over multiple computers. The use of PVM significantly decreased development time by allowing us to bypass architectural and network issues.

Software should be developed in a framework to guarantee robustness, flexibility, maintainability and platform portability. In this sense we are building upon a component-based approach [7]. Using a component based framework ensures that the rules can be set for developing software that tolerates both hardware updates and future versions of drivers and applications by reusing most of the already existent code.

In Fig. 4, the software architecture is shown. The principal building blocks are shown as solid blocks, and they are grouped into semantic super-modules. The latter are contextually grouped in terms of what functions they cover and not on what block type (horizontal, vertical or application according to [7]) they are. This grouping gives

³this is because the two WAMs present in our lab are from different generations and driving them from the integrated PC104 is still being analyzed, hence each arm has it's own controller and control software

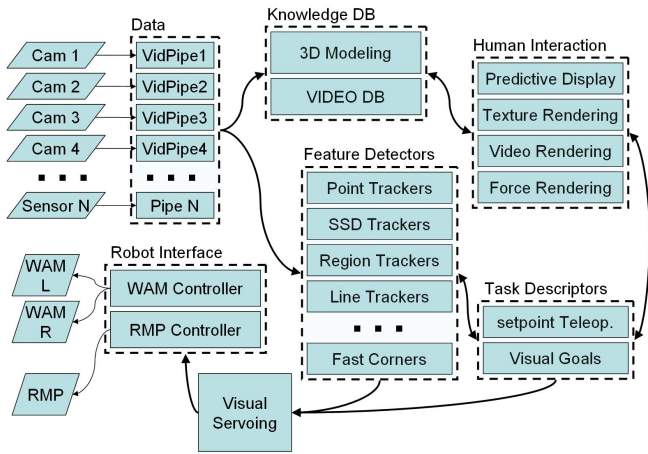


Fig. 4. Software setup.

a better understanding of the information flow across the system. On a more detailed level, all of these super-modules have all three types of blocks, ensuring the modularity and consistency of the software run in the system. To ensure the robustness and safety of the system, we also plan to include the ideas outlined in [4] (an extensible framework to easily and systematically test software by applying reuse and modularity).

As mentioned, our system is heterogeneous and consists of the PCs and a PC104 board. Our PVM setup was designed such that one master process controls several worker processes that operate the robot's different components. Computationally heavy vision tasks are run on the PCs under Fedora and the less demanding but time-critical motion control is run on the PC104 under real-time Linux. We use C/C++ libraries for sensing and control, including XVision [10] and VISP [19]. However, recompiling for each change or test is inconvenient. We use Matlab for easy prototyping and scripting of new mobile manipulator behavior, while MEX allows us to wrap fast and efficient C code once the prototype software module is fully functional.

In general the control of the robot is built in different layers, of which the closest to hardware control run on the innermost loops, whereas the controllers for high level processes such as vision and display run in the outermost loops. Also, the loops that are closer to the hardware, have higher running speeds, *e.g.*, the WAMs run at 750Hz and 1kHz, on the PC104 and Shuttle Zen respectively, while the visual servoing controller runs at 60Hz maximum (and in this case is not used but has been already tested individually). This layered approach allows for expandable controllers with increasing capabilities to achieve more complex tasks.

To cope with safety issues, the system will run daemons for each subsystem that will monitor the conditions of operation and in case of malfunction, shut down the monitored module, automatically attempt to reboot the subsystem and, if failure occurs, asses the malfunction if possible. At the present moment, the WAMs are the only subsystems featuring safety shutdown.

VII. SYSTEM INTEGRATION AND TEST

The software integration is done by using PVM and by following component-based framework directives, which in turn makes integration straight forward at the software level, although this is not always the case for the hardware. Fig. 3 shows the future hardware configuration⁴. On the left of the diagram, all components connected by solid lines are wired to a wireless router. The same happens to the user graphic display, user cpu and haptics modules on the right of the diagram. The dashed line stands for the wireless link between the user side and the robot's side. In this system, all the major computations are done onboard the robot, whereas the visualization and task specification is processed on the user side and then the commands are sent over to the robot through the wireless link. At the moment, all these systems are networked together using the wireless router mounted on the robot.

In order to test the integrated system, we used the robot to pick up a box off of the floor (Fig. 5). To achieve this we recorded taught trajectories and replicated them to pick up the box. The WAMs are taught the trajectories through direct human manipulation of the arms. The ability of a human to physically interact with the arms is rare among robotic manipulators, making the WAMs particularly well suited to our application. To this moment, visual servoing and task specification modules were not included in the test, as further hardware integration tests are needed in order to ensure the system's propagation times and data availability are suitable for all the control processes running on the robot.

During the test, the computation was distributed across the PCs. A 64-bit Linux machine ran a Segway worker process in charge of controlling the robot and a vision worker for visual tracking. This latter process was not used but it is considered for near-future experiments and has been tested for visual servoing applications. The other two machines were in charge of controlling the WAMs using a modified gravity compensation scheme. The WAMs have different configuration, so different software was used to control either arm. One arm was controlled by a worker process using OpenMAN [16], and the other was controlled by a worker process using Barrett's software API⁵. The control machines run Realtime Linux system running RTAI and XENOMAI.

The operator has control of the system from a wireless laptop running cygwin using a Matlab Mex interface.

VIII. CONCLUSIONS AND FUTURE WORK

Designing and building a mobile manipulator used to be a complex task reserved for those with access to engineering facilities. The recent availability of lightweight robot arms with built-in joint motor amplifiers makes it possible to integrate purchased arms with a commercial mobile base.

⁴The only significant difference in our future configuration is that the controller for the WAM arms in this case will be residing in the same computer. Currently, each WAM has its own computer running the control interface for each arm

⁵In a near future we plant to control both WAM arms using a single computer and a single driver

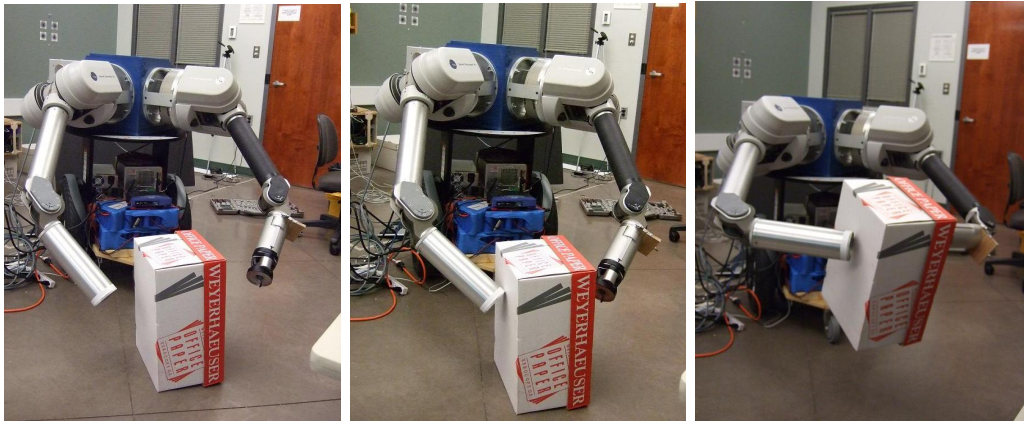


Fig. 5. Two arm manipulation: Lifting a box

We investigated computing, battery powering and distributed computing software, and how to integrate these components to build a mobile manipulator.

In the future, we aim to use the mobile manipulator to research semi-autonomous vision and contact-sensing behaviors in tele-robotics. Such higher level routines can relieve the tele-operator from tedious detailed motion control and elevate the user interface from motion level to task level. Furthermore this approach is likely to be less sensitive to the inevitable time delays in tele-manipulation.

In order to facilitate tele-robotic control of the WAMs, we plan on using a Phantom Omni haptic device. This was tested on one arm, and we plan on extending it to both in order to provide straight-forward teleoperation. Using the Phantom, we can directly telemanipulate the joint angles of the WAM or use it as a teaching method to create primitive motions. Our last objective is to issue deictic commands, using a visual interface in a hand-held device or a workstation, to the robot, e.g., instructing a peer to do certain task only by a minimal, high level description.

ACKNOWLEDGEMENTS

We would like to thank a laboratory colleague Simon Leonard for the use of OpenMAN 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.
- [2] A. Albers, S. Brudniok, J. Ottad, 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. Diffler, 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] S. Bensalem, M. Gallien, F. Ingrand, I. Kahloul, and N. Thanh-Hung. Designing autonomous robots. *IEEE Robotics Automation Magazine*, 16:67–77, 2009.
- [5] C. Borst, T. Wimbock, F. Schmidt, M. Fuchs, B. Brunner, F. Zacharias, P. R. Giordano, R. Konietschke, W. Sepp, S. Fuchs, C. Rink, A. Albuschaffer, and G. Hirzinger. Rollin justin - mobile platform with variable base. In *ICRA*, 2009.

- [6] R. Brooks, L. Aryananda, A. Edsinger, P. Fitzpatrick, C. Kemp, U.-M. O'Reilly, E. Torres-Jara, P. Varshavskaya, and J. Weber. Sensing and manipulating built-for-human environments. *International Journal of Humanoid Robotics*, 1(1):1–28, 2004.
- [7] D. Brugali and P. Scandurra. Component-based robotic engineering (part i) [tutorial]. *IEEE Robotics Automation Magazine*, 16(4):84–96, 2009.
- [8] 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.
- [9] A. Edsinger-Gonzales and J. Weber. Domo: A force sensing humanoid robot for manipulation research. In *Humanoids*, volume 1, 2004.
- [10] G. Hager, S. Puri, K. Toyama, and D. Burschka. A brief tour of xvision. —<http://www.cs.jhu.edu/CIPS/xvision/index.html>, 2010.
- [11] 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.
- [12] 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.
- [13] D. Katz, E. Horrell, Y. Yang, B. Burns, T. Buckley, A. Grishkan, V. Zhylykovskyy, 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.
- [14] O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holmberg, and A. Casal. Vehicle/arm coordination and multiple mobile manipulator decentralized cooperation. In *IROS*, volume 2, 1996.
- [15] N. Klopčar and J. Lenarčič. Kinematic model for determination of human arm reachable workspace. *Meccanica*, 40(2):203–219, 2005.
- [16] S. Leonard. openman: An open source C++ toolbox for control and simulations of manipulators.
- [17] T. Morita, H. Iwata, and S. Sugano. Human symbiotic robot design based on division and unification of functional requirements. In *ICRA*, volume 3, 2000.
- [18] U. of Massachusetts 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.
- [19] 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.
- [20] C. Smith and H. Christensen. Robot manipulators. *IEEE Robotics Automation Magazine*, 16, 2009.
- [21] 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, 2009.
- [22] M. Stilman, J. Olson, and W. Gloss. Golem krang: Dynamically stable humanoid robot for mobile manipulation. In *ICRA*, 2010.
- [23] Willow-Garage. Technical specifications. <http://www.willowgarage.com/pages/robots/technical-specs>, January 2010.