WHOLE-BODY STRATEGIES FOR MOBILITY AND MANIPULATION

A Dissertation Presented

by

PATRICK DEEGAN

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2010

Computer Science

WHOLE-BODY STRATEGIES FOR MOBILITY AND MANIPULATION

A Dissertation Presented

by

PATRICK DEEGAN

Approved as to style and content by:

Roderic Grupen, Chair

Oliver Brock, Member

Andrew Barto, Member

Richard Van Emmerik, Member

Andrew Barto, Department Chair
Computer Science

To my family.

ACKNOWLEDGMENTS

I would like to thank my thesis advisor, Rod Grupen, for all his help, leadership, and support throughout my graduate school career. Rod's vast knowledge, genius, and insight into robotics has been a constant inspiration. I feel very fortunate to have been a part of the Laboratory for Perceptual Robotics. The environment Rod created not only allowed me to develop my passion for robotics, but his understanding and patience allowed me to grow personally, overcome many obstacles, and develop the confidence I needed to bring this work to fruition.

I would also like to thank my committee members for their inspiration and assistance: Oliver Brock, for always seeing through to the essence of things and subsequently guiding me to raise the standard of my work; Andrew Barto, for his support of the many directions that the uBot development took; Richard Van Emmerik, for his enthusiasm and insightful perspectives that led me to explore the uBot's potential to exploit dynamics as it related to human motor coordination; and Paul Utgoff, for his academic mentoring and for making me aware of the beauty of computation.

I am fortunate to have worked with many talented graduate students. I am especially grateful for Ed Hannigan and Dirk Ruiken. Without their hard work and dedication, the uBot would not have become a reality. Their humor and lightheartedness made many late night demos and development sessions very memorable. I feel very fortunate to have participated in several high profile uBot demonstrations, the success of which is a direct reflection of Ed and Dirk's paramount skill and competence.

I am very grateful to the UMASS Computer Science Department, faculty, staff, Priscilla Coe, and especially Leeanne Leclerc. I owe a debt of gratitude to Leeanne and without her encouragement and administrative magic, I would not have been able to pull this off.

I would like to thank my close friends: Tim Schwider, Kimberly Martin, Jonathan Livingston, and Lindsey Constantz. It is because of their love, faith, and support that being able to say, "I've finished" was one of the greatest moments in my life.

ABSTRACT

WHOLE-BODY STRATEGIES FOR MOBILITY AND MANIPULATION

MAY 2010

PATRICK DEEGAN

B.S., HARVEY MUDD COLLEGE

M.S., UNIVERSITY OF MASSACHUSETTS AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Roderic Grupen

The robotics community has succeeded in creating remarkable machines and task-level programming tools, but arguably has failed to apply sophisticated autonomous machines to sophisticated tasks. One reason is that this combination leads to prohibitive complexity. Biological systems provide many examples of integrated systems that combine high-performance and flexibility, with logically-organized low-level control. Sophisticated organisms have evolved that depend on physical dexterity to thrive in a particular ecological niche while mitigating computational and behavioral complexity.

This dissertation investigates the potential for a new kind of hybrid robotic design process. A design for performance that combines mechanical dexterity with low-level embedded firmware that organizes behavior and facilitates programming at a higher

level. I propose that dexterous machines can incorporate embedded firmware that express the "aptitudes" implicit in the design of the robot and hierarchically organize the behavior of the system for programming. This is a win-win situation where the quality of the embedded firmware determines how efficiently programmers (autonomous learning algorithms or human programmers) can construct control programs that are robust, flexible, and respond gracefully to unanticipated circumstances.

This dissertation introduces the uBot-5—a mobile manipulator concept for human environments that provides dexterous modes for mobility and manipulation and control firmware that organizes these behavioral modes locally for use by applications code. Postural control underlies the uniform treatment of several mobility modes that engage different combinations of sensor and motor resources. The result is a platform for studying "whole-body" control strategies that can be applied jointly to simultaneous mobility and manipulation objectives.

The thesis examines the specification and development of both: (1) a dexterous robot for unstructured environments, and (2) the embedded firmware that organizes dexterous behavior for mobility and manipulation tasks. Integrated solutions are proposed that control transitions between postural "modes" and provide a logically organized dexterous behavior hierarchy. Firmware programming can also be used to construct an efficient API for user programming and autonomous machine learning.

My goal is to contribute technologies that can support new robotic applications in our culture that require fully integrated dexterous robots in unstructured environments. Personal robotics is an important emerging application that depends on seamlessly integrated and sophisticated machines, controllers, and adaptability. Logically organized representations for use in task-level application development are critical to pull this off. The impact of such technology could be significant—with applications that include healthcare and telemedicine, exploration, emergency response, logistics, and flexible manufacturing.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure **Page**

CHAPTER 1
INTRODUCTION

The state of the art in robotics presents many examples of very capable machines that do such things as vacuum floors, assemble cars, or assist in surgery. These systems provide significant benefits that include reducing the burden for humans in dull, dirty, and dangerous tasks, increasing efficiency and quality in factory and supply chain operations, and augmenting human performance and situational awareness in remote and inhospitable environments. However, most robots today are single purpose. They are designed to do specific tasks and often require significant integration and well engineered environments in order to perform well. To advance the state of the art, new designs are required for flexible, autonomous, and *dexterous* robots that are capable of performing multiple tasks in multiple domains.

Dexterity refers to agility, proficiency, and resourcefulness when rapidly modifying a solution in light of new run-time constraints, by effortlessly switching between multiple strategies. This flexibility is due to excess mechanical degrees-of-freedom and controls that can exploit (continuous and discrete) alternative solutions.

Mobile manipulators are robots that can move about and manipulate their surroundings. Whole-Body Mobile Manipulators (WBMM) are able to engage sensor and motor resources in service to mobility and manual interactions with the world. For instance, a WBMM with adequate resources might be able to simultaneously turn a knob, lean into a door, and walk through the entry way. The same system may also be able to pick items from a floor, push a vacuum cleaner, and carry out the trash. More flexible mechanical structures can lead to more interesting possibilities.

The cost of flexibility is increased complexity. That is, coordinated motion with excess degrees of freedom requires the mastery of many units of action. Although dexterous WBMMs offer flexibility, the complexity of controlling multiple independent degrees-of-freedom is reflected by an explosion in the number of states and actions and making them difficult to program.

This thesis presents a comprehensive approach to the design and development of a unique whole-body mobile manipulator including decisions for mechanisms, embedded sensing, and computational architectures. A robot represents a form of embodied computation—a constructive relationship between form, sensing, computation, and control that captures the "aptitudes" of the machine. I aim to demonstrate an integrated design for a dexterous robot that is easy to program.

1.1 Designing for Flexibility

Foremost among the criteria required for dexterous mobile manipulators in unstructured worlds is a rugged, untethered design that can support exploration and the freedom to "make mistakes" without severely damaging the robot. Safety is a related concern. Compliance (both passive and active) is an important means of minimizing impact forces. This was an important criterion influencing my choice of a dynamic balancer. Among other attractive features of the inverted pendulum, it provides a low input impedance that minimizes the magnitude of collision forces.

A dynamically stable platform is characterized by a degree of freedom close to the ground plane that allows the robot to comply to environmental forces and to exploit body dynamics to increase performance in some tasks. This design point, in particular with respect to dynamic stability, is a hallmark of biological and evolutionary design. Athletes in the animal kingdom exploit body dynamics to yield high performance despite the potential risks. Consequently, a primary goal of this work is to prototype a machine subject to a set of functional specifications and to demonstrate the capacity to address safety and stability by combining sound mechanical design with appropriate embedded control architectures.

1.1.1 Compute Architecture

Dexterous machines require complementary flexibility in a hierarchical compute architecture. The embedded control system must provide the bandwidth required for high-performance feedback control to produce smooth, reactive, coordinated body motion. At the lowest level, closed-loop motor units have minimal computational requirements but require high update rates and low latency between sensors and actuators. Higher level sequential controllers may require lower update rates and may tolerate increased latency.

In general, special-purpose hardware is required to support high-performance embedded control, while high-level applications and programming may be best served by processing available in today's laptops and personal computers. Moreover, many applications employ high bandwidth sensors such as cameras for video processing, localization, and mapping. Yet, these applications require compute support from off-board networks of processors. Thus, mobile manipulators will require a computational hierarchy that extends into control and perceptual processes supported in the network.

1.1.2 Embedded Control Architecture

As I argued earlier, careful specification of the physical structure and a complementary compute architecture is required to exploit complex, dexterous robot designs. Such a design leverages native structure to maintain flexibility. This architectural principle is illustrated by many biological systems. For example, a prey animal, like a horse, can stand and walk within thirty minutes of birth. A genetic commitment to native structure allows the animal to master these tasks quickly in order to survive. Appropriate combinations of mechanical, perceptual, and computational flexibility combined with native behavioral structure can promote viability in unstructured environments for robots as well, while yielding a useful organization for application development. This is the sweet-spot I attempted to realize in uBot-5.

At the lowest-level in the hierarchical compute architecture built into uBot-5, control firmware is used to implement several kinds of homeostatic behavior to preserve safe operation and organize low-level behavior. The firmware also establishes a baseline performance from which higher-level control functions expand competency and skill. A central contribution of this thesis is a suite of low-level control firmware that supports discrete modes of postural stability and the transitions between modes. This firmware overlays control logic on the uBot-5 to yield high performance, minimize the risks of a dynamic balancer, and provide a convenient means of compiling hierarchical control programs.

1.1.3 Postural Stability Control Suite

WBMMs can combine resources normally reserved exclusively for mobility or manipulation to configure several hybrid modes as conditions require. For instance, in the transition from crawling to standing, an infant is able to free its arms by developing control that replaces statically stable structures with active, dynamically stable mobility modes. The commitment to dynamic stability in uBot-5 requires an equal commitment to an array of embedded control circuits for managing postural modes and their transitions. Skills for preserving postural stability will become an aspect of virtually all subsequent application code and guaranteeing stability will influence virtually all system behavior.

uBot-5 achieves high performance, to a significant degree, by embracing intrinsic instability in the machine. This design requires an array of homeostatic stability controllers to make sure that the robot does not damage itself or the environment and to keep from getting into states from which it cannot recover. One of the contributions of this thesis is a family of domain independent reflexive stability controllers. This firmware will influence all future application code.

1.2 Thesis Contributions

This thesis describes the design and implementation of a scientific apparatus, the uBot-5, to facilitate application development and research into whole-body mobile manipulation technologies. Such a machine can serve as a catalyst for software and control development and can be used to realize skills that support multiple applications.

Software has been developed to leverage the dynamics of the whole body to enhance performance, these include: postural modes for prone, 4-point, 3-point, and 2-point stances, and transitions between them. Furthermore, an emergency transition to protect against uncontrollable transitions to prone is also contributed such that if balancing fails, the arms are automatically reconfigured to absorb the energy of a fall and restore postural stability. Multiple mobility and manipulation modes ensure performance in a wide variety of contexts. The transitions between modes form a comprehensive policy for managing whole-body dynamics that yields a basis for high-performance applications. The robot, its control framework, and the postural stability suite of controllers are presented. Several real world applications demonstrate that the uBot-5 explores new territory for mobile manipulation.

The three main contributions of this thesis are: (1) the methodology and implementation of a hardware and software co-design for a novel, dynamically balancing bimanual mobile manipulator, (2) an integrated array of reusable and task independent embedded firmware that express the aptitudes of the design including: vestibular reflexes and whole-body impedance control, (3) a hierarchical native control structure that expresses a comprehensive suite of sequential behaviors including controlled transitions between postural modes from prone to upright with hands free and recovery strategies for when postural stability is compromised. The uBot demonstrates these features in a variety of tasks that exploit multiple engagements of the robot's degrees of freedom including: bracing and recovery from a fall, pushing, throwing, bi-manual grasping, and transporting. Furthermore, these contributions form the basis for an integrated high performance mobile manipulator, well suited for service as a research apparatus in unstructured environments.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

In the past, robots have been successful in highly-structured industrial applications where they have revolutionized manufacturing, assembly, packaging, and logistics (supply-chain and distribution). These environments are typically engineered to eliminate uncontrollable circumstances. State of the art industrial applications depend on robots with exacting mechanical, electrical, and software specifications that perform flawlessly in highly structured operating conditions.

There are fewer examples of commercial systems that can cope autonomously with unstructured environments. Remote telepresence is being used fruitfully in dull, dirty, and dangerous tasks; to perform emergency response, to dispose of roadside bombs, and to conduct surveillance. However, these systems rely on humans in the loop and current methods for handling latency and exploiting redundancy are inadequate. Better applications interfaces are required at levels of abstraction that hide all the high-bandwidth embedded processing related to homeostasis.

It is my view that hybrid design methodologies that combine dexterous mechanical concepts with logically organized embedded firmware have not received an adequate amount of attention in the research community. The performance of biological systems depends on interdependent components and often, it is unclear whether one subsystem precedes or follows another in the anthropological record. Practical designs for dexterous robots will likewise depend on integrated co-design and will find added utility in the principles of integration—sometimes to the exclusion of incorporating the most sophisticated components. More functional machines will catalyze new research in controls and knowledge organization that do not receive enough attention today.

2.1 Historical Milestones in Mobile Manipulators

2.1.1 Mobility and Manipulation

The Stanford Assistant Mobile Manipulator (**SAMM**) (Figure 2.1) utilized the XR4000 with a PUMA 560 arm and a parallel-jaw gripper [25]. The XR4000 is a holonomic base with four powered wheeled casters that provide smooth and accurate motion. The holonomic base allows full use of the null space to improve workspace and makes gross motion planning easier. Coordination between multiple cooperating mobile manipulators is accomplished by exploiting the precision of the base and controlling the arm motions using operational space control. This mechanical geometry is inspired by the desire to make a motion planner work better, that is, the mechanism is in service to the algorithm.

Figure 2.1. The Stanford SAMM

The UMass **UMan** (Figure 2.2) uses a Barrett Whole Arm Manipulator (WAM) 7 degree-of-freedom arm by Barrett Technologies (Cambridge, MA) [1] with a 3-finger, 4 DOF Barrett hand also mounted to Nomadic XR4000 mobile base [32]. It is designed to support the development of a multi-objective, task-level control framework. The WAM is back-drivable, allows position or force control of each joint, and it houses most of the large motor masses within its base (with a cable driven design) for highly dexterous, dynamic, and accurate motion. The UMan, thus, increases the workspace, backdrivability, and dexterity of the

Figure 2.2. The UMass UMan

SAMM robot. Both systems employ planners and motion controllers to reduce the probability of collisions and neither robot can respond passively to inadvertent bumps and collisions. UMan's WAM is capable of low impedance (backdrivable) modes but the XR4000 base is not. The XR4000 platform is relatively tall and heavy and therefore, some platform postures and accelerations can destabilize the base.

The Willow Garage **PR2** (Figure 2.3) is based on the PR1, a concept for an intrinsically safe personal robot [60]. PR2 has an omni-directional wheel base, a telescoping spine, two force-controlled 7 degree-of-freedom arms, and an actuated sensor head. Each arm has a 1 degree-of-freedom gripper attached to it. PR2 has demonstrated several tasks within a hierarchical control framework including autonomous navigation, door opening, and recharging using standard electrical outlets [39].

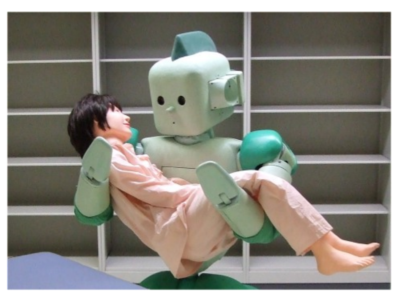
Figure 2.3. The Willow Garage PR2

2.1.2 Humanoids

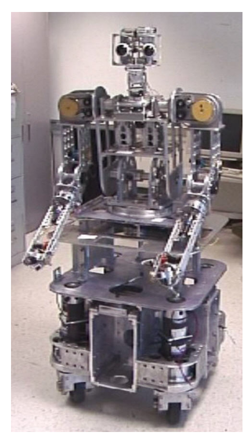
a) Statically Stable

Figure 2.4 shows several statically stable mobile manipulator concepts. The **AR-**

(a)



(b)



(c)

Figure 2.4. Several statically stable mobile manipulator platforms including: (a) The ARMAR robot, (b) Dav, and (c) RI-MAN.

MAR [8] robot from Forschungszentrum Informatik Karlsruhe (FZI) features a differential drive, wheeled, and statically-stable base, a 4 DOF body, and two 7 DOF arms with parallel jaw grippers. It is designed to be mobile in a kitchen setting and careful enough to handle plates. The humanoid robot, **Dav** [18], developed at Michigan State University (MSU) is an experimental platform combining a non-holonomic mobility platform with a torso, arms, hands, neck and head. The **RI-MAN** robot has two 6 DOF arms capable of lifting 80lbs. It is completely covered with a human friendly soft skin and tactile force sensors. The robot is capable of supporting human care and welfare tasks.

b) Legged

Dynamically stable platforms have demonstrated the viability of mobile manipulators in human environments. Legged humanoid robots with a similar morphology to an average human may be ideally suited for tasks in this domain. Unlike statically stable systems, legged bases do not need to be large or heavy in order to be stable. With the ability to pivot near the ground contact point, this type of robot can be modeled by an inverted pendulum and can accommodate large masses mounted high above the ground plane. Thus, dynamically stabilized platforms can position sensor packages where they are most useful in human environments and can transport relatively large payloads.

Sony **QRIO** (Figure 2.5) is a small biped entertainment robot. The safe design incorporates round-shape surfaces, small size, and a joint structure that eliminates pinch points [36]. Whole body cooperative motion control utilizing ZMP control allows the QRIO to resist external forces and walk on irregular surfaces. The robot can also fall over and right itself. Walking motion is generated by a gait pattern generator

Figure 2.5. The Sony QRIO

with feedback from inertial and force sensors to adjust the posture of the robot to terrain inclination and external forces.

After more than sixteen years of development, The Honda **ASIMO** (Figure 2.6), implements ground reaction force control, model Zero Moment Point (ZMP) control, and foot landing position control, to maintain stability on a wide variety of surfaces while subject to unexpected perturbations [23]. ASIMO is capable of climbing stairs and navigating in man-made environments such as offices, museums, and hospitals. Notably, the humanoid has demonstrated one of the hallmarks of dynamic locomotion and human-like running [51, 55]. ASIMO employs a whole body posture control that utilizes the center of mass and joint torques to maintain stability. Most of the ASIMO's development was spent perfecting legged locomotion without significant attention being placed on its manipulation abilities.

Figure 2.6. The Honda ASIMO

Figure 2.7. The Kawada Industries HRP-2.

The **HRP-2** [31] is a humanoid developed by Kawada Industries and has demonstrated cooperative work with humans in carrying one end of a panel or table. The HRP-2 is also capable of walking on uneven surfaces, and notable for a robot its size (154cm tall and 58kg Mass), getting into and standing up from the prone position.

Figure 2.8. Hitachi Emiew 2

The Hitachi **EMIEW2** (Figure 2.8) use a hybrid wheeled leg. The "feet" of this robot are composed of controllable drive wheels with deployable supports. This allows the robot to dynamically balance in either of three modes: legs only, legs and wheels, and only wheels. The robot primarily utilizes the wheeled mode and the legs assist the platform over uneven terrain. The EMIEW2 was specifically designed to operate safely around humans, incorporating a small size and obstacle avoidance technology. For office environments, the robot uses voice communication functions, and provides necessary services such as delivering drinks or documents.

2.1.3 Dynamically Balancing Mobile Manipulators

Figure 2.9 shows several dynamically balancing wheeled bimanual manipulators. **JOE** [17] is an example a of "hobby" platform that has demonstrated the agility,

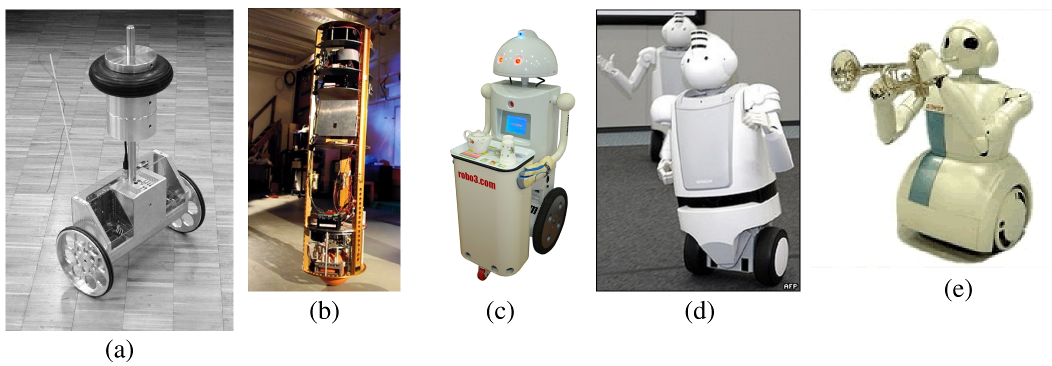


Figure 2.9. Several wheeled balancing platforms including: (a) The JOE robot, (b) CMU ballbot, (c) ROBO3 ROBIN, (d) Hitachi Emiew, and (e) the Toyota rolling partner robot

speed, and stability that dynamically balancing platforms can deliver. **Ballbot** [38], is a holonomic platform that balance around a single spherical wheel. However this configuration may be at a disadvantage when traveling on uneven terrain and in practice, its dynamics do not allow the platform to change directions instantaneously. The **Robin R3** [3] is a commercial dynamically balancing platform that can deliver small payloads. However, the dexterity of the arms is very limited. The Hitachi **Emiew** [2], the predecessor to EMIEW2, favored a two-wheeled platform with a hip joint to allow leaning into turns for added stability. The **Toyota Partner** rolling robot [5] features dexterous arms and hands with the ability to play musical instruments.

The Segway® Robotic Mobility Platform (RMP) [43] is based on a Segway Human transporter. Compared to existing robots, the reconfigured platform is more agile and less expensive, and it can carry larger payloads. Unfortunately, part of this repurposing required that a large mass be mounted on the top of the platform to ensure a high center of mass for control stability. The resulting RMP (without a manipulator or instrumentation attached) weighs 57Kg or more [53]. This can present a serious danger to both the robot's environment and the robot itself as the robot generates very large impact forces if it falls over. Platforms such as these rely on special purpose hardware, such as kickstands to prevent falls or damage. Typically, however, this hardware is only for emergency, and not for routine use [43].

The MIT **Cardea** (Figure 2.10) consists of a force controlled Series Elastic Actuator, (SEA) arm [49] on top of a balancing Segway RMP [10, 59, 14], to take advantage of the inherent compliance and small footprint of dynamically balancing platforms. Cardea has a single manipulator and no end effector to address tasks that require substantial force controlled motions. As a proof of concept, Cardea applied a behavior based approach to tasks including corridor navigation and door finding, opening, and pushing. However, hardware limitations, and less well integrated perceptual mechanisms require further work.

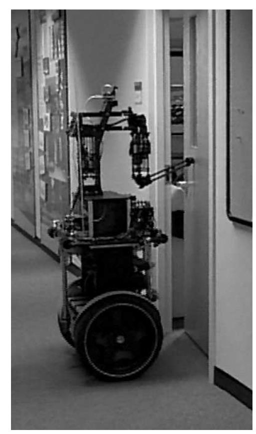


Figure 2.10. The MIT Cardea

Figure 2.11. The Stanford STAIR

The **Stanford AI Robot** (STAIR) [52] utilizes a 4kg, 5-DOF arm equipped with a parallel plate gripper, and can lift a 500g payload. The arm is precisely controlled and it can move to positions triangulated by a webcam vision system. The manipulator is also mounted to a Segway RMP adapted for statically stable balancing with an omnidirectional caster mounted to an outrigger support. Moreover, both Cardea and the Stanford AI Robot represents a dramatic contrast in mass and power distribution, with a large, massive, powerful base supporting a significantly smaller, lighter, and less powerful upper body. As as result, the mobile manipulator concepts are not capable of exploiting the whole body as a dexterous mobility and manipulation platform.

(a) (b)

Figure 2.12. (a) The NASA Segwanaut and (b) NASA Centaur

The JSC **Robonaut** can be configured as a wheeled balancer using the Segway RMP (Figure 2.12(a)). Robonaut has over 40 degrees of freedom designed specifically for operations in space with EVA equipment designed for human beings. This platform has demonstrated navigation, opening doors, the use of tools, and the transfer of objects (such as tools) to and from a human [7, 16]. Another mobility option for Robonaut employs NASA's all-terrain, four-wheeled base called **Centaur** [13] (Figure 2.12(b)). At 450lbs, the statically stable Centaur is not quick or agile. Yet, Robonaut and Centaur excel at manipulation under teleoperation in tasks that can accommodate low-speed movements.

2.1.4 Whole-Body Humanoids

The Vecna Battlefield Extraction-Assist Robot, **BEAR** concept is characterized by a powerful upper torso and a reconfigurable tracked base that can operate in both statically stable and balancing postures (Figure 2.13). BEAR has demonstrated the ability to pick up and transport a human to safety [6]. This design is focused on the battlefield for extracting wounded soldiers, disarming bombs, and searching through rubble.

Figure 2.13. The Vecna BEAR

The **iCub** [4, 44] is a humanoid robot about the size of a two year old child designed to study motor development. It was designed by an international consortium and is intended to be expanded ultimately to 54 degrees-of-freedom.

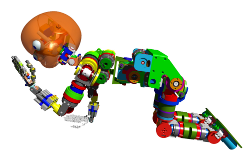


Figure 2.14. The iCub

Most robots today have only demonstrated limited functionality, and cannot generate large interaction forces (there are exceptions, i.e. Vecna BEAR). Many do not appear to be designed to survive a fall or right themselves. Many go to great lengths to avoid contact with the environment and thus are limited with respect to manipulation. In general, humanoid robots (due mainly to their size and expense) are often suspended in safety harnesses from gantrys to protect these expensive machines from damaging falls. Current research focuses on balance and locomotion and insufficient attention is given to whole-body strategies that support dexterity in mobility and manipulation.

Despite technological advances in packaging, miniaturization, computation, sensing, and actuation, integrated and multi-purpose robot systems that can safely work in human environments are still in the future. This dissertation seeks a new combination of dexterous hardware and hierarchical firmware that make it possible to achieve high-performance and programmability. A general purpose robot, like the

Rosie in the Jetsons cartoon, will be able to act autonomously, to manipulate objects, to traverse irregular terrain, and to communicate with human beings in natural ways.

2.2 Design Desiderata

In order to thrive in human environments, a robot must be robust to a routine levels of abuse (as are humans). This is a critical technical challenge; it is currently unreasonable to expect that a personal robot can survive long in the field. Thus off-the-shelf components should be used whenever possible to increase serviceability, leverage proven designs, ease replication, reduce complexity, and exploit existing economies of large scale production. Moreover, a low price point is essential to disseminate a platform and to develop a community of users. If commercial applications are to attract investment, demonstrable functionality, robustness, and performance must be combined at an effective cost.

2.2.1 Human Environments

Human environments offer advantages and disadvantages for robot designers. The advantages include a great deal of structure and ergonomic concern for the human morphology. Robots that live in human environments and participate in human tasks can be significantly simpler than robots designed to operate "in the wild." However, these environments still present a significant degree of variation.

To be able to perform human tasks, a robot must have a similar capacity to do work. Designs that consider size, workspace, mobility, battery life, strength and speed, sensing and computation will be necessary to accommodate the large degree of variation present in human environments. Presently, suitable benchmarks are absent in the field. One of the measures of performance of the whole-body mobile manipulator is the ability to directly interact with and collaborate with their human counterparts. In this section, we develop specifications relative to human performance.

Table 2.1. Ideal (human referenced) specifications of a whole-body mobile manipulator for use in human environments.

Household Robot Specifications

• Height, width, and footprint	Typical human proportions (Approximately 50cm wide, 1.8m tall)
• Reachable Workspace	Typical kitchen cabinet dimensions (Height of 1.8m and depth of 0.8m)
• Power Capacity	Greater than 8 hours of untethered runtime
• Mobility and Terrain	Typical office buildings, sidewalks, kitchen environment
• Payload Capacity	Weight of items typically found in grocery store (less than 2.5kg)
• Mobility Speed	Average human walking speed (2-3 kilometers per hour)
• Manipulation Speed	1-2 meters per second
• Grasped object scales	Size of grocery store items (from nuts & bolts to large beach balls)
• Sensing capability	Vestibular, proprioceptive, kinesthetic, tactile, audio-visual
• Network Bandwidth	Greater than gigabyte per second
• Processing	Teraflops, equivalent to 50-100 billion neurons

2.2.2 Geometrical and Kinematic Specifications

Human environments reflect the tall and slender morphology of the adult body. Additionally, robots must traverse ramps, thresholds, gaps, clutter, and provide space for other occupants to pass by. Suitable platforms will have a small footprint, navigate in spaces shared by humans, and have the capability to reach low and high places. For instance, in a human kitchen design, spaces can be as narrow as 36 inches, and accessible spaces, such as cabinets, may extend over 2 meters in height.

Human environments prominently feature graspable objects on the floor, on tables, and on high shelves, therefore a robot should be capable of having a similar range.

Since placement of the arm(s) on the platform is critical for defining the reachable workspace, they should be positioned near the top of the robot.

Furthermore, human environments are designed to make important visual cues visible from “Head Height” (3-7 ft.). Thus, some instrument packages (such as cameras and microphones) are best located up high to oversee the manipulator’s workspace, and to have the best perspective on the environment for navigation. A robot with a morphology similar to a humans can take advantage of as much of the structure already present in the environment.

2.2.3 Payload Specifications

Manipulation systems encounter objects that span a great degree in size, shape, and mass. Depending on the object or tool, fingers may be sufficient for small precise operations and power grasps can handle objects up to several centimeters. The mass of objects that human typically manipulate range over three orders of magnitude. To support common chores around the home, a robot should be able to grasp and lift common items such as those found in grocery stores. In addition, pay load capacity should support picking up and carrying significant loads. For transporting large payloads, bimanual grasps may be necessary. Table 2.2 lists object scales and masses that are characteristic of human manual behavior.

Grasp type	Object log_{10} Scale (m)	Object log_{10} Mass (kg)	Object log_{10} Mobility (m)
fingertip:	-2	-1	-2
power grasps:	-1	0	-1
bimanual:	0	1	0
mobile manipulators:	1	2	1

Table 2.2. Typical scales for grasped objects in human environments.

The range of specifications for grasped objects requires multiple scaled mechanisms. The progression from large to small objects and mass is associated with a proximal to distal mapping of actuation engagements. Thus, given the trade-off between power and weight, it is natural to place large actuators close to the body. In this location, limb inertia is minimized and the kinematic chain can branch towards multiple small end effectors as the lever arms supporting payloads are decreased.

2.2.4 Mobility Specifications

Table 2.2 highlights the complementary nature of manipulation and mobility in their capacity to move objects with varying size and mass. Fingertip grasps are generally applied to small objects that are controlled to fine precision (velocity and range of motion). However, human scale velocity (in Table 2.3) can easily exceed several meters per second when the entire body is utilized in tasks such as throwing. Thus, the appropriate delivery of velocity and range of motion requires the design of high performance dexterous machines.

Grasps	log_{10} Object Velocities (m/sec)
fingertip:	-2
power grasps:	-1
bimanual:	0
mobile manipulators:	0

Table 2.3. Typical log_{10} velocity required by robots for human environments.

Gross motor mobility (over large ranges of motion) are commonly implemented using statically stable mechanical structure. Statically stable designs get wider as they get taller. Mass should be concentrated near the ground and acceleration should be kept low enough to keep the center of gravity inside the supporting polygon of the base. For these reasons (and previous arguments for safety), the uBot-5 employs a dynamically stable chassis that can be used to support large payloads high above a small footprint on the ground and realize high longitudinal accelerations. Legged or wheeled systems can traverse the greatest variety of terrain while meeting tall and narrow form factor specifications. Legs require a significant amount of energy to operate, require more complex control, and often add significant weight, however, many terrains that are accessible to humans are also accessible to a wheeled robot (ADA guidelines are designed to make buildings available to wheeled vehicles). A wheeled robot allows controls to be simplified and overall mass can be significantly less than its legged counterpart. A balancing robot is also able to have larger wheels than statically stable platforms (with the same footprint), and thus it can roll easier over rough terrain and across cluttered environments. As a result of all these considerations, uBot-5 was designed as a dynamic balancer (inverted pendulum) to produce a small footprint, capable of growing to support large payloads (instrument packages and upper body) up high, and with wheels instead of legs.

2.2.5 Safety Specifications

Dynamic stability introduces potential performance advantages, but a robot that can fall over eventually will. This (infrequent) event may cost more than the added value of dynamics. Dynamically stable robots (and humans) are potentially unsafe and appropriate designs should incorporate both passive and active considerations for minimizing the probability of unsafe behavior. One means of minimizing predictable risk is to control the Cartesian impedance of the robot. The controlled inverted pendulum yields a low overall stiffness for a balancing robot.

In some cases, as with exploring unknown environments, humans rely on low impedance (compliance) in the arms and hands. When many human beings are moving about in a crowded room, they rely on low impedance to keep the forces from inadvertent collisions to a minimum acceptable level. This design is extended into an impedance controlled upper body. Precise manipulation tasks often require high impedance so that error is minimized. Primitive force control allows limits to be set on the magnitude of interaction forces between the robot and the environment. Controllable impedance characteristics (the ratio of force output to velocity input) enhances safety, performance, and robustness by matching impedance to different situations. Furthermore, minimizing the mass of the robot and stored potential energy also reduces the magnitude of impact forces in uncontrolled collisions.

2.3 Summary

The design of whole-body mobile manipulators for human environments should be driven by consideration for simplicity, serviceability, robustness, and cost. Thus, uBot-5 focuses on system integration. System performance is derived from synergistic relationships between a small number of components. We expect that integrated designs will perform better by combining mechanical toughness, with perceptual sensitivity, and behavioral skills that preserve long term viability. My goal is to put mechanism and control on equal footing and to increase performance in a more integrated hardware and software co-design. I claim that an integrated design methodology (mechanical, sensing, and computation) will produce a greater level of performance, ease programming complexity, and decrease the net cost of the machine.

CHAPTER 3

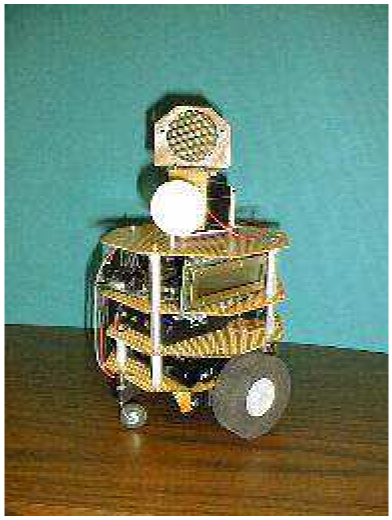
THE UBOT CONCEPT FOR WHOLE-BODY MOBILE MANIPULATION

3.1 The uBot Series

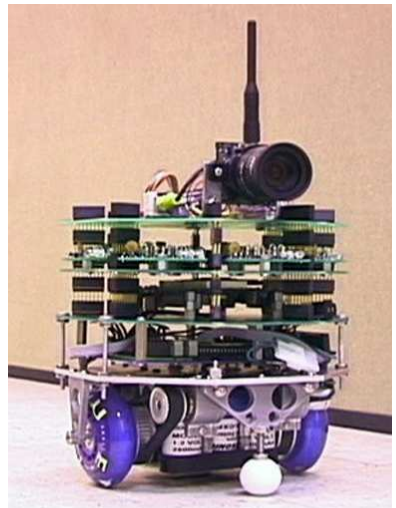
The design of the uBot-5 platform began with the goal of developing a way to combine mobility and manipulation in a seamless manner. The uBot-5 design is informed by several generations of earlier designs. The uBot platform has evolved from a reconfigurable micro-robot for studying multi-agent behavior to a balancing robot featuring trunk rotations, two 4 degree-of-freedom arms, and whole-body postural control.

3.1.1 uBot-0

The first uBot was a simple statically stable mobile base and did not support manipulation. The uBot-0 (Figure 3.1(a)) was constructed in 1997 at UMass from inexpensive and widely available materials and hardware. Hobby airplane wheels were attached to a base made from perforated PCB board in a differential drive configuration. It was a prototype element of an approach to a distributed, multi-robot experimental platform concerned with multi-robot collaboration and communication. Hobby airplane casters were placed at the front and back for static stability. A Motorola HC6811 processor monitored the sensors, controlled wheel motors, and computed odometry. The platform used a reactive control framework, inspired by Braitenberg [9], that provided inputs to the differential drive. uBot-0 executed controllers that were attracted to thermal stimuli and repelled from obstacles detected by sonar.



(a) uBot-0.



(b) uBot-1 navigating a maze-like environment.

Figure 3.1. The first two uBots in the series

3.1.2 uBot-1

uBot-1 was a one generation improvement of uBot-0 used in the same type of research question (Figure 3.1(b)). The uBot-1 borrowed design ideas from the uBot-0 including a small form factor and limited on-board computational power. It improved upon the design by incorporating high performance motors, reconfigurable sensors (infrared proximity), and wireless communication (infrared or radio). This configuration was used to demonstrate a control theoretic framework for multi-robot collaboration. The design supported exploration within a set of control states and actions that satisfy global specifications on the behavior of the team. Sweeney et al. [54] demonstrated this framework in a maze exploration task. In Sweeney's maze experiment, pairs of robots explore a maze while maintaining line-of-sight to protect global network connectivity. Figure 3.2 shows two uBot-1 robots searching a maze using a leader-follower control composition while simultaneously preserving visual line of sight. The control scheme is used to assemble multi-objective controllers to manage redundant degrees of freedom and applies (theoretically) to heterogeneous team of robots.

Figure 3.2. Two uBot-1 robots performing a maze exploration while maintaining the line-of-sight constraint.

One of the disadvantages of the uBot-1 design was that the drive system performed poorly on surfaces that were not smooth and flat. This was because its wheels provided limited surface contact and thus resulted in poor traction. To compound the problem, the spherical Teflon® casters that provided static stabilization, had small contact areas with the floor, thus limiting the effective velocity of the platform and catching on small surface imperfections and debris.

3.1.3 uBot-2

Due to symmetry in the base drive geometry, the uBot-2 (Figure 3.3) was adaptable for balancing behavior. The symmetry was originally chosen so the platform could have a nearly circular shape and turn in place as well as traverse horizontal ground planes. The successive iterations of the platform have retained this mobility feature.

uBot-2 explored closed-loop balancing control to address the principal limitations of uBot-1. As a result, the platform was able to traverse a wide variety of terrain. Reconfigured this way, uBot-2 increased speed, performance, and maneuverability but required active stability control. A classical Linear Quadratic Regulator (LQR), whose inputs derived from a rate gyro and accelerometer, was employed to stabilize

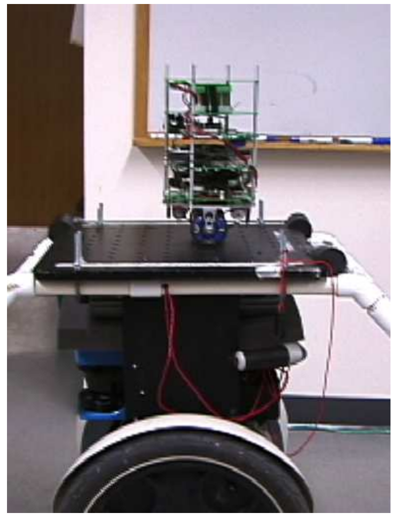


Figure 3.3. The uBot-2 balancing on top of a balancing Segway RMP.

the inverted pendulum. The LQR optimizes the behavior of the inverted pendulum in the neighborhood of the vertical posture. However, the platform could fall from vertical to prone when the state of the platform departs significantly from the original linearized model. Balancing on two wheels creates a robot that is naturally compliant since the longitudinal impedance of the non-holonomic base is dominated by the low input impedance of the inverted pendulum. The natural compliance enhances safety when the platform operates around people since the platform tends to comply easily to environmental forces.

uBot-2 showed a respectable mean time between failures despite many crash landings caused by inexperienced drivers or environmental perturbations. The ruggedness is due in part to the small scale of the platform that results in relatively small impact forces. This robustness was considered extremely attractive because the platform could be used to explore the bounds of the stability controller, and to attempt tasks that could potentially result in a loss of stability. For these reasons, a commitment to small scale was adopted throughout the subsequent uBot line.

The balancing behavior of the inverted pendulum can also mimic human bipedalism and suggests that a humanoid robot with two arms and two wheels would be suitable for research in mobile manipulation for human environments. The University of Massachusetts Amherst **Dexter** (Figure 3.4) consists of two Barrett WAM arms, each with a 3-finger Barrett hand. The fingertips are instrumented with 6-axis force/torque sensors. Dexter has shown dexterous manipulation of unknown objects and tool use with and without teleoperator control [15].

Dexter is normally stationary and relies on a rigid base for stability and precision. However, by placing the dynamically balancing uBot-2 platform within Dexter's reachable workspace, manipulation tasks were performed that were subject to interaction forces. These interactions mimicked those that a dynamically stable WBMM would experience. The system achieved stable grasps and relied on force feedback from a master arm that translated balancing uBot-2 self-motion into displacements on the slave arm while it simultaneously turned a light bulb [50]. The results suggest that high fidelity force feedback and active control is necessary to deal with interaction dynamics. While it is also necessary to compensate for self-motion, a typical characteristic of balancing platforms.

3.1.4 uBot-3

The uBot-3 (Figure 3.5) started as a feasibility study for a dynamic balancer that was equipped with two 3 degree-of-freedom arms actuated by hobby servos. The uBot-3 was built by a CS503 class (Embedded Systems) and was used primarily to conduct simple pushing experiments. Quantitative analysis of the advantages of postural control are discussed at greater length in Section 5.1. The initial study also concluded that with only 3-DOF per arm, this version of the uBot had a limited bimanual workspace. Because of the low power motors, uBot-3 was also not able to right itself after a fall. Despite these shortcomings, uBot-3 established the the first

Figure 3.4. The uBot-2 presents dynamically stable environment to the Barrett WAM stationary manipulator in the task of screwing in a light bulb.

technical results at UMass regarding the interaction of a balancing platform and a "live load" upper body. This problem space was viewed as an important opportunity that has not yet received a great deal of attention. Thus, a commitment to dynamically stable bimanual manipulators was reflected in subsequent iterations.

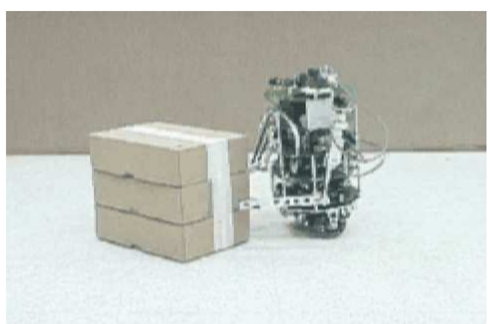


Figure 3.5. Balancing uBot-3 pushing a box.

3.1.5 uBot-4

The uBot-4 mobile manipulator concept began with models represented in engineering design CAD software beginning in mid 2005 (Figure 3.6). Relative to the uBot-3, the uBot-4 has a higher center of mass, larger wheels, a higher top speed, more powerful motors, greater on-board computation, larger workspace, more degrees-of-freedom, and greater battery capacity.

The uBot-4 has two four degree-of-freedom arms and a rotating shoulder girdle. This design allows the platform to handle objects that are not directly in front, without having to change the heading of the base. The platform is relatively stiff laterally and compliant longitudinally. The trunk rotation allows the robot to apply this anisotropic passive stiffness to tasks appropriately.

The geometry of the arms and the upper body was designed to create an effective bimanual workspace while grasping a 12 inch sphere. Figure 3.7 demonstrates various bimanual grasp configurations for this prototype object. The length of the arms was chosen so that the bimanual workspace also includes a large region on the ground plane.

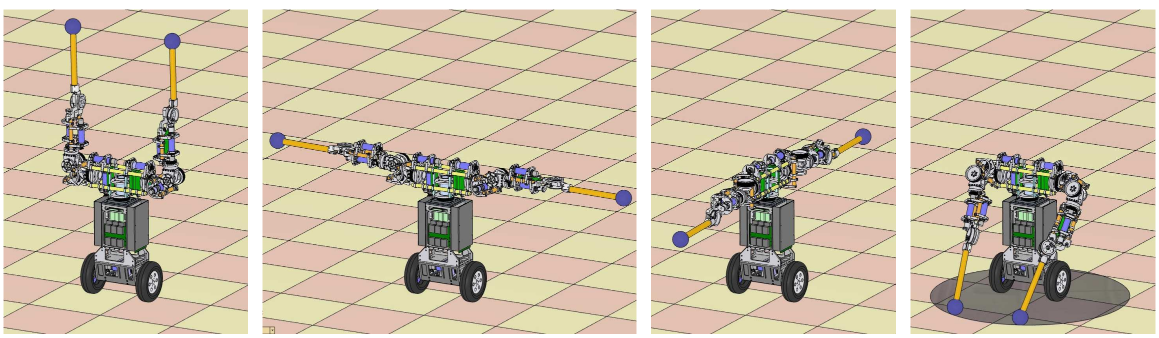
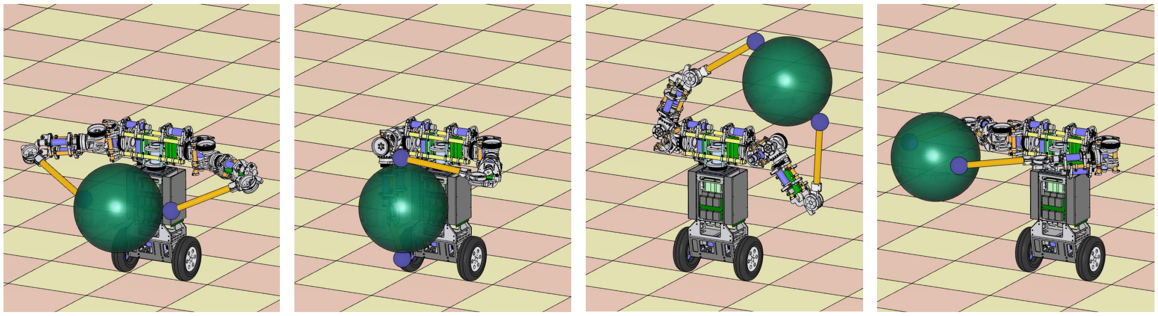


Figure 3.6. Poses reflecting some of the uBot-4 workspace. The circle represents reachable workspace on ground plane (body is vertical).

CAD models of the robot were used to estimate the weight and center of mass of the robot. Prior to manufacturing, body mass distribution and strength-to-weight ratios were estimated for the aluminum frame using motor models and pulley and gear reduction ratios in the representative postures in Figures 3.6–3.8. These parameters were integral to balance the strength and speed requirements for the limb, and to guide the selection of motor/gearbox combinations. uBot-4 used the same motors as in previous uBot generations. As a result, it was not possible to achieve



(a) (b) (c) (d)

Figure 3.7. Various grasps of a 1 foot diameter ball. Images (c) and (d) demonstrate bimanual joint range limits for this object.

an optimal reduction ratio. Design tasks were selected to specify velocities and force requirements in the actuator and drivetrain design including throwing, bracing for a fall, dissipating impact forces with the arms, and performing push-ups. Some of these tasks were right on the performance boundary of the design and required that motor and drivetrain design be reconsidered in uBot-5.

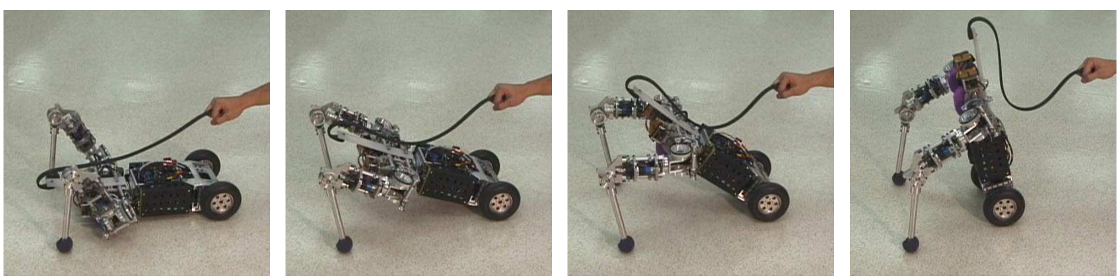


Figure 3.8. uBot-4 demonstrating push-up from prone position.

In Figure 3.8, uBot-4 demonstrates a push-up motion represented as a cubic spline through predetermined via points. As a proof of concept, the push-up motion exposed hardware inadequacies in the form of drivetrain belt slipping and end-effectors slipping on the ground. In certain tasks, the under-actuated elbow of the uBot-4 proved marginally inadequate. Often, when doing push-ups the elbow would collapse during the motion and success was very sensitive to battery charge.

The uBot-4 prototype involved a significant integration effort. This work included custom machining, printed circuit board (PCB) fabrication, and the development of a

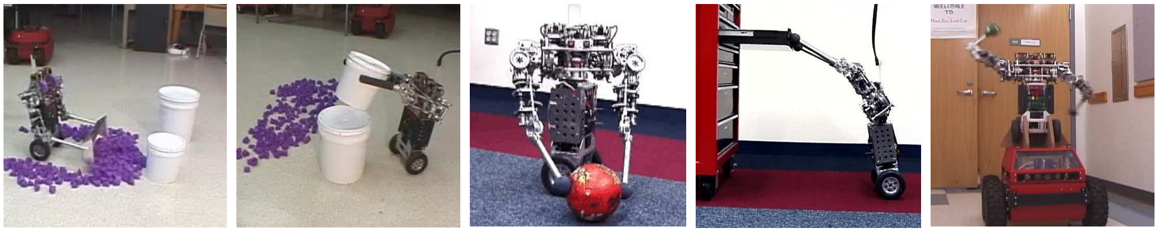


Figure 3.9. uBot-4 demonstrating range of abilities (from left): shoveling, bucket stacking, drawer pushing, and throwing.

computing platform including an embedded motor control architecture. Printed circuit boards for power distribution were developed using PCB design software. CAD software was utilized to generate drawings for custom mechanical parts. The computing platform and peripherals for motor current amplification and low-level motor control were constructed from commercial off-the-shelf components. The embedded compute system required custom software to interface the peripherals and provide network connectivity to off-board processing.

I led a team including: Bryan Thibodeau, Mike O'Malley, Ed Hannigan, and Dirk Ruiken to demonstrate a variety of fully integrated tasks with the uBot-4. These tasks included pushing a drawer closed at a height greater than its center of mass, grasping, lifting and transporting of a small rubber ball, and throwing a racquet ball (see Figure 3.9). At around $15,000, the uBot-4 demonstrated that the concept could successfully merge performance with cost effectiveness.

3.2 The uBot-5

The uBot-5 is a dynamically stable, differential drive two-wheeled robot with a rotating trunk and two four degree-of-freedom arms (Figure 3.10). The robot incorporates advanced on-board computing, high density power supplies, and high performance motor controllers with intrinsic force measurement. Approximately the size and weight of a four year old child, the uBot-5 is designed to be playful, approachable, and safe around humans.

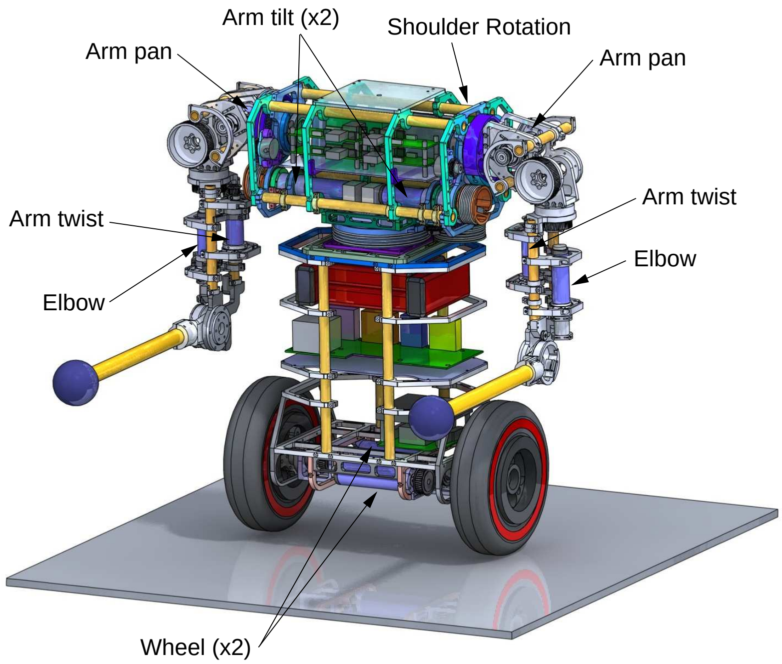


Figure 3.10. A solid model rendering of the uBot-5 design showing the location of the actuator for each degree of freedom.

The innovative platform integrates speed and agility with the capacity to interact forcefully with the environment in various bimanual manipulation tasks. Dynamic balance enables the platform to support relatively large payloads high above the ground with a small footprint. The marriage of mobility and manipulation affords the ability to create momentum in powerful, massive structures and to transform it into kinetic energy in smaller limbs and grasped objects to address throwing, weightlifting, and hitting tasks. Figure 3.11 shows the uBot-5 throwing a regulation baseball nearly 20 feet while balancing. The platform can impart forces to environmental surfaces with magnitudes on the order of its body weight (so it can perform a push-up) and during collisions with the environment can significantly dissipate the impact (e.g. a safe anticipatory brace for a fall). These characteristics represent a designed balance between the capacity for velocity and strength.

uBot-5 is small, lightweight, and mechanically robust. It uses an open chassis design to protect mechanical and electrical components from direct impact forces.

Figure 3.11. The uBot-5 exploits upper body dynamics to throw a baseball while maintaining postural stability.

The small stature of the robot contributes to toughness by limiting the potential energy in the system. Within the chassis, care was taken to place actuator mass as close to the trunk as possible so that the inertia of each arm is minimized.

The performance of the uBot platform is increased by several improvements over its predecessors, including motor upgrades, a high performance FPGA motor control system, and a redesign of the chassis to increase the internal volume for electronics. Table 3.1 lists some general parameters of the uBot-5 design.

Table 3.1. uBot-5 physical parameters

Property	**Value**
Height	0.70m
Width	0.58m
Depth	0.20m
Footprint Width	0.40m
Footprint Depth	0.20m
Weight	17kg
Battery Capacity	13.8Ah at 13.2V
Lifting Capacity of two arms at Full Extension	49 N
Top speed	12km/h

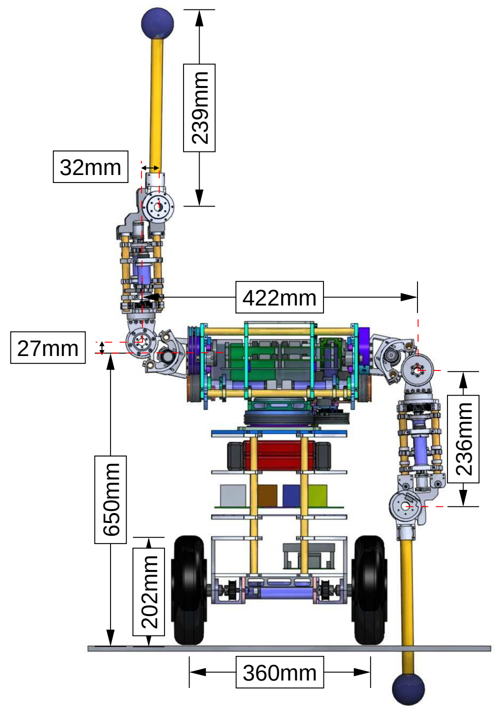
3.2.1 Platform Kinematics

A dexterous robot is able to manipulate objects in the environment in many different contexts. Dexterity itself conjures ideas of agility, swiftness, and adroitness in problem solving, especially relating to hands. A great deal of thought has been given to the dexterity of the uBot, even at the expense of other design goals. For example, a unique function of a mobile manipulator concerns the ability to reach to and touch environmental surfaces in order to participate in mobility functions and use contacts to stabilize the platform, including bracing against the ground in the event of a fall. Figure 3.12(b) demonstrates grasping a ball on the floor. For a robot with wheels instead of legs, this functionality leads to a rather short ape-like stature (Figure 3.12(a)). While this design may be beneficial in bracing for a fall, it hinders the robot's ability to reach elevated spaces (e.g. the top shelf of a kitchen cabinet).

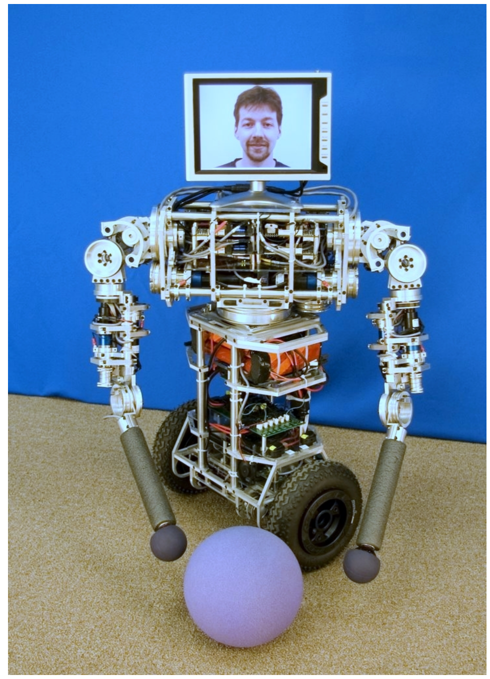
joint	i	α_{i-1}	a_{i-1}	θ_i	d_i
base rotate	1	0.0	0.0	0.0	0.101
wheel tilt	2	$\pi/2$	0.0	0.0	0.0
torso rotate	3	$\pi/2$	0.006	0.0	0.549
arm tilt	4	$\pi/2$	0.002	0.0	0.211
arm pan	5	$\pi/2$	0.027	0.0	0.0
arm twist	6	$\pi/2$	0.0	0.0	0.236
elbow	7	$\pi/2$	0.032	0.0	0.0
end effector	8	$\pi/2$	0.0	0.0	0.239

Table 3.2. D-H Parameters for left arm.

The body and mobility design enables the robot to manage tight corridors and shared spaces with humans such as kitchens and hallways. The ability to reach the floor in an upright posture enables the robot to return to a vertical posture from the prone position (push up) and to brace against a fall. Small mass and stature are important developmentally to reduce the potential energy and reduce the risk associated with learning to exploit dexterous mobility options. Once the risk has been mitigated, the chassis can be scaled to match the reach of an adult human.



(a) CAD model of uBot-5 design illustrating chassis dimensions and link lengths.



(b) The uBot-5 demonstrates a large, bi-manual workspace while reaching to a ball on the ground

Figure 3.12. uBot-5 kinematic design and workspace

The Table 3.2 lists the Denavit and Hartenberg (DH) parameters that reflect the rigid bodies and the revolute joints that connect them for the present uBot-5 design.

3.2.2 Differential Drive Configuration

Mobility in the uBot series is provided by two wheels in a differential drive configuration. The balancing differential drive is designed to go most places where humans typically go, and to move into and out of handicap accessible buildings (automatic doors, elevators, ramps, etc.) at roughly human walking speeds. Since the platform can rotate in place, balancing control is only considered about the wheel axis. In situations where terrain is more challenging, the upper body can deliver forces on environmental surfaces to stabilize and propel the robot.

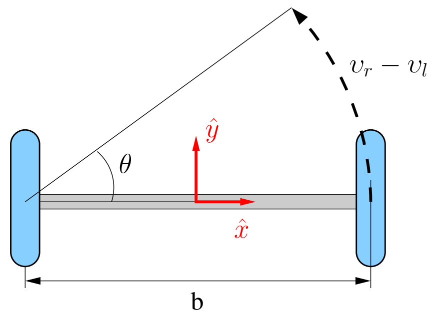


Figure 3.13. Geometry of the differential drive mechanism. Rotation is the result of a different between wheel velocities.

Figure 3.13 shows the change in orientation, θ, of the platform when the left wheel velocity, υ_l, and the right wheel velocity, υ_r, are not the same. The change in orientation from the point of view of the left wheel is described by the differential equation:

$$\frac{\mathrm{d}\,\theta}{\mathrm{d}\,t} = \frac{(\upsilon_r - \upsilon_l)}{b} \tag{3.1}$$

Integrating Equation 3.1 with the initial orientation of the robot, $\theta(0) = \theta_0$, the robot's orientation as a function of wheel velocity and time:

$$\theta(t) = \frac{(\upsilon_r - \upsilon_l)t}{b} + \theta_0 \tag{3.2}$$

The robot's overall motion depends on the velocity at the midpoint of the axle, the average of for the two wheels $(\upsilon_r + \upsilon_l)/2$. The differential equations for the x and y positions of the midpoint are:

$$\begin{aligned} \frac{\mathrm{d}\,x}{\mathrm{d}\,t} &= \frac{(\upsilon_r + \upsilon_l)}{2}\cos(\theta(t)) \\ \frac{\mathrm{d}\,y}{\mathrm{d}\,t} &= \frac{(\upsilon_r + \upsilon_l)}{2}\sin(\theta(t)) \end{aligned} \tag{3.3}$$

Integrating, the equations for the x and y positions of the midpoint are:

$$
\begin{aligned}
x(t) &= x_0 + \frac{b(\upsilon_r + \upsilon_l)}{2(\upsilon_r - \upsilon_l)} \left[\sin\left(\frac{(\upsilon_r - \upsilon_l)t}{b} + \theta_0 \right) - \sin(\theta_0) \right] \\
y(t) &= y_0 + \frac{b(\upsilon_r + \upsilon_l)}{2(\upsilon_r - \upsilon_l)} \left[\cos\left(\frac{(\upsilon_r - \upsilon_l)t}{b} + \theta_0 \right) - \cos(\theta_0) \right]
\end{aligned}
\tag{3.4}
$$

3.2.3 Actuator and Drivetrain Design

The trade-off analysis between high torque and backdrivability for uBot-5 involved several guiding cases. On one hand, push-up tasks require large torques. However large output forces require a significant gear reduction and/or large rotor mass and both parameters tend to decrease passive backdrivability. On the other hand, throwing and bracing motions require high speed actuators. However, in consideration for uBot-5's limited on-board power capacity, choosing a large gear reduction helps to conserve power. This is particularly important as motors are frequently operated near zero speed or stall conditions during manipulation tasks. Thus, the final configuration of the uBot-5 errs toward strength and efficiency over high speed, backdrivable actuators. Table 3.3 summaries the actuator and drivetrain choices for uBot-5.

Joint	**Motor**	**Power (W)**	**Reduction**	**Pulley Reduction**	**Type**
Wheel	3257	84	14:1	1:1	Belt
Shoulder	2642	23	43:1	1:1	Cable
Arm Tilt	2342	20	43:1	2:1	Cable
Arm Pan	2342	20	43:1	3:1	Belt
Arm Twist	2342	20	43:1	3:1	Belt
Elbow	2342	20	43:1	3:1	Cable

Table 3.3. Micromo coreless 12V DC motor and drivetrain specifications for uBot-5.

3.2.4 Force Feedback Options

Force based control is considered important to uBot research. Many technologies have been explored including current feedback, series elastic actuators (SEA), and intrinsic load cells.

The uBot-5 is capable of measuring current in all upper body motors. A low pass filter is used to smooth the signal prior to analog to digital sampling. Unfortunately the motor current sensor is also affected by a $\pm 9\%$ non-linearity over its range due temperature related changes in resistance of the motor coil, current amplifiers, and lead wires. Moreover, the measurement is on the wrong side of the gearbox. Due to this design choice, current sensing is affected by a significant loss in accuracy over more direct sensing methods. Therefore, motor current sensing provides only very low resolution force feedback.

The Series Elastic Actuator (SEA) is an example of an actuation technology that balances passive compliance and performance [48] using with substantial gear reduction ratios. This type of actuator has also been used to implement active impedance control [47]. Such an actuator is constructed by placing an elastic element in series with a motor, either between the motor and the load (SEA) or between the motor and the motor mount (Force Sensing Compliant Actuator, FSCA). Force control is enabled by sensors that measure the spring displacement and relate displacement to motor torque. The elastic element also provides compliance that helps to protect the drivetrain. However, the main disadvantage in terms of performance is that with increased compliance, the bandwidth and precision of the manipulator can be diminished.

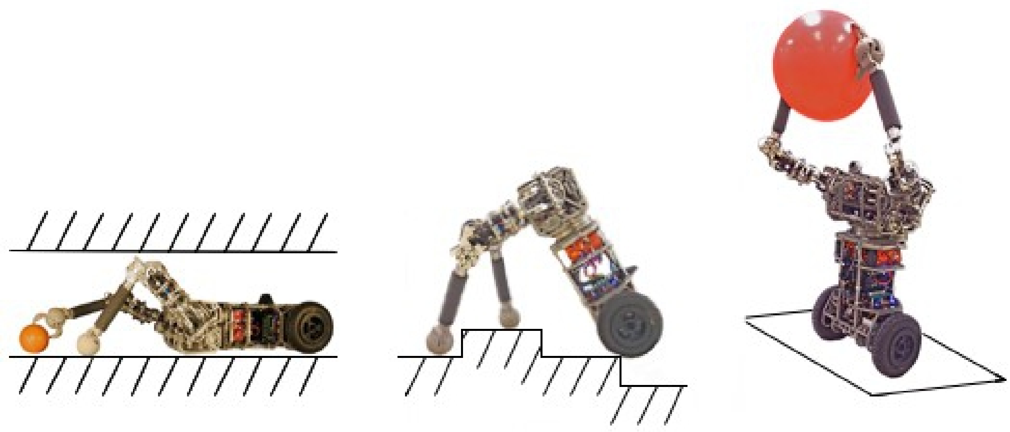
The uBot-5 employs a motor mount design that supports plans for future use of series elastic elements or intrinsic load cells on the load side of the gearbox. The intermediate implementation employs a stiff aluminum beam as the elastic element. To implement force sensing, a strain gauge can mounted to the beam. The gauge changes resistance in proportion to strain in the beam and signal processing hardware can provide a calibrated output torque measurement. At this time, a prototype strain gauge has been implemented and tested. Additional empirical work is required to determine if the sensor provides suitable performance (e.g. meeting noise, accuracy, and precision specifications) for use as calibrated torque feedback variables in a force control system.

Due to implementation difficulties with the complex current sensing and strain gauge FPGA design software, force feedback was put off until subsequent revisions of the platform. Thus, part of the ongoing development of the platform will be to assess the performance of the force control and actuator design.

3.3 Hybrid Mobility Modes

uBot-5 explores a design point that couples mobility and manipulation at the expense of introducing challenging control and learning issues. Nikolai Bernstein observed that biomechanical systems exploit dexterity by creating seamless solutions to the "degree of freedom" problem [37]. Bernstein proposed that in motor learning, degrees of freedom are initially frozen to decrease motor learning complexity and unfrozen incrementally to re-introduce degrees of freedom. This strategy, according to Bernstein, bootstraps the motor learning process and creates coordinated motions that exploit dynamics. Often, redundant degrees of freedom are exploited to address secondary objectives while executing a reference endpoint trajectory [62]. The uBot takes the approach that mobility and manipulation are two aspects of a multi-objective control system that can be applied to whole-body applications. The robot design, mechanically and computationally, is organized to support learning mechanisms for acquiring dexterous strategies for mobility and manipulation.

uBot-5 is capable of a variety of discrete postural stability modes, with differing resource engagements. The simplest (and most stable) mode of the uBot-5 is the prone posture of the robot, where it lies (face down) on the ground plane. The free hands may be used for some other task, or repositioned in a sequential gait to "scoot" the platform across the floor. Figure 3.14(a) demonstrates the robot reaching for a ball from the prone stance. Prone mobility is useful in vertically restricted environments such as under tables.



(a) Prone (b) 4-point Stance (c) 2-point Stance

Figure 3.14. Various hybrid mobility modes

In a sequence of control actions that place its arms in a push-up configuration and then execute the push-up, the robot is capable of transitioning from the prone posture to the "4-point" posture. From there, one hand may be withdrawn to yield the "3-point" stance. Three contact points is the minimum necessary to achieve a statically stable posture. These postures are considered mobility modes if the robot is able to translate or rotate without falling over. This configuration serves to support knuckle walking gaits over terrain that is too challenging for a 2-point cart-pole balancer. The additional degrees of freedom in the upper body increase the number of mobility modes by employing permutations of 3-point contact geometries. The maximum number of contact points we will consider is four, the two end-effectors and the two wheels (Figure 3.14(b)).

The 2-point stance employs the dynamically stable, cart-pole balancer to leave arms and hands free (Figure 3.14(c)). In this mode, the differential drive equations described earlier are used to provide gross motor control in large scale spaces.

3.4 Summary

The uBot-5 is a whole body mobile manipulator designed to interact in unstructured environments, characterized by great variability. The uBot-5 can use its entire structure for mobility and manipulation, enabling it able to reach out and grasp objects that vary in scale and position. The platform's 11 degrees-of-freedom support a large workspace and enable versatile mobility. The design also features a rugged, untethered frame so that programmers and learning algorithms can rely on exploration, and the freedom to "make mistakes" without severely damaging the robot. In this chapter, I have provided a complete review of decisions and the resulting mechanical design for uBot-5. In the next chapter, I will present the complementary hierarchical compute architecture that manages the flexibility of the robot.

CHAPTER 4

HIERARCHICAL COMPUTE AND CONTROL ARCHITECTURE

The mechanical flexibility and dexterity of the uBot-5 is equally complemented by an embedded compute architecture designed to control multiple degrees-of-freedom and produce smooth, reactive, coordinated arm and body motion. At the lowest level, FPGA fabric supports reactive motor units characterized by minimal computational requirements, high update rates, and low latencies between sensors and actuators. More complex, higher level sequential and coordinated motions are possible within the FPGA core and at the PC/104 level as they require lower update rates and latencies, but have greater computational utilization. This architecture implements control hierarchically in three levels (Figure 4.1).

A Field Programmable Gate Array (FPGA) provides the lowest level of motor control at 2kHz. The FPGA fabric processes signals directly from quadrature encoders (motor position and velocity) and updates the motor torque via a pulse-width modulated (PWM) signal using PID controllers. This embedded computation implements controllers analogous to human spinal motor units. The FPGA includes an integrated PowerPC code where the balancing controller is implemented. Other reflexes are implemented on the PowerPC core or on the PC/104+. Policy-level algorithms requiring more computation and less bandwidth and sensor processing are implemented on the PC/104 or off-board using the wireless network. Figure 4.2 shows the location of several of the embedded control sub-systems.

The uBot-5 also utilizes a single board compute architecture (PC/104+ standard) running a variant of the Linux OS or Windows XP to provide wireless 802.11 network

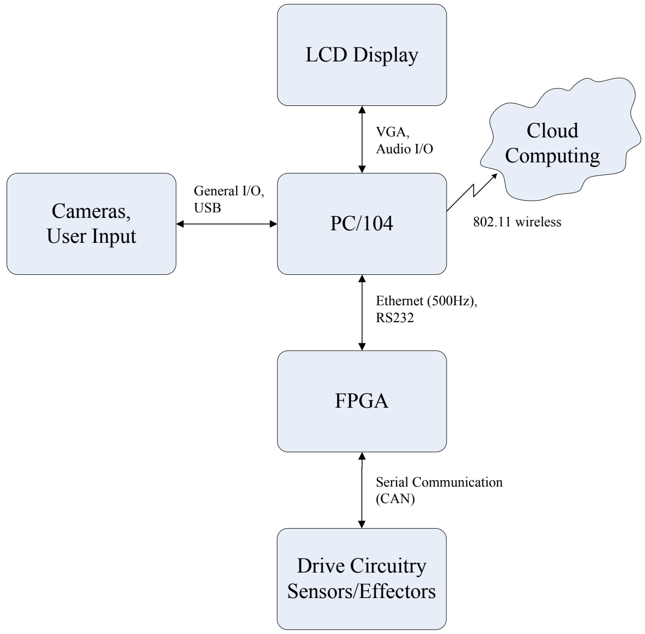


Figure 4.1. Diagram showing the design of the hierarchical compute architecture.

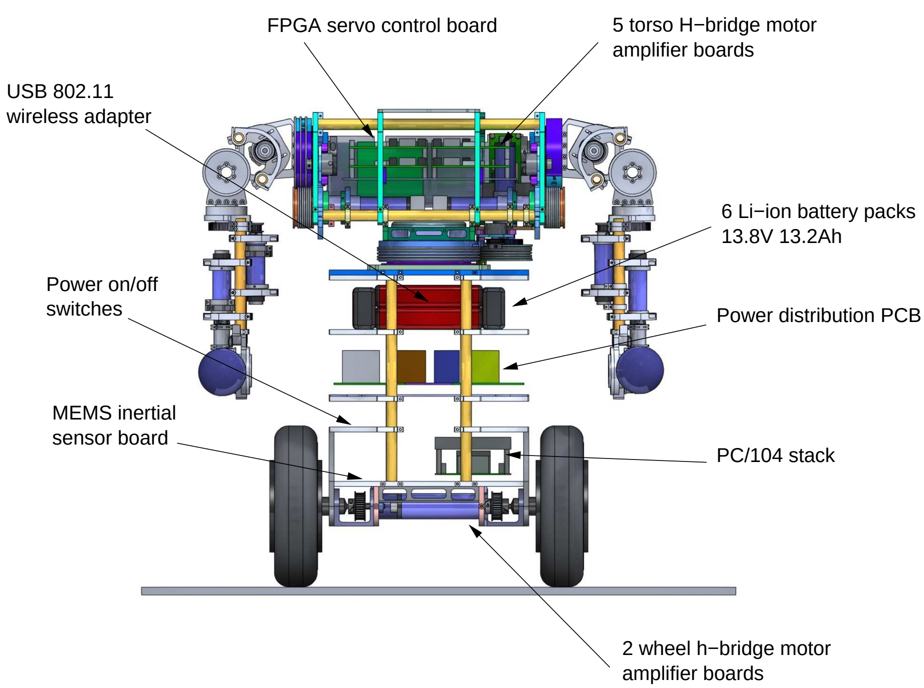


Figure 4.2. A solid model rendering of the uBot-5 design showing the location of embedded control sub-systems.

connectivity and to implement a policy and action level interface for application

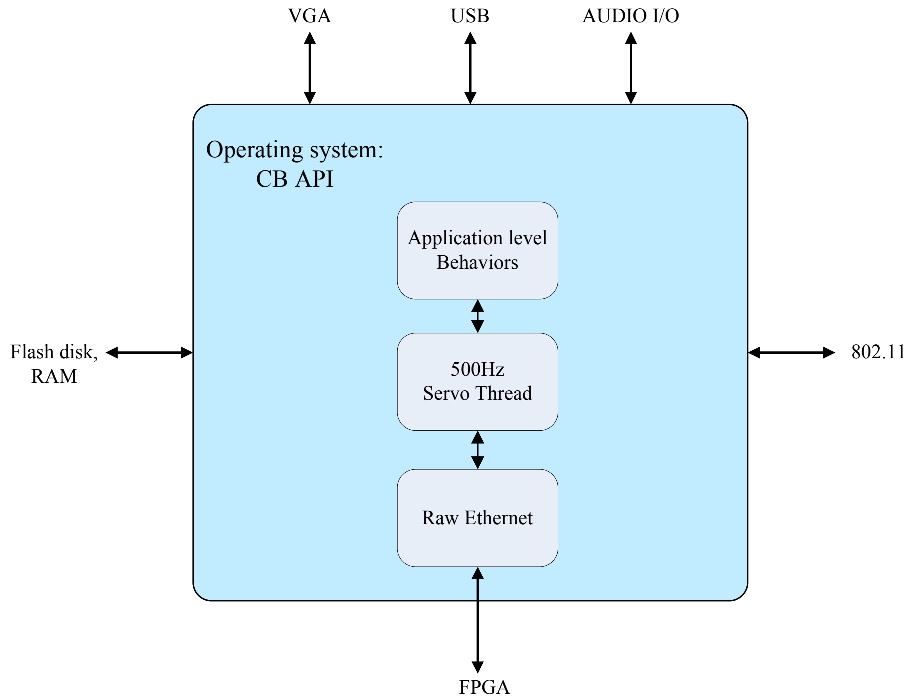


Figure 4.3. PC/104 System Design.

developers (Figure 4.3). A wireless connection simplifies the on-board hardware and enables the uBot to participate as a client of sensor networks.

4.1 Embedded Controllers in the FPGA

The Controller Area Network (CAN) bus is one of the most commonly used protocols for interprocess communication in embedded control applications. It delivers low-latency, high bandwidth (1Mbps), and reliable communication between distributed processes. However, the throughput of CAN was considered insufficient. Therefore, in order to realize coordinated motion in all 11 degrees-of-freedom of uBot-5 a co-located motion control approach was selected.

Custom motor control hardware was developed based on a Xilinx Virtex®-4 Field Programmable Gate Array (FPGA). The FPGA provides an independent 2kHz update rate for position, velocity, and direct PWM duty cycle for 12 channels of motor control (one more than necessary in the basic uBot-5 implementation).

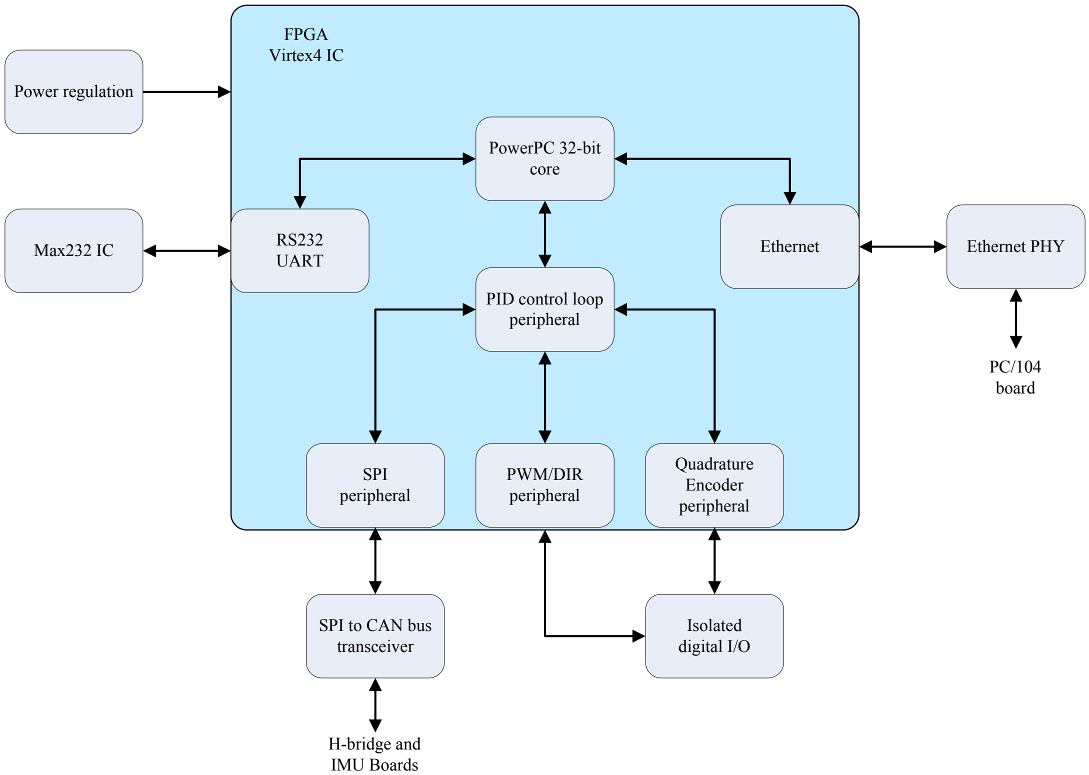


Figure 4.4. Diagram showing the design of the FPGA subsystem control hardware.

Figure 4.4 shows the functional peripherals implemented in the FPGA fabric. These include separate free running peripherals for PWM signal generation (12-bit, 20kHz motor power modulation), quadrature encoder capture (32-bit position and velocity measurement), and several Serial Peripheral Interconnect (SPI) communication bus channels. An on-die PowerPC core, utilizing interrupts to maintain real-time guarantees, bridges between the FPGA, the PC/104, and off-board computing by way of a serial or Ethernet bus. Figure 4.5 illustrates the embedded system architecture and FPGA feedback.

The drawback of utilizing a co-located motor controller is the added complexity of cabling. Each actuator typically requires many individual signals including wires

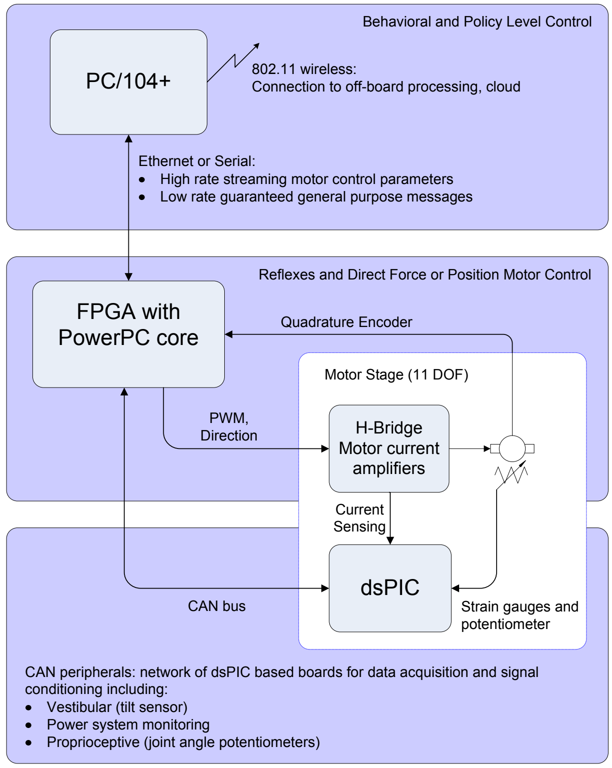


Figure 4.5. Diagram showing information flow between low-level motor control hardware.

for fault or status monitoring, feedback, and drive circuitry. An approach where the control processors are distributed spatially is often able to consolidate wires and concentrate data over a serial communications bus. Figure 4.6 shows the motor amplifier PCB design with separate connections from the FPGA to realize closed-loop control with intrinsic force feedback. A dsPIC microprocessor was utilized to partially mitigate the cabling issue by implementing a local sensor interface whereby filtered data could be sent back to the FPGA over the CAN bus. This was a practical solution because some sensor feedback (e.g. strain gauge measurement) had much lower bandwidth and timing constraints (vs. quadrature encoder signals for instance). This also did not require a dedicated peripheral in the FPGA fabric to meet performance requirements.

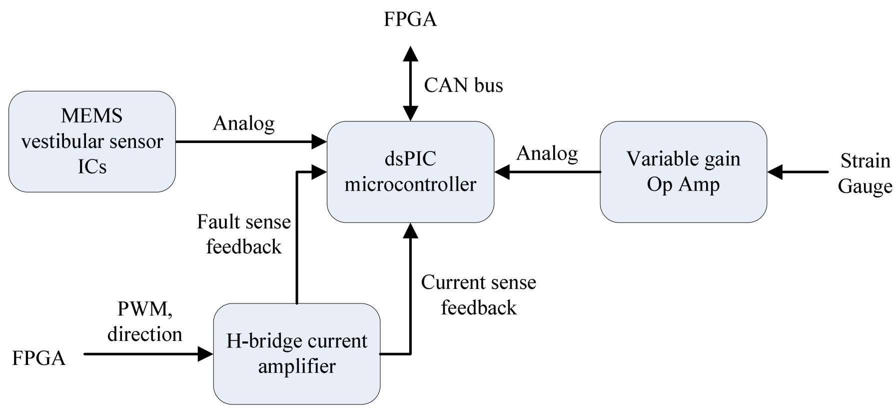


Figure 4.6. Diagram showing the design of the H-bridge subsystem control hardware.

The CAN bus provides expandable sensor I/O that refreshes data directly into the PowerPC memory. In this manner, force measurement and signal conditioning can be distributed on the microcontrollers located near each motor to reduce signal noise. In order to implement balancing behavior, filtered values for body tilt and rate of tilt are available on the CAN bus from an inertial measurement unit (IMU) that utilizes a miniaturized accelerometer and rate gyro (Figure 4.7). The accelerometer relates tilt of the body to measured gravity depending on the orientation of the sensor. The tilt and rate of tilt are filtered to eliminate steady state inaccuracies and reduce sensitivity to non-gravity related accelerations.

The uBot-5 features reflexes that must operate with a very low latency response. High level processes are not suitable for the implementation of reflexes because they suffer from large and variable communication latencies between the high level hardware and the embedded controllers. High level reflex processes would also be subject to preemption, jitter, and latency by the operating system. Therefore, the embedded FPGA hardware is specially designed to provide a low latency control loop between sensors and effectors. However, resources such as memory and computational cycles

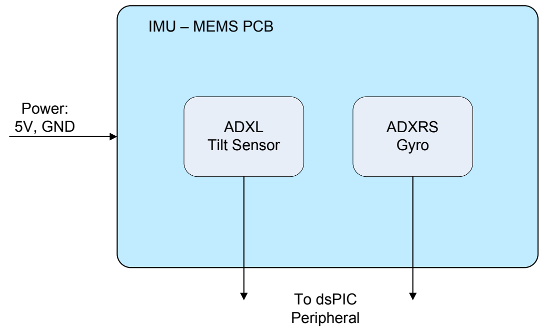


Figure 4.7. The uBot-5 inertial measurement system design.

are very limited and the embedded control code is designed to meet trade-offs between algorithmic complexity and implementation resources. Therefore, high-level applications code resides on the PC/104 stack where CPU and memory resources are less constrained and timing requirements are easily met with commercial operating systems.

4.1.1 Motor Units

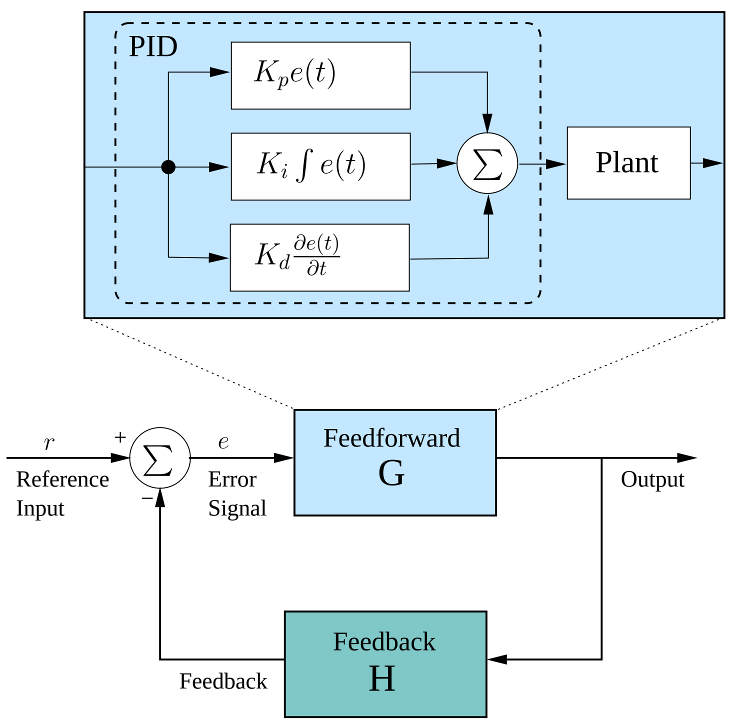


Figure 4.8. Generalized feedback control structure.

The independent degrees of freedom of the robot are actuated using embedded equilibrium setpoint controllers (or motor units) that apply torques to the mechanisms comprising uBot. These motor units use control laws with negative feedback (Figures 4.8 and 4.9) to stabilize the output load, $\frac{1}{M}$, subject to environmental perturbations, f_d while minimizing the error between the observed state feedback, $\mathbf{x}_{act}$, and real-valued position references, $\mathbf{x}_{ref}$. Figure 4.9 shows the separation between computation (digital) and the physical world (analog), where the control parameters are provided by higher-level controllers. At the lowest level, the digital computed control utilizes a Proportional-Integral-Derivative (PID) control law. The PID forms a primitive basis for subsequent hierarchical closed-loop control.

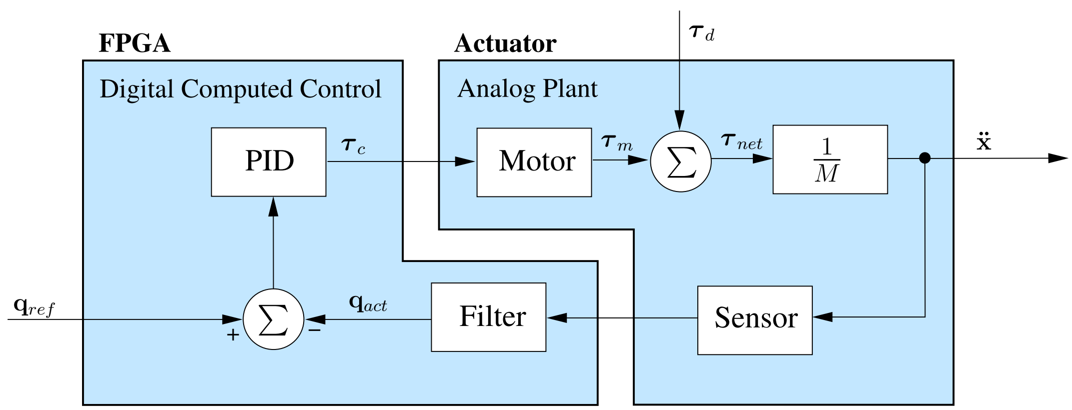


Figure 4.9. Hybrid digital/analog feedback control structure.

4.1.2 Impedance Control

Impedance is defined as the ratio of force to velocity.

$$Z = \frac{F}{V} \tag{4.1}$$

For instance, the transfer function of an ideal spring is the ratio of velocity and resulting force output, $\frac{K}{s}$, where K is the spring stiffness,

$$Z_K(s) = \frac{KX(s)}{sX(s)} = \frac{K}{s} \tag{4.2}$$

Equation 4.3 represents the general impedance relation describing the Spring-Mass-Damper system. A controller based on this expression tracks a reference trajectory and rejects disturbance forces governed by a simple spring-mass-damper model (Figure 4.10).

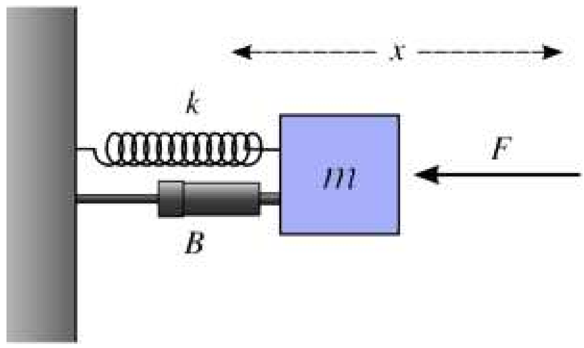


Figure 4.10. Desired impedance behavior modeled as a Spring-Mass-Damper system.

$$M\ddot{X} + B(\dot{X} - \dot{X}_o) + K(X - X_o) = F_e \tag{4.3}$$

where $X = [x\ y\ z]^T$ is the 3×1 vector of the task space coordinates, and X_o, $\dot{X}_o$ are the desired trajectory and velocity of the end effector. The forces $\in \Re^3$ derived from Cartesian space impedance law (Equation 4.3) can be mapped into configuration space

$$\tau_e = J^T(q)F_e \tag{4.4}$$

and applied to the whole-body computed torque dynamic equation of motion to yield

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + \tau_e = \tau_{act} \tag{4.5}$$

where $H(q)$ is the configuration dependent inertia matrix, $C(q,\dot{q})$ is the Coriolis and centrifugal term, $G(q)$ is the gravitational term, and τ_e is the disturbance force from environment on mass.

The expression for the nonlinear feedback law for impedance control in operational (task) space of manipulator using Jacobian is given by Hogan [24]:

$$\begin{aligned}
\tau_{act} &= G(q) + C(q,\dot{q})\dot{q} - HJ^{-1}\dot{J}\dot{q} - J^T F_e \\
&\quad + HJ^{-1}M^{-1}F_e + HJ^{-1}M^{-1}K(X_o - X) \\
&\quad + HJ^{-1}M^{-1}B(\dot{X}_o - \dot{X}) \qquad (4.6)
\end{aligned}$$

The error in position and velocity is multiplied by the desired interaction stiffness K and damping B terms, resulting in Cartesian PD feedback gains. Sensing of the interaction for F_e is required in order to achieve an apparent interaction inertia M that is different from the manipulator's inertia.

a) uBot-5 Impedance Control Modes

The structure of a task can indicate impedance parameterizations that complement environmental dynamics. For many operations, such as crank turning, a useful strategy is to set the robot impedance to be high in directions where environmental impedances are low and vice versa. Therefore, the robot will produce high forces in response to velocity errors along the tangent of the crank circle and low forces in response to velocity errors radially. Low impedance provides a safe mode for interacting with unknown geometries in a manner that keeps forces low. Whole-body impedance specifications can, likewise, be used effectively in the representation and control of mobile manipulators. For example, a "guarded move" can exploit compliant (low impedance) configurations to bound contact forces and during tactile exploration. High impedance can be useful during precision object manipulation and forceful interaction tasks. Whole body impedance configurations can be used to push a heavy door open efficiently, i.e. without creating unwanted and wasteful internal forces.

Several basic whole-body impedance "modes" are anticipated including: an impact dissipation mode, a low impedance safety mode, and a high impedance mode for precision endpoint control. Hannigan [19] demonstrated that anisotropic stiffness in Cartesian space could be used in the uBot-5 arms to address haptic probing actions that were precise in subspaces of $\Re^3$ and compliant in the complementary subspace. Forceful probing tests have also been performed to identify performance limits of force control implementations in uBot hardware.

Bracing for a fall requires a sequential impedance specification. For example, a bracing reflex begins by quickly moving the arms to a bracing position (stiff/high impedance). When the bracing arm configuration is achieved, an anisotropic operational space impedance specification is utilized that is compliant in the vertical direction and stiff parallel to the ground plane so that bracing contacts don't "kick out" during a fall. During impact absorption, the arms to dissipate energy as they compress so that forces can be minimized and the uBot's gearheads, joints, cables, and electronics are protected.

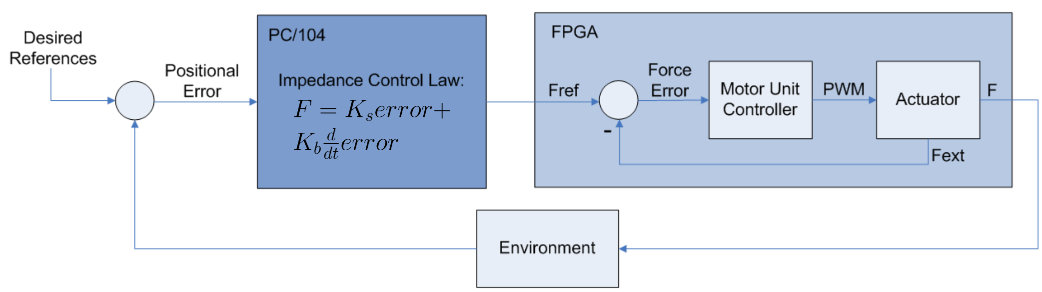


Figure 4.11. Impedance control feedback diagram.

Figure 4.11 illustrates the impedance controller that provides reference forces to an embedded force servo and the underlying control law that relates displacements (positional errors due to contact with the environment) to commanded forces, F_{ref}. Compliance achieved in this manner requires intrinsic force feedback. However, the uBot-5 can also operate without force feedback by modifying the parameters of a joint's low-level PID position controller with task specific spring and damper parameters. In this mode, the I-term of the PID is set to zero. Unfortunately, without force feedback the output impedance will diverge from the desired impedance due in part by significant passive properties of the system (e.g. a motor with a large gear reduction will appear as an large inertial term due to rotor mass). Furthermore, in some cases the uBot-5 implementation utilizes a damper term of zero as well since the system is already well damped by drivetrain friction.

4.1.3 Balance Control

In addition to supporting general-purpose postural control, balance control can free the upper body for manipulation tasks. Motor controllers for today's humanoid robots rarely intentionally exploit the coupling of dynamic balance and manipulation. Bipedal humanoids usually control the position of the Zero Moment Point (ZMP) within the supporting polygon of the feet. The result is a quasi-statically stable mobility platform. Dynamic balance of the uBot employs a closed-loop controller that maintains the inverted pendulum at a near vertical posture [40]. In practice, we employ a linearized model of the cart-pole system near the reference vertical posture. The feedback structure of the closed loop controller makes it robust to bounded disturbances.

Inertial sensors on the robot as well as wheel encoders resolve chassis position, velocity, tilt, and tilt rate about the wheel axis. Using the state space representation of the inverted pendulum, we determine the control input (wheel torque) to balance the robot vertically and minimize translational error.

Figure 4.12 shows inertially fixed axes $\hat{\mathbf{i}}$ and $\hat{\mathbf{k}}$ and a vector $\mathbf{r}$ describing the position of the center of mass of the pendulum. The vector $\mathbf{r}$ can be expressed as:

$$\mathbf{r} = x\hat{\mathbf{i}} + l(\hat{\mathbf{i}}\sin\theta + \hat{\mathbf{k}}\cos\theta)$$

The first derivative of $\mathbf{r}$ is

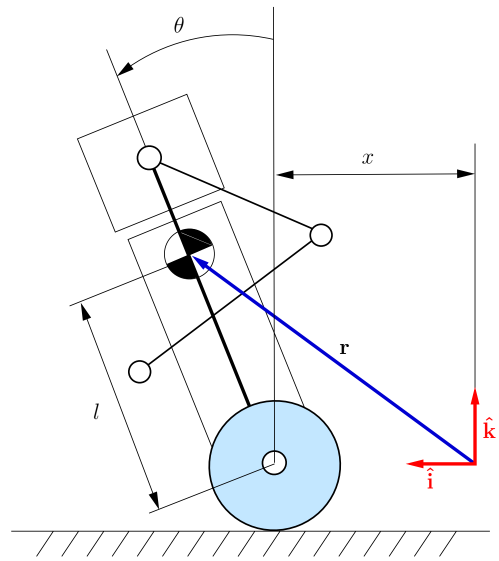


Figure 4.12. Position vector of the pendulum used to determine acceleration components.

$$\dot{\mathbf{r}} = \dot{x}\hat{\mathbf{i}} + l\dot{\theta}(\hat{\mathbf{i}}\cos\theta - \hat{\mathbf{k}}\sin\theta)$$

and the second derivative of $\mathbf{r}$ is

$$\ddot{\mathbf{r}} = \ddot{x}\hat{\mathbf{i}} + l\ddot{\theta}(\hat{\mathbf{i}}\cos\theta - \hat{\mathbf{k}}\sin\theta) - l\dot{\theta}^2(\hat{\mathbf{i}}\sin\theta + \hat{\mathbf{k}}\cos\theta)$$

Figure 4.13 shows the free body diagram of the uBot-5 represented as a cart-pole system. Summing the forces of the cart in the horizontal direction, yields the following equation of motion:

$$M\ddot{x} + b\dot{x} + N = u \tag{4.7}$$

where u is the force, F exerted by the wheels of the uBot on the ground.

Summing the forces of the pendulum in the horizontal direction

$$N = m\ddot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta. \tag{4.8}$$

Substituting Equation 4.8 into Equation 4.7, creates the first dynamic equation for the system.

$$(M+m)\ddot{x} + b\dot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta = u \tag{4.9}$$

For the second equation of motion, summing the forces acting perpendicular to the pendulum will simplify the algebra. This results in the following equation

$$P\sin\theta + N\cos\theta - mg\sin\theta = ml\ddot{\theta} + m\ddot{x}\cos\theta \tag{4.10}$$

To eliminate P and N terms in the equation above, the moments around the centroid of the pendulum are summed to yield:

$$-Pl\sin\theta - Nl\cos\theta = I\ddot{\theta} \tag{4.11}$$

Combining equations 4.10 and 4.11 determines the second dynamic equation:

$$(I+ml^2)\ddot{\theta} + mgl\sin\theta = -ml\ddot{x}\cos\theta. \tag{4.12}$$

The system equations are linearized about the vertical posture by applying a small angle assumption, $\theta \approx 0$. Substituting $\cos\theta = 1$, $\sin\theta = \theta$, and $\dot{\theta}^2 = 0$ into Equations 4.9 and 4.12 yields:

$$\begin{aligned}(I+ml^2)\ddot{\theta} + mgl\theta &= -ml\ddot{x}, \text{ and} \\ (M+m)\ddot{x} + b\dot{x} + ml\ddot{\theta} &= u. \end{aligned} \tag{4.13}$$

Rearranging the linearized equations of motion into state space representation:

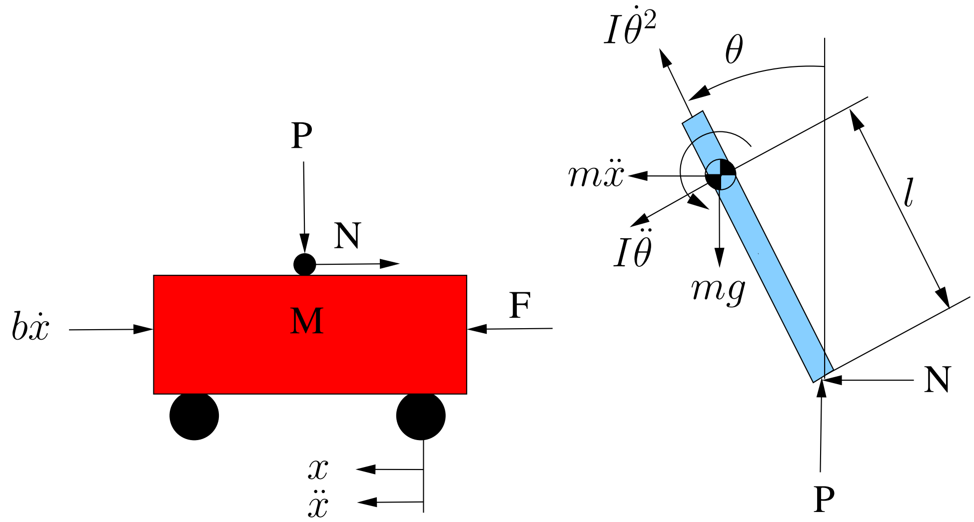


Figure 4.13. Free Body Diagram of inverted pendulum model.

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-(I+ml^2)b}{I(M+m)+Mml^2} & \frac{m^2gl^2}{I(M+m)+Mml^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{mlb}{I(M+m)+Mml^2} & \frac{-mgl(M+m)}{I(M+m)+Mml^2} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{I+ml^2}{I(M+m)+Mml^2} \\ 0 \\ \frac{-ml}{I(M+m)+Mml^2} \end{bmatrix} \mathbf{u}$$

$$\mathbf{y} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \mathbf{u} \tag{4.14}$$

where $\mathbf{y} = [x\theta]^T$ is the output of the system.

The torque required at the wheels, u, to stabilize the pendulum about a vertical orientation is given by a Linear Quadratic Regulator (LQR) feedback control law with gains, $\mathbf{k}$

$$u = -\mathbf{k}^T\mathbf{x}. \tag{4.15}$$

The goal of the controller is to keep output variables x and θ near zero.

The advantages of the LQR over pole-placement methods is that the LQR results in an optimal controller for the output variables, x and θ, with respect to cost function, $\mathcal{J}$:

$$\mathcal{J} = \int_0^\infty (\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{u}^T\mathbf{R}\mathbf{u})dt \tag{4.16}$$

where $\mathbf{Q}$ and $\mathbf{R}$ are specified to determine a reasonable trade-off between performance $\mathbf{Q}$ and control effort $\mathbf{R}$. For the state space representation of the system specified in Equation 4.14,

$$\dot{\mathbf{x}} = A\mathbf{x} + \mathbf{b}u, \tag{4.17}$$

the value of $\mathbf{k}$ in Equation 4.15 that minimizes the cost function is

$$\mathbf{k} = R^{-1}\mathbf{b}^T S. \tag{4.18}$$

Matrix S, in Equation 4.18 is a positive definite symmetric matrix satisfying the matrix Algebraic Riccati equation

$$A^T S + SA - S\mathbf{b}R^{-1}\mathbf{b}^T S + Q = 0 \tag{4.19}$$

It was well documented that the LQR is an improvement over PID control techniques [42]. The algebraic Riccati equation is easy to solve, either by matrix factorizations or by iterating on the Riccati equation. The implementation of the controller on the uBot involved initializing the parameters

$$\mathbf{R} = 1, \text{and } \mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

for the cost function and manually tuning them to optimize the quantitative performance.

4.2 Sequential Controllers in the PC/104

The development of mobile manipulation behavior in humans appears to follow a pattern of staged learning that accumulates control for reuse in subsequent tasks. One of the most rewarding side effects of learning to walk is that the infant requires a mobility mode that leaves hands free to pursue new or multiple objectives (i.e. to enhance the potential for dexterity). It appears that the development participates in the creation of context driven behavior. For this reason, we equate robot programming to be the behavioral equivalent of development in biological systems.

Although walking control and dynamic stability initially require significant physical and computational resources and they are risky compared to crawling, they eventually become automatic. We define dexterous mobility as the fluid and autonomous reassignment of effectors for the purpose of maintaining control in a variety of unanticipated situations.

In humans, bipedalism made limbs available for manipulation, communication, and tool use. Evidence suggests that a tremendous explosion of brain evolution accompanied the evolution of the hand and bipedal lower bodies to explore these new and incredibly rich contexts for interaction. The brain evolved and cognitive structure emerged to make use of an increased platform functionality. The remarkable feature of cognitive development is its ability to use knowledge to address incrementally harder problems. Structure should be reused and hierarchies should form by accumulating successful control compositions. Just as development structures the formation of neural organizations, control should layer complex, adaptive motor solutions on top of fixed, reflexive, and automatic processes. Thus, the accumulation of control structure should start from the bottom and work up, incorporating successful behavior into the working set of available control.

Taking cues from infant development, the uBot-5 addresses stability and other WBMM applications as sequential combinations of more primitive controllers. This includes critical functionality related to postural stability and captured as firmware implemented on the PC/104 stack.

Postural stability in uBot-5 is represented in the form of a suite of controllers that transition between discrete postural modes: from prone to a bracing 4-point stance, to 3-point stances, and finally to a 2-point, balancing mode with arms and hands free. This control suite includes and "emergency" bracing controller to provide transitions from 2-point stances to a 4-point, energy dissipation mode in the event of a fall.

Postural stability firmware will influence every subsequent, task-level application for this WBMM concept. As such, it is not a discretionary control application– it is a necessary complement to the dexterous mechanical design.

In this chapter, I will introduce postural stability firmware designed to maintain platform viability and to provide a functional scaffolding for future applications (some of which will be introduced in Chapter 5). The discussion focuses on controller that perform transitions between discrete postural modes.

4.2.1 The Control Basis

In this section, we describe a combinatoric framework for constructing closed-loop controllers. The control basis is defined by three sets: navigation functions Ω_ϕ, feedback signals Ω_σ, and motor resources Ω_τ—the latter two are defined exclusively at design time, grounded in the robot's sensors and actuators. Ω_ϕ, in contrast, describes a set of potential functions that serve as primitive subtasks in integrated behavioral programs.

Primitive controllers, like the motor units (FPGA fabric) and the LQR for balancing (FPGA core) have been defined previously. However, postural stability in a machine like the uBot-5 requires higher-level, sequential control organizations to support discrete mobility mode changes– the hallmark of a dexterous mobility device. The Control Basis API (CBAPI) was developed at the Laboratory for Perceptual Robotics to create programming interfaces to dexterous robots [22]. The CBAPI supports the construction of sequential actions combinatorially by binding sensor and effector resources to navigation functions to create closed-loop controllers. This approach also addresses multiple objectives and expresses constraints for safety. To manage the combinatorial complexity, constraints in the form of valid control combinations are expressed within the control basis framework.

The Control Basis framework was used to express sequential programs for postural stability in the uBot-5 in the form of transitions between postural modes and policies for recovering when postural stability is threatened. Simultaneous objectives, such as manipulation can be addressed in a control configuration that is guaranteed *a priori* to preserve postural stability. The representation also specifies which actions the policy should choose if stability (the primary objective) is jeopardized. Therefore, this work provides the functional preconditions of subsequent manipulation behavior.

Primitive actions in the control basis framework are closed-loop feedback controllers constructed by combining a potential function $\phi \in \Omega_\phi$, with a feedback signal $\sigma \in \Omega_\sigma$, and motor resources $\tau \in \Omega_\tau$. The notation we use to describe a controller is $c(\phi, \sigma, \tau)$. In any such configuration, $\phi(\sigma)$ is a scalar *navigation* function [34] defined to satisfy properties that guarantee asymptotic stability.

Examples of potential functions include fields that describe kinematic conditioning [21], harmonic functions for collision-free motion [29], and force closure functions [46]. In this work, we will make extensive use of a simple quadratic function,

$$\phi(\sigma) = (\sigma - \sigma_{ref})^T(\sigma - \sigma_{ref}),$$

where both σ and σ_{ref} are determined from features in the feedback signals and ϵ denotes the feedback error. Potential function $\phi(\sigma)$ can be used to compute quadratic error functions in generic spaces (workspace, configuration space, force space, and/or torque spaces).

The sensitivity of the potential to changes in the value of motor variables is captured in the error Jacobian, $J = \partial\phi(\sigma)/\partial u_\tau$, where u_τ are real-valued control inputs to lower-level motor units (on the FPGA) are designated by resources, τ,

$$\Delta u_\tau = J^{\#}\phi(\sigma),$$

and $J^{\#}$ is the pseudo-inverse of J [41].

Multi-objective control actions are constructed by concurrent control actions. Concurrency is managed by projecting subordinate/inferior actions into the nullspace of superior actions.

$$\Delta u_\tau = J^{\#}_{sup}\phi_{sup} + \left[I - J^{\#}_{sup}J_{sup}\right] J^{\#}_{inf}\phi_{inf}. \tag{4.20}$$

This prioritized mapping assures that inferior control inputs do not destructively interfere with superior objectives, and this approach can be extended to n-fold concurrency relations. In the following, we will use a shorthand for the nullspace projection that uses the "subject-to" operator "$\triangleleft$." The control expression $c_{inf} \triangleleft c_{sup}$—read, "$c_{inf}$ *subject-to* c_{sup}"—is shorthand for Equation 4.20.

The combinations of potentials Ω_ϕ, and resources Ω_σ and Ω_τ defines all primitive closed-loop actions $\mathcal{A}$ that the robot can employ. Previous work by Huber [27, 26] and Platt [45] addresses how to direct exploration using constraints on $\mathcal{A}$ and how such programs generalize to new contexts.

The error dynamics ($\phi(\sigma)$, $\dot{\phi}(\sigma, \tau)$) created when a controller interacts with the task domain supports a natural discrete abstraction of the underlying continuous state space [30]. In this work, we will use a very simple discrete state definition based on *convergence events*. In the following, we define a Boolean predicate $p(\phi, \dot{\phi})$ associated with a controller

$$p(\phi, \dot{\phi}) = \begin{cases} X & : \ \phi \text{ state is unknown} \\ - & : \ \phi \text{ has undefined reference} \\ 0 & : \ |\dot{\phi}| > \epsilon_\tau \\ 1 & : \ |\dot{\phi}| \leq \epsilon_\tau, \end{cases} \tag{4.21}$$

where ϵ_τ is a small positive constant that depends on effector resources τ. For asymptotically stable controllers, ϕ is positive definite and $\dot{\phi}$ is negative definite. The "$-$" condition means that no target stimuli is present in the feedback signal. A collection of n distinct primitive control actions forms a discrete state space $\mathcal{S} \equiv [p_1, \cdots, p_n)]$.

Within the Control Basis Framework, integrated behaviors are generated by combining and sequencing primitive controllers in a prioritized manner. In the following, we will derive some of the principle controllers required to construct robust transitions between the uBot-5's postural modes.

4.2.2 Controlling Postural Transitions

The mobility behavior for the uBot include several postural modes: prone (face down), 4-point, 3-point, and 2-point (balancing) stances. Transitions between these modes were implemented as control firmware on the PC/104. such as balancing, crawling, scooting, pushing, and bracing were implemented. The integrated policy for "righting" and moving in an upright posture forms a behavioral foundation for subsequent control applications for this robot.

The behavior generated in this Section use the Control Basis to represent sequences of composite controllers. The primitive operator in the Control Basis is the

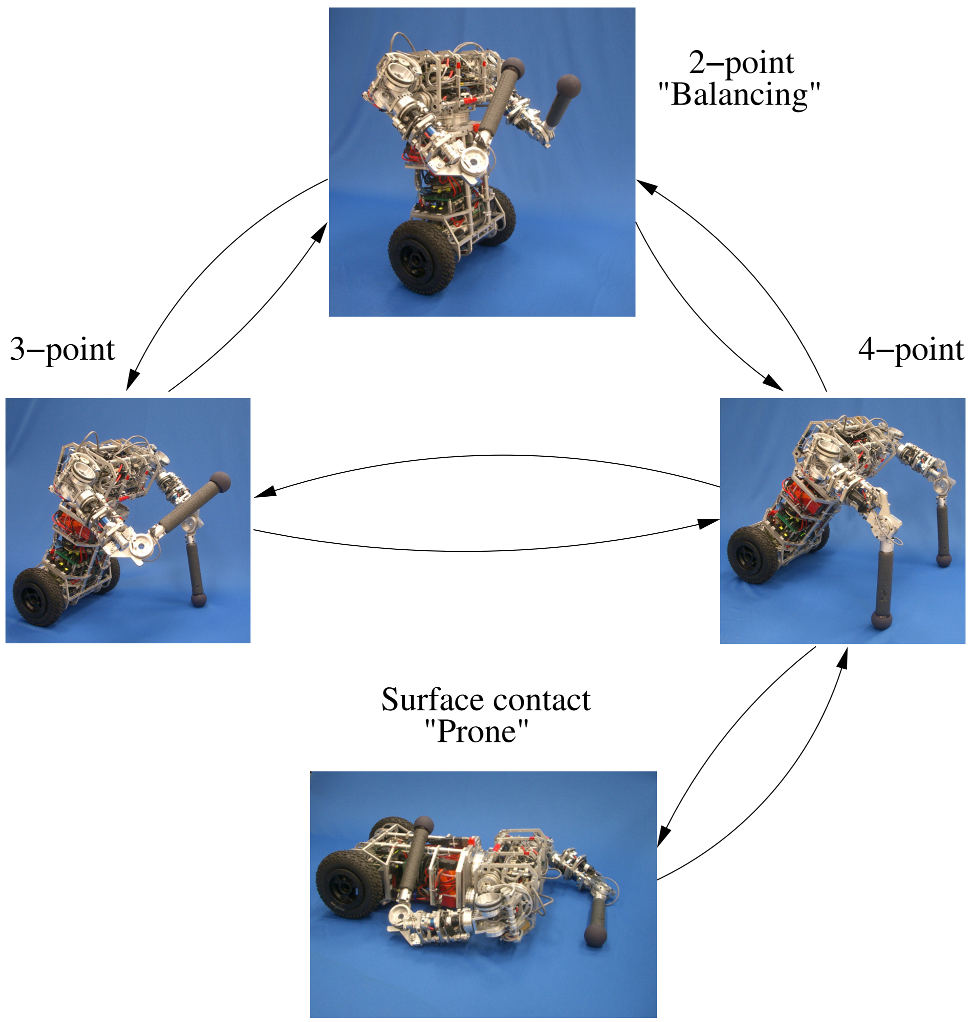


Figure 4.14. The uBot-5 postural stability transition graph.

closed-loop controller. Fully specified, the controller includes sensor and effector resources bound to an artificial potential specifying the objective for a class of controllers. The sensor allocation binds the feedback for artificial potentials to a specific set of sensory signals. The uBot-5 has several sensor values that are continuously updated as part of the firmware of the robot. These signals present joint angle and velocity feedback from all the joints and current (torque) feedback from each of the 4-DOF in the arms. Vestibular feedback from inertial sensors provides body tilt angle or pitch. In addition, several platform coordinates are available as potential control variables, these include the (x,y) position of the body and heading from odometry.

The effector resources specify the actuated degrees of freedom that accomplish the control objective [46]. The effectors consist of all 11 degrees-of-freedom representing each of the uBot's arms and two drive wheels.

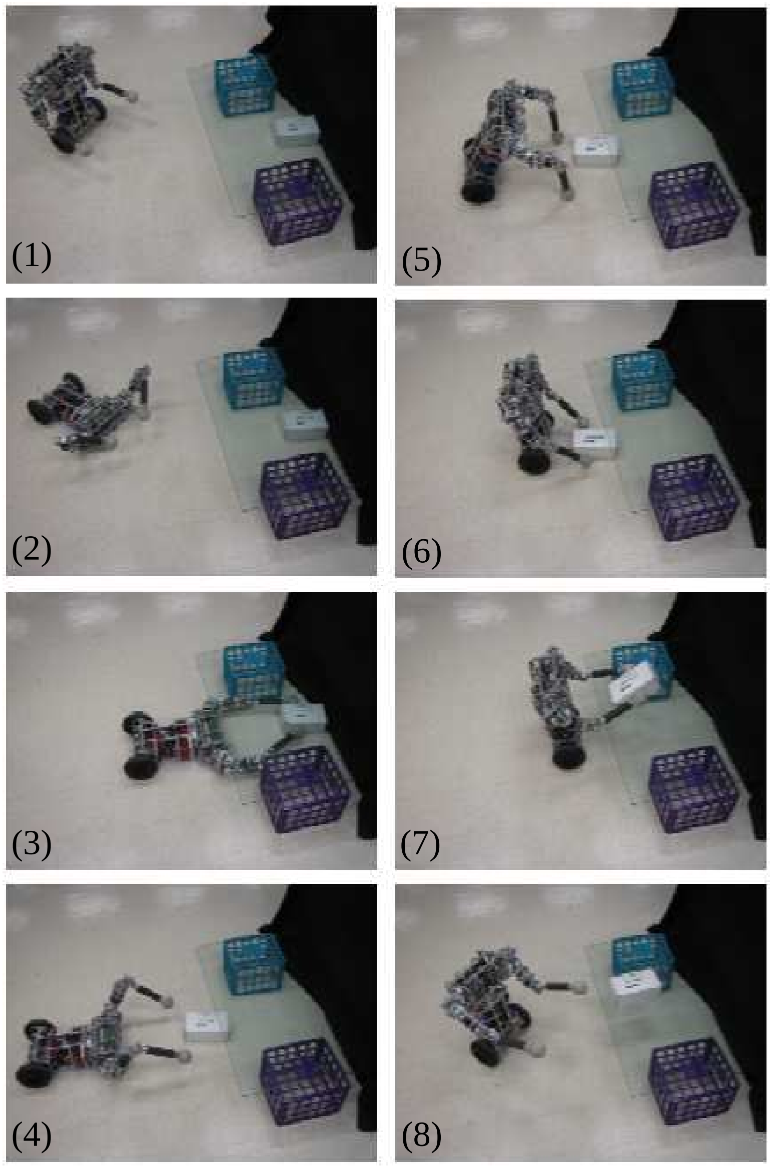


Figure 4.15. The uBot-5 performing a fetch task requiring transitions between several postural stability modes including: (1) 2-point stance, (3) prone, and (5) 4-point stance.

Figure 4.14 shows the transitions between postural modes for uBot-5. From prone, or the 3- or 4-point stance, the robot can push-up further into a nearly vertical posture from which the robot can transition to a 2-point balancing postural mode. In this mode, both hands are free to serve as manipulation devices as well as stabilizing mechanisms for mobility. These modes and transitions constitute an initial organization of whole-body postural controllers with the uBot-5 and encodes recovery strategies to ensure postural stability for subsequent tasks. Figure 4.15 illustrates an integrated policy with various mode transitions in order to fetch an object from underneath a table.

4.2.2.1 Prone to 4-point Stance Transition

Often, when doing push-ups the elbow would collapse during the motion and success was very sensitive to battery charge. Rather than swapping motor/gearboxes, a closed-loop controller was implemented to offset this problem. The closed-loop controller reduced slippage by regulating interaction forces parallel to the ground plane. To protect the relatively weak elbow joint, the control strategy maintains the forearm perpendicular to the ground, while the shoulder and base wheels transition the platform between prone and full upright stances. This kinematic condition minimizes the torque required by the elbow motor, and is essential for repeatable push up motion.

The push-up behavior is conceived as the combination of two objectives. The first is related to protecting the relatively weak elbow joint in uBot-5 from the large forces involved in the push-up maneuver. To do so, we design a primary control task that constrains the attitude of the forearm with respect to the ground.

The second objective is to raise the shoulder of the robot vertically from the prone position up to an upright posture that bears most of the robot's weight on the wheels. The resulting 4-point stance is very stable. However, the motors in the upper limbs are operated at or beyond their thermal limits. The 3-point stance is even more demanding and the system can only maintain this posture statically for very short periods.

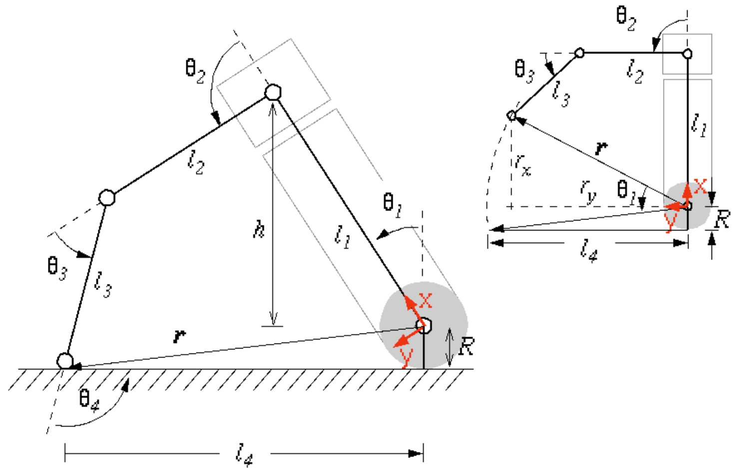


Figure 4.16. The geometric definitions underlying the push-up controllers.

Figure 4.16 provides a kinematic description of the robot that is sufficient to design the controllers for the push-up maneuver. In this application, configuration variables include the attitude of the body relative to vertical, θ_1, joint angles θ_2 and θ_2, the relative orientation of the forelimb with respect to the ground plane, θ_4, and the distance from the contact on the limb endpoint to the wheels, l_4.

$$\mathbf{q} = [\theta_1 \ \theta_2 \ \theta_3 \ \theta_4 \ l_4]$$

If we assume that the wheels and the arms remain in contact with the horizontal ground plane, the resulting closed-chain mechanism has only two degrees of freedom. The uBot-5 measures joint angles θ_2 and θ_3 directly using joint encoders and we determine the value of the rest of the configuration variables (θ_1, θ_4, and l_4) using the kinematic relations of the closed-chain mechanism.

The inset on the right of Figure 4.16 illustrates the uBot in an vertical posture with the same upper body posture posture. The endpoint position of the arm position in the coordinate frame indicated is written:

$$\begin{aligned} r_x &= l_1 + l_2 cos(\theta_2) + l_3 cos(\theta_2 + \theta_3) \\ r_y &= l_2 sin(\theta_2) + l_3 sin(\theta_2 + \theta_3) \end{aligned} \tag{4.22}$$

Given θ_2 and θ_3, the value of θ_1 that brings the hand into contact with the ground plane is

$$\theta_1 = atan2(r_x, r_y) + atan2(R, l_4).$$

The orientation of the forelimb with respect to the ground plane is determined from this closed-chain by observing that

$$\theta_1 + \theta_2 + \theta_3 + \theta_4 + \pi/2 = 2\pi.$$

so that

$$\theta_4 = \frac{3\pi}{2} - (\theta_1 + \theta_2 + \theta_3).$$

To compute the last configuration variable, l_4, we note that

$$\begin{aligned} |\mathbf{r}|^2 &= r_x^2 + r_y^2 = R^2 + l_4^2, \text{ and} \\ l_4 &= [r_x^2 + r_y^2 - R^2]^{\frac{1}{2}}. \end{aligned}$$

A multi-objective controller for the transition from prone to the 4-point posture is designed using potential fields to map feedback, $\sigma = [\theta_2\ \theta_3]$, to a stream of reference inputs for motor units $\Delta u_\tau = [\Delta\theta_2\ \Delta\theta_3\ \Delta l_4]$ in a manner that reflects the independent objectives of the movement: (1) forelimb attitude, and (2) shoulder elevation. Since l_4 changes throughout the push-up behavior, we must also send reference positions to the wheels consistent with the changes in θ_2 and θ_3 to avoid contention between the arm and wheel motors due to ground friction. All other motor units in the robot are held fixed, while these closed-loop systems implement the push-up behavior.

The first objective, is described by requiring that θ_4 be maintained near $\theta_{4,ref} = \pi/2$ radians to minimize the torque on the elbow during the push-up. The error therefore can be expressed as:

$$\epsilon = (\theta_{4,ref} - \theta_4) = \theta_1 + \theta_2 + \theta_3 - \pi$$

The forearm orientation constraint can be represented as a quadratic potential function (an analog of Hooke's law) whose Jacobian describes the sensitivity of the orientation constraint with respect control variables, θ_2 and θ_3.

$$J_{FPC} = \left[\frac{\partial \epsilon^2}{\partial \theta_2} \;\; \frac{\partial \epsilon^2}{\partial \theta_3} \right], \text{ where}$$

$$\frac{\partial \epsilon^2}{\partial \theta_2} = 2\epsilon \frac{\partial \epsilon}{\partial \theta_2} = 2\epsilon \frac{\partial}{\partial \theta_2} [\theta_1 + \theta_2 + \theta_3 - \pi] = 2\epsilon \left[\frac{\partial \theta_1}{\partial \theta_2} + 1 \right]$$

$$\frac{\partial \epsilon^2}{\partial \theta_3} = 2\epsilon \frac{\partial \epsilon}{\partial \theta_3} = 2\epsilon \frac{\partial}{\partial \theta_3} [\theta_1 + \theta_2 + \theta_3 - \pi] = 2\epsilon \left[\frac{\partial \theta_1}{\partial \theta_3} + 1 \right]$$

and

$$
\begin{aligned}
\frac{\partial\theta_1}{\partial\theta_2} &= \left[\frac{1}{1+(r_x/r_y)^2}\right]\frac{\partial}{\partial\theta_2}\left[r_x r_y^{-1}\right] + \left[\frac{1}{1+(R/l_4)^2}\right]\frac{\partial}{\partial\theta_2}\left[R(r_x^2+r_y^2-R^2)^{-1/2}\right] \\
&= \left[\frac{1}{1+(r_x/r_y)^2}\right]\left[r_x(-1)r_y^{-2}(\frac{\partial}{\partial\theta_2}r_y) + (\frac{\partial}{\partial\theta_2}r_x)r_y^{-1}\right] \\
&\quad + \left[\frac{1}{1+(R/l_4)^2}\right]\left[-R(r_x^2+r_y^2-R^2)^{-3/2}(r_x(\frac{\partial}{\partial\theta_2}r_x) + r_y(\frac{\partial}{\partial\theta_2}r_y))\right] \\
&= \left[\frac{1}{1+(r_x/r_y)^2}\right]\left[\frac{r_x(-1)(l_2cos(\theta_2)+l_3cos(\theta_2+\theta_3))}{r_y^2} + \frac{-l_2sin(\theta_2)-l_3sin(\theta_2+\theta_3)}{r_y}\right] \\
&\quad + \left[\frac{-R(r_x^2+r_y^2-R^2)^{-3/2}}{1+(R/l_4)^2}\right]\left[\begin{array}{l} r_x(-l_2sin(\theta_2)-l_3sin(\theta_2+\theta_3)) \\ +\, r_y(l_2cos(\theta_2)+l_3cos(\theta_2+\theta_3)) \end{array}\right] \\
&= \left[\frac{-1}{(1+(r_x/r_y)^2)r_y}\right]\left[\frac{r_x(l_2cos(\theta_2)+l_3cos(\theta_2+\theta_3))}{r_y} + l_2sin(\theta_2)+l_3sin(\theta_2+\theta_3)\right] \\
&\quad + \left[\frac{R}{(1+(R/l_4)^2)(r_x^2+r_y^2-R^2)^{3/2}}\right]\left[\begin{array}{l} r_x(l_2sin(\theta_2)+l_3sin(\theta_2+\theta_3)) \\ -\, r_y(l_2cos(\theta_2)+l_3cos(\theta_2+\theta_3)) \end{array}\right]
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial\theta_1}{\partial\theta_3} &= \left[\frac{1}{1+(r_x/r_y)^2}\right]\frac{\partial}{\partial\theta_3}\left[r_x r_y^{-1}\right] + \left[\frac{1}{1+(R/l_4)^2}\right]\frac{\partial}{\partial\theta_3}\left[R(r_x^2+r_y^2-R^2)^{-1/2}\right] \\
&= \left[\frac{1}{1+(r_x/r_y)^2}\right]\left[r_x(-1)r_y^{-2}(\frac{\partial}{\partial\theta_3}r_y) + (\frac{\partial}{\partial\theta_3}r_x)r_y^{-1}\right] \\
&\quad + \left[\frac{1}{1+(R/l_4)^2}\right]\left[-R(r_x^2+r_y^2-R^2)^{-3/2}(r_x(\frac{\partial}{\partial\theta_3}r_x) + r_y(\frac{\partial}{\partial\theta_3}r_y))\right] \\
&= \left[\frac{1}{1+(r_x/r_y)^2}\right]\left[\frac{r_x(-1)(l_3cos(\theta_2+\theta_3))}{r_y^2} + \frac{-l_3sin(\theta_2+\theta_3)}{r_y}\right] \\
&\quad - \left[\frac{R(r_x^2+r_y^2-R^2)^{-3/2}}{1+(R/l_4)^2}\right]\left[r_x(-l_3sin(\theta_2+\theta_3)) + r_y(l_3cos(\theta_2+\theta_3))\right] \\
&= \left[\frac{-1}{(1+(r_x/r_y)^2)r_y}\right]\left[\frac{r_xl_3cos(\theta_2+\theta_3)}{r_y} + l_3sin(\theta_2+\theta_3)\right] \\
&\quad + \left[\frac{R}{(1+(R/l_4)^2)(r_x^2+r_y^2-R^2)^{3/2}}\right]\left[r_xl_3sin(\theta_2+\theta_3) - r_yl_3cos(\theta_2+\theta_3)\right]
\end{aligned}
$$

The second objective of the push-up action is to elevate the shoulder of uBot-5 to vertical height l_1 in the world coordinate frame. This objective is stated most succinctly in terms of the postural variable θ_1, that is the reference value for θ_1, $\theta_{1,ref} = 0$[1]. Under these circumstances,

$$\epsilon = (\theta_{1,ref} - \theta_1) = -\theta_1.$$

The push-up objective is the square of error ϵ and the Jacobian, J_{PU} describes the sensitivity of the orientation constraint with respect control variables, θ_2 and θ_3.

$$J_{PU} = \begin{bmatrix} \frac{\partial \epsilon^2}{\partial \theta_2} & \frac{\partial \epsilon^2}{\partial \theta_3} \end{bmatrix}, \text{ where}$$

$$\frac{\partial \epsilon^2}{\partial \theta_2} = 2\epsilon \frac{\partial \epsilon}{\partial \theta_2} = 2\theta_1 \frac{\partial \theta_1}{\partial \theta_2}$$

$$\frac{\partial \epsilon^2}{\partial \theta_3} = 2\epsilon \frac{\partial \epsilon}{\partial \theta_2} = 2\theta_1 \frac{\partial \theta_1}{\partial \theta_3}$$

Now that we have precisely defined our two control subtasks, we combine them using the approach to multi-objective control described by 4.20. Namely, we wish to elevate the shoulder joint *subject-to* maintaining a near vertical forelimb attitude. We write this mathematically as:

$$\begin{bmatrix} \Delta\theta_2 \\ \Delta\theta_3 \end{bmatrix} = J_1^{\#} \Delta\phi(\pi/2, \theta_4) + (I - J_1^{\#} J_1) J_2^{\#} \Delta\phi(0, \theta_1)$$

or $c(\epsilon_2^2, \sigma, \tau) \triangleleft c(\epsilon_1^2, \sigma, \tau)$. Here the forelimb postural control objective is higher priority than the shoulder elevation objective. Note that if the priorities were reversed, the

[1] Note that this objective cannot be met if this control task is subject-to constraints on the forearm orientation.

robot might elevate the shoulder more quickly. However, while elevating the shoulder it may violate the constraint on forelimb attitude, which could result in instability due to insufficient elbow motor torques.

4.2.2.2 4-Point to 3-point Stance Transitions

In order to transition to and maintain stable 3-point stances, it is not simply enough to withdraw one endpoint while in a 4-point stance. This would, in most cases, result in destabilizing the robot. Kuindersma [35] developed the following stability criterion and means of achieving a stable 3-point stance.

Commonly, the zero moment point (ZMP) is used as a measure of quasi-static stability. Given a set of N ground contact points, $\{p_1, p_2, \ldots, p_N\}$ and a set of ground reaction forces, $\{f_1, f_2, \ldots, f_N\}$, we use the z-component of the ground reaction force at each point to calculate the ZMP,

$$ZMP = \frac{\sum_{i=1}^{N} p_i f_{iz}}{\sum_{i=1}^{N} f_{iz}}$$

We say the 3-point stance is stable if the robot's center of mass (COM) projects onto the ground plane within some small distance of the ZMP. Thus, the controller that stabilizes the robot in the left 3-point stance must move the ground plane projection of the robot's COM toward the ZMP. The ZMP is calculated for the contacts formed with the ground plane of the wheels and the endpoints of the left arm. The calculation of the ZMP follows similarly for the right 3-point stance and the 4-point stance. Figure 4.17 shows the simplified kinematic model used to define the controller.

We define a quadratic potential function,

$$\phi_T = \phi(\sigma_{com}, \sigma_{zmp}) = (\sigma_{com} - \sigma_{zmp})^T (\sigma_{com} - \sigma_{zmp}),$$

where $\sigma_{com} = [x_c \; y_c]^T$ and $\sigma_{zmp} = [x_z \; y_z]^T$ are signals corresponding to the location of the COM and ZMP in the ground plane. To minimize the potential, we use control resources $\tau = [y_z \; \alpha]$ (Figure 4.17)to produce base translations and rotations on the robot. We make the assumption that for small changes in y_z and α, the ZMP remains fixed in the world frame. While the COM projection remains fixed in the robot base coordinate frame. In practice, this assumption holds because the mass of the body is much greater than the mass of the arms. Moreover, the ground contact forces change very little during small movements.

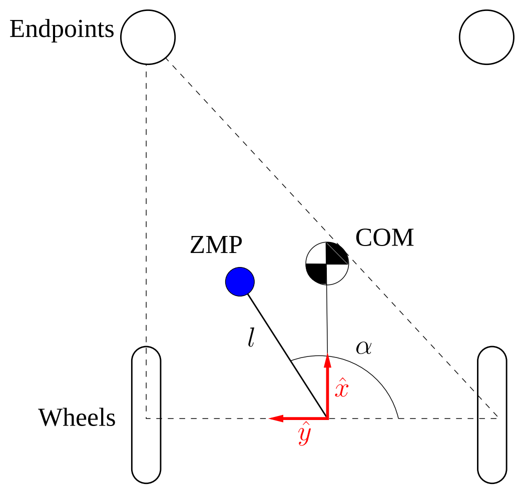


Figure 4.17. Geometry of the uBot-5 in the ground plane. The dashed line represents the support polygon for the left 3-point stance.

Note that unlike the control resources in the push-up controller, y_z and α do not correspond directly to motors on the robot. However, changes in these parameters are converted to specific motor commands, u_τ. For example, a change in y_z can be translated into positional changes in each wheel. Similarly, corresponding changes in each endpoint position can be made to avoid contention between wheel and arm motors. The Jacobian for transitioning to a 3-point stance is:

$$J_3 = \left[\frac{\partial \phi_T}{\partial y_z} \; \frac{\partial \phi_T}{\partial \alpha} \right],$$

where $\frac{\partial \phi_T}{\partial y_z} = 2y_z - 2y_c$ and $\frac{\partial \phi_T}{\partial \alpha} = 2lx_c \sin(\alpha) - 2ly_c \cos(\alpha)$.

4.2.2.3 4-point Stance to 2-point Stance Transitions

The LQR balancing controller cannot be activated unless the robot is postured at or near vertical. From the 3 or 4-point stance, the robot can push-up further by setting the reference body tilt angle to 0 radians. If the posture is near enough to vertical, it can activate the LQR for balancing and withdraw its endpoints to transition to the upright mode. Figure 4.18 shows the uBot-5 autonomously transitioning from a 4-point stance to balancing mode.

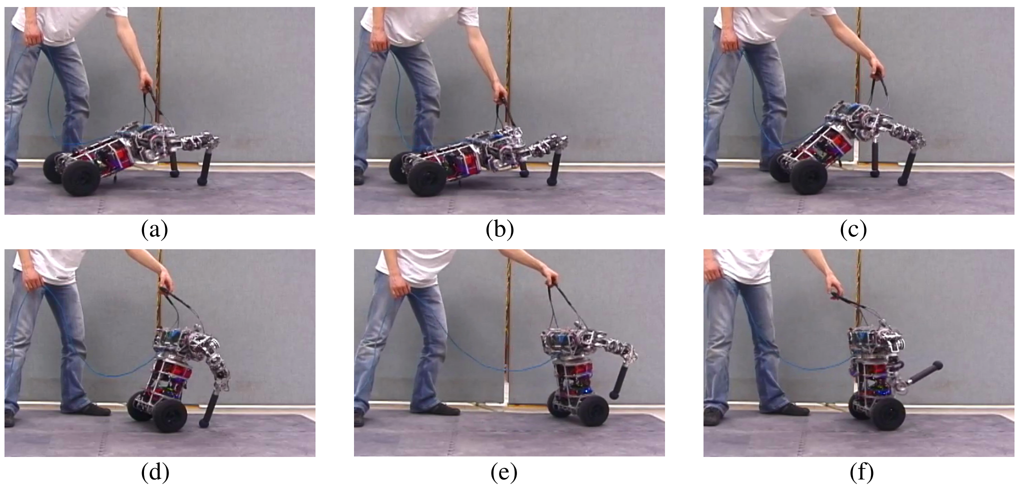


Figure 4.18. Postural Stability FSA transitions on actual robot. a) torso is initially at rest in 4-point stance b) push-up controller is activated d) kinematic conditioning trigger for transition is reached e) balancing controller is activated f) endpoints are withdrawn.

4.2.2.4 2-point Stance to 4-point Stance Transitions

The robot can transition back to prone by positioning its endpoints for a 4-point brace and deactivating the LQR controller. From the 4-point stance, the robot achieves the prone posture by using the push-up controller with a reference body tilt angle of $\pi/2$ radians, resulting in a set-down behavior.

4.2.3 Bracing Reflex

Many animals display vestibular reflexes that enhance stability and protect against falls, mitigate impact, or otherwise control the risk associated with postural dynamics.

The righting reflex in a house cat emerges at between 3 and 7 weeks after birth. It orients the falling animal so that is lands on its feet, protecting the animal from injury. The parachute reflex in humans serve the same purpose. In animals, reflexes not only act as hard-wired safety mechanisms but may also provide primitive control modes that can bootstrap learning.

A dynamically balancing platform with a high center of mass has the potential to transfer large impact energies as a result of a fall. Reflexes in the control firmware of the uBot-5 act to mitigate the risk of falling during dynamic activity in unstructured environments. The uBot-5's "bracing" reflex, shown in Figure 4.19, is an example of a controller that exploits dexterity to provide contingencies for higher order mobile manipulation behavior. When the combined effort of the balancing controller and the upper body fail to preserve postural stability, a vestibular event is triggered. If the arms are engaged in a task, the bracing controller will supersede and withdraw and orient the arms, extending them for impact. The arms then move into a free space configuration prior to the endpoints making contact with the ground. The subsequent mode creates a multi-contact support polygon consisting of one or both hands and wheels (a 3- or 4-point stance). After creating a suitable contact geometry, the bracing reflex employs appropriate impedance modes in the arm to dissipate energy and reduce the impact of the structure with the ground.

Virtually all robot behavior is subordinate to the primitive bracing reflex that protects the robot in the event of an unanticipated fall. This allows mobility capabilities to be explored during the development of dexterous behavior without having to worry about the details and contingencies of recovery. Challenging control situations that are likely to destabilize the balancing controller are very common in human environments. Since balancing will serve as the behavioral foundation for future manipulation tasks, this requires an efficient policy for stability recovery.

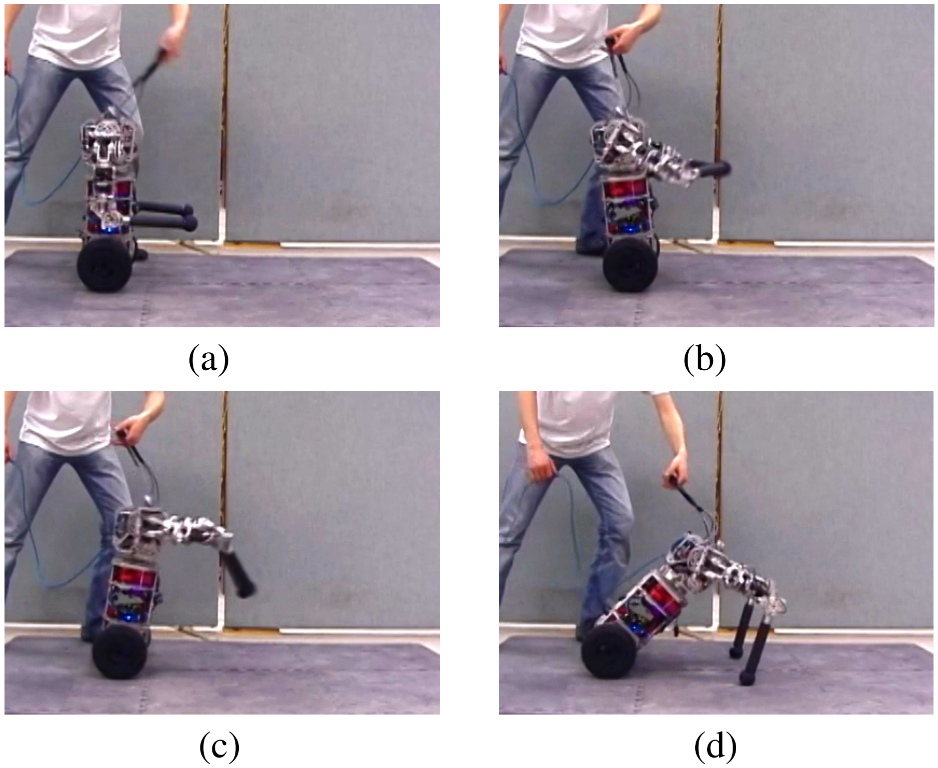


Figure 4.19. Video snapshots showing the transitions of the brace controller FSA. a) A perturbation is applied to the stabilized platform and the bracing reflex is triggered b) Arms withdraw and orient c) Bilateral extension configures arms for impact d) Impact is detected and arms absorb the body's momentum before it comes to rest in 4-point stance or prone configurations.

The reflex motion and automatic impedance mode switching is determined by a logical reflex design, a composition of sequential controllers. The transitions are triggered by events in the states of control objectives. Figure 4.20 shows the finite state automata (FSA) that governs the transitions between control segments. The bracing reflex can be "turned off" if it conflicts with high-level goals. Otherwise, the bracing reflex remains activated. Vestibular feedback indicates when the balance system is approaching stability boundaries. Events on this feedback signal trigger the sequential stages of control illustrated in Figure 4.20. Boundaries in the phase space of the tilt angle error are sufficient conditions to trigger the appropriate motor response in the arms. When the brace reflex is triggered, the arms are **withdrawn** toward the medial axis to a nominal configuration and **oriented** along the longitudinal direction

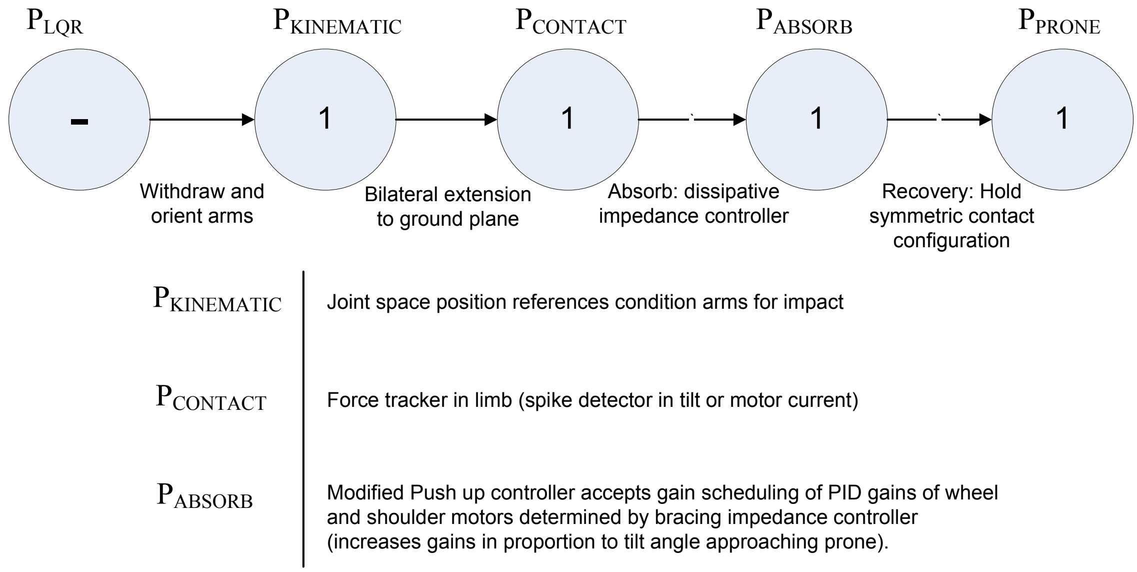


Figure 4.20. Diagram showing the design of the brace controller FSA. Bracing is essentially a "ballistic" open-loop emergency response. The LQR and PRONE controllers are absorbing states in the Postural Stability Control Suite.

of the fall. Before impact, the arms are **extended bilaterally** in the direction of the fall to a point well conditioned fr bracing forces. Once contact is detected, the robot impedance control mode is changed to a **dissipative mode** that absorbs energy as the arms compress toward a prone posture and thus, protects the body of the robot from colliding energetically with the ground or its own arms. The impedance of the shoulder and wheel motors are controlled to absorb impact forces while running the "set down" controller to bring the robot to rest. The gains are increased linearly as the robot nears the ground. After the robot has come to rest, the controllers are returned to high impedance position controlled mode to **recover** and/or reconfigure the robot arms into an acceptable push up configuration.

The stability of the LQR balance controller controller is bounded and experimentation is required to determine the stable operating envelope. Large instantaneous changes in tilt error and the rate of tilt signal an instability in the balance control and triggers a bracing response. Empirical results support representing the trigger in the form of a boundary in the (tilt, tilt rate) space.

Figure 4.21 shows the tilt and rate of tilt values for a brace event with all joint gains set to their default maximum values. The figure shows a sharp spike in rate of tilt indicating a large and undesirable impact deceleration. Furthermore, it is likely that the default gains do not match the task well, such that an oscillation emerges after impact. This discrepancy is likely the result of a coupling between wheel and elbow controllers and poor tuning of the Kalman filter for tilt and rate of tilt sensor readings. Further studies suggested that both the elbow and wheel motors were underpowered. Thus, increasing the damping in the elbow and wheel motors had little effect when the proportional term was also high. With sufficient motor power, it is expected that the response can be tuned with high gains to bring about an over- or critically-damped response. The goal of choosing impedance gains is to achieve an overdamped response that also reduces the impact forces. The magnitude of the impact can be estimated from the sharpness of the inversion of the rate of tilt, and the slope of the deceleration shown in Section b) of Figure 4.21.

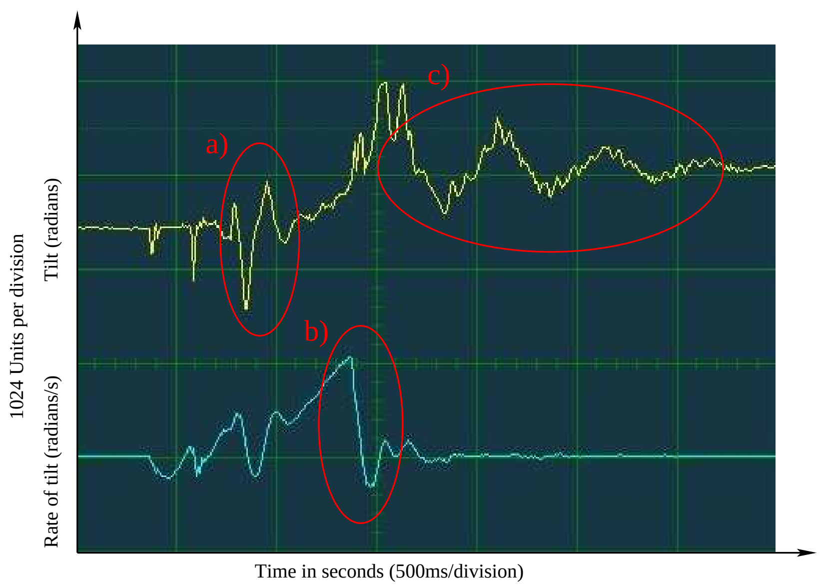


Figure 4.21. Tilt and rate of tilt vs. time during brace. The response shows an underdamped oscillation after impact. Region a) denotes the balancing instability caused by an external disturbance. Region b) shows the sharp reduction in tilt speed at the moment the the arms contact the ground. Region c) shows the oscillation after impact as the platform comes to rest in the 4-point stance.

Figure 4.22 illustrates the result of lowering the gains beyond the point of being able to bring the platform to rest prior to the entire body impacting the ground (the prone configuration). Although the impact magnitude was reduced, the values are unusable. This data demonstrates that a significant deceleration at impact will occur, even when the gains are very low.

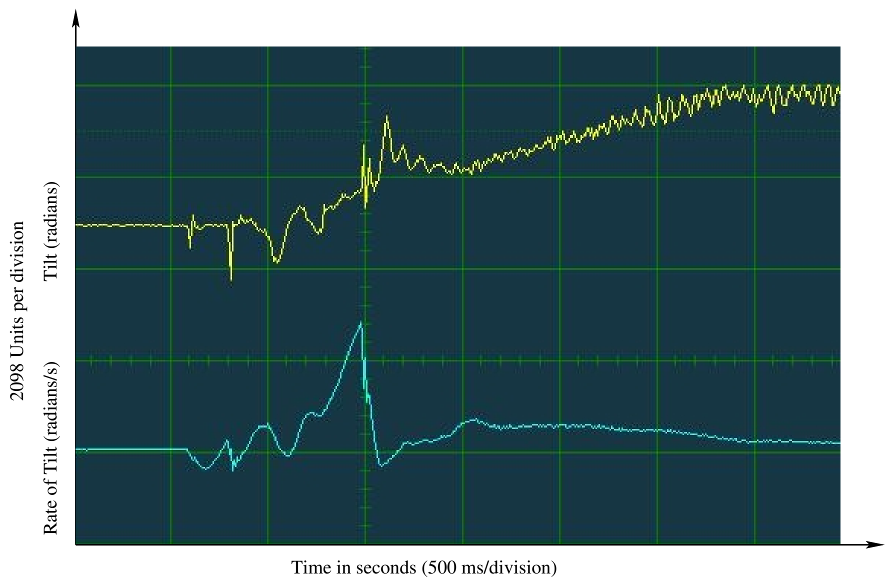


Figure 4.22. Tilt and rate of tilt vs. time during brace. Gains for arm tilt position control (Kp 4000, Ki 0, Kd 200) and wheel velocity control (Kp 150, Ki 0, Kd 400) were used.

Figure 4.23 demonstrates the final values chosen. In this case, the robot experiences a slightly reduced spike in the rate of tilt during impact and comes to rest prior to reaching the prone position. To simultaneously minimize the impact and control during the absorb phase, the impedance controller specified arm tilt gains to be zero at impact and linearly scaled them to 100% as the platform approached the prone position. The wheel gains were set to zero and the elbow gains were reduced to eliminate the dynamic coupling and the resulting oscillation.

During the trials it was discovered that the robot would often brace successfully, even if the shoulder and wheel gains were set to zero. In these cases, the impact would be absorbed without displacing the tilt joint. It was likely that without a

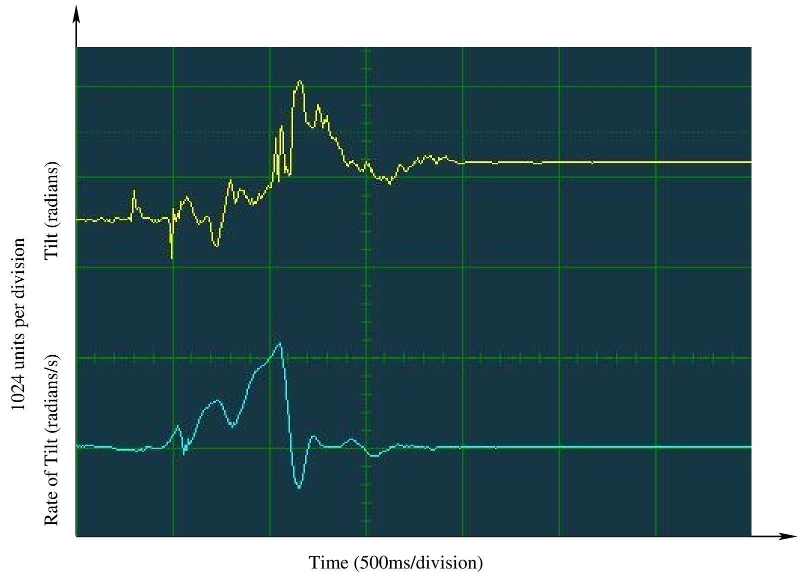


Figure 4.23. Screen capture of tilt and rate of tilt vs. time during brace. Gain scheduling for arm tilt position (Kp 2000, Ki 0, Kd 300) controller, and elbow position (Kp 30, Ki 0, Kd 0) controllers was used. Wheel gains were set to zero.

larger impact, the reflected inertia of the motor through the gearboxes provided a significant source of damping. If a small commanded motion with a minimal gain in the shoulder tilt joints were introduced prior to contact, the robot tended to absorb less at the moment of impact. However, specifying the moment to initiate the motion was difficult as it was desirable to start as close to the impact as possible. Otherwise, the arms would drastically retract and jeopardize the success of the absorb phase. Since the tilt data was too noisy to accurately estimate the moment of impact, this strategy was not implemented.

Another strategy that worked to guarantee a longer absorption phase was to create a longer impact. This was accomplished by modifying the reference tilt angle during the bilateral extension phase. The effect was to allow the robot to fall further, increasing the momentum to the point of being able to overcome the gearbox friction. This was the utilized strategy. The final value of the reference tilt angle was chosen to be 1.95 radians although other values proved successful as well. A trade-off was managed so as to allow the robot enough time to extend the arms, yet fall far enough to experience a significant enough impact to make distinctive landmark events.

Detecting contact with the ground is required by the brace controller to trigger the dissipative phase. This transition is signaled by large velocities and/or significant spikes in measured forces. Figure 4.24 shows the rate of tilt, and motor current in the shoulder and elbow. The significant deceleration in the tilt angle and the elbow current both provide distinctive events when the end effectors impact the ground while shoulder current is somewhat ambiguous. The elbow current was chosen and initial trials showed that a threshold filter was sufficient to indicate the impact event had occurred.

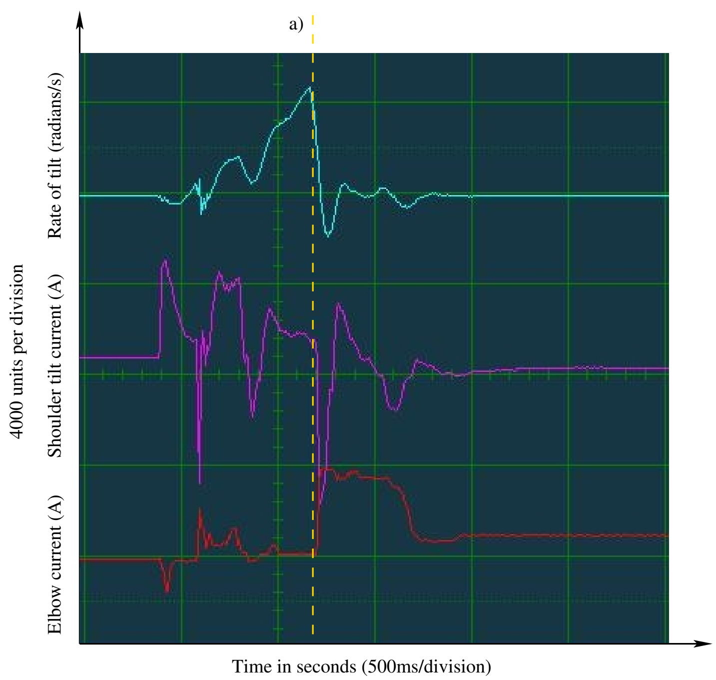


Figure 4.24. Rate of tilt, elbow current, and shoulder tilt current vs. time a) when the hands reach the ground during a bracing maneuver.

4.3 Discussion

The design of the uBot-5 incorporates native code for low-level motor control, Cartesian impedance control, and balancing. This approach organizes the behavior of the dexterous system while mitigating complexity and supporting efficient application programming. uBot-5's motor units are implemented in the fabric of the FPGA and represent the lowest-level of control. Impedance controllers use motor units to implement anisotropic impedance relations and provide useful modes of disturbance rejection, energy dissipation, and protect the platform's electromechanical components and the environment. The native control structure is designed organize the behavior of the dexterous system and it is intended to bootstrap the development of higher-level controls by users and by automated learning techniques. Figure 4.25 summarizes the distribution of the control over associated compute layers.

The uBot-5 can combine resources normally reserved exclusively for mobility or manipulation and is able to configure several hybrid modes as conditions require. The platform incorporates mechanical redundancy and a hierarchical compute architecture for combining control actions with multiple objectives and constraints. Table 4.1 summarizes the objective functions that form the basis of the policies developed in this chapter.

This chapter develops a family of native control policies that can be used to regulate postural stability (Figure 4.26). It describes controlled transitions from prone to 4-point and 3-point stances and ultimately up to 2-point (balancing) stances. To brace for a fall, uBot-5 reuses these controllers with appropriate impedance specifications to absorb the impact of a fall and restore the platform to a valid postural mode. This sequence exploits the flexibility of the whole body configuration, reduces complexity for programmers, and supports viability in variable terrain contexts.

The hierarchical postural stability framework also enhances the development of mobile manipulation behavior and organizes the complex robot into lower dimensional behavioral state spaces. The result is that the integrated hardware and software co-design reflects support for future applications, both scientific and technological in realistic and unstructured settings. In the following "Applications" chapter, I will

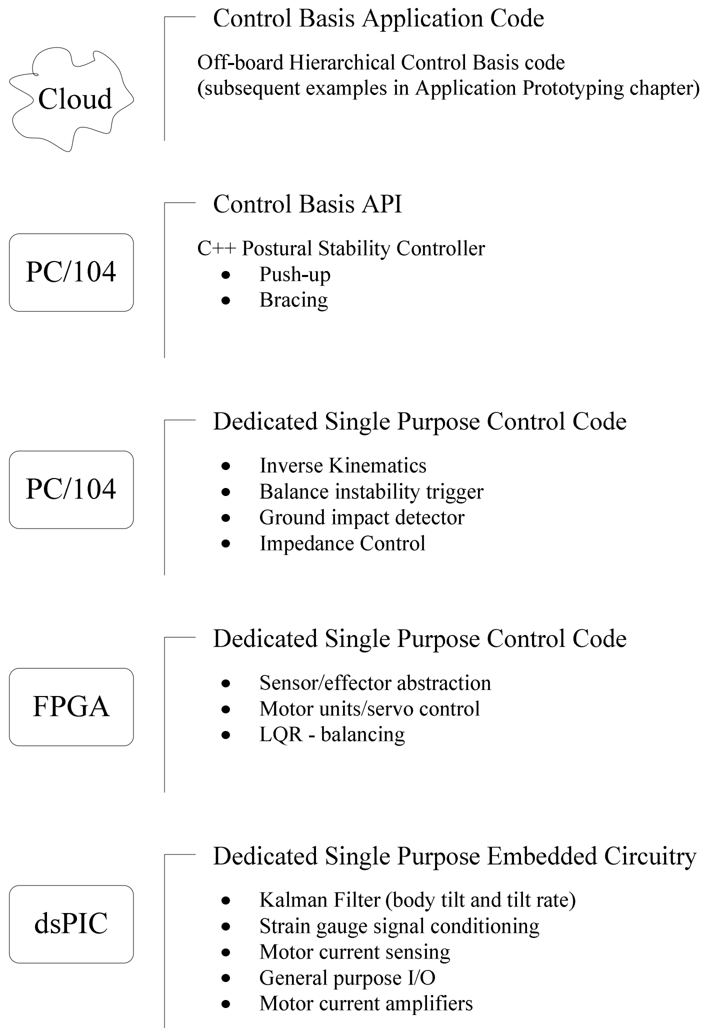


Figure 4.25. Diagram showing the compute architecture hierarchy and the organization of software development.

present some examples of sequential pushing behavior and learning performance in knuckle walking translation and rotation tasks dexterous mobility tasks.

Table 4.1. List of postural stability objective functions.

$\Phi_{kincond\_prone}$:	While prone, positions an arm or arms such that arms are in the initial push-up configuration.
$\Phi_{pushup\_setdown}$:	Performs a "push-up" to transition the robot from prone positions to 3 or 4-point stance to balancing and vice versa.
Φ_{ZMP} :	Orient the center of mass over the zero moment point (location where sum of all moments results in zero moment, a pure reaction force).
Φ_{LQR} :	Balancing algorithm for stable mobility on two wheels.
Φ_{brace} :	Bracing reflex is superior to all other control applications, it requires two arms to catch the robot during a fall.
$\Phi_{kincond\_standing}$:	Positions an arm or arms in an intermediate posture of the 4-point stance that is simultaneously stabilizable by LQR balancing controller.

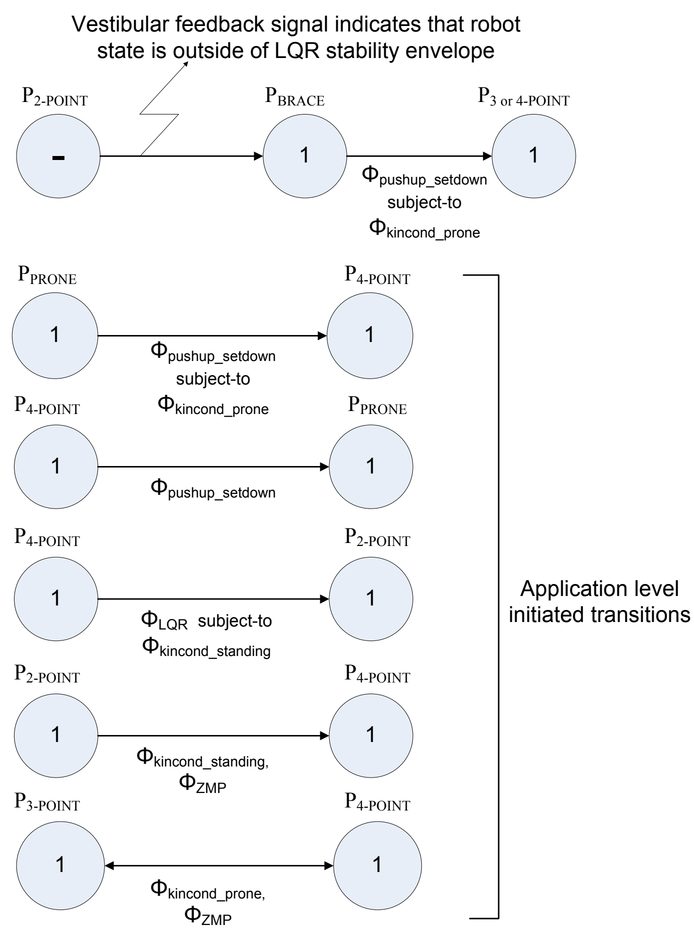


Figure 4.26. Postural Stability Control Suite set of FSA provides mobility modes with varying effector utilizations while meeting distinct performance objectives.

CHAPTER 5

APPLICATION PROTOTYPING

5.1 Whole Body Pushing Tasks

Active posture control can improve performance and stability. Leaning into a heavy door is energetically efficient and higher performance than strategies exclusively focused in the arm. To take full advantage of a small footprint, a controllable joint near the ground serves as another degree-of-freedom for tilting the body. The human ankle joint performs this role in order to posture the body and to redistribute mass for reaching out into the world, for increasing leverage, or for traversing inclined terrain.

Thus far, little research has explored the potential for whole-body policies and dynamic posture control– a potential that is very significant. Postural control is considered mostly for humanoid legged robots that must preserve stability. Khatib presents a framework for performing tasks while maintaining postural constraints for a humanoid robot [33]. In this work the emphasis is on logically decoupling tasks from postural control rather than exploiting postural control to increase performance on the task. Takubo and Harada consider the control of a humanoid's posture during pushing tasks to ensure platform stability [57, 56, 20]. Self-motion creates challenges related to precision that could undermine open-loop grasping and manipulation techniques, thus an active control system for corrective movements will be required to establish and to maintain grasps. The technique incorporates measures of hand reaction force to augment the center of mass positional control and to define the ZMP trajectory for maintaining momentum. Yoshida presents a method that optimizes the static posture of a humanoid robot with respect to an evaluation function to incorporate factors including force generation and the ability to reject disturbance forces [61].

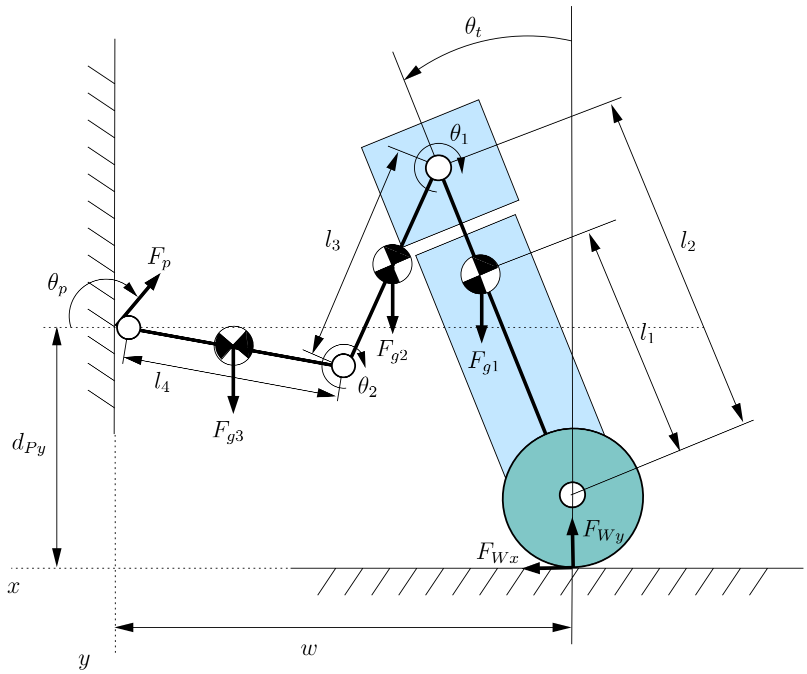


Figure 5.1. Free body diagram of the uBot-5 in a pushing task.

Upper body dexterity adds mobility modes (of varying effectiveness) for bracing against environmental surfaces. This provides the opportunity to use upper-body resources in postural control tasks and balance. One example of such as task is the capacity to push (create interaction forces). uBot-5 was designed to make it possible to exploit mass and inertial forces when interacting with the environment. As a result, the uBot-5 is able to generate significantly greater pushing forces using postural control, than statically-stable platforms with the same form factor and footprint [58].

Static analysis of pushing forces and stability was performed using the uBot-3 generation of the platform. Like uBot-5, uBot-3 can control the angle of its body with respect to vertical. This is the key to improving the capacity of a mobile manipulator to exert force on the environment.

A free body diagram of the posture controlled platform in a pushing configuration is showing in Figure 5.1. For purposes of comparison, we assume that the statically stable version of the robot has a wheel in front and back, as in Figure 5.2, to provide the maximum possible stability given the width of the base. We assume that there is no difference in the robots total mass or its distribution due to the difference in number or arrangement of wheels. F_{g1}, F_{g2}, and F_{g3} are the forces due to gravity on the body of the robot, the first link of the arm, and the second link of the arm respectively. F_{Wx} and F_{Wy} are the x and y components of the force exerted by the ground on the wheels. F_p is the force exerted on the end effector by the environment. We consider all possible values for θ_p such that $0 \leq \theta_p \leq 2\pi$.

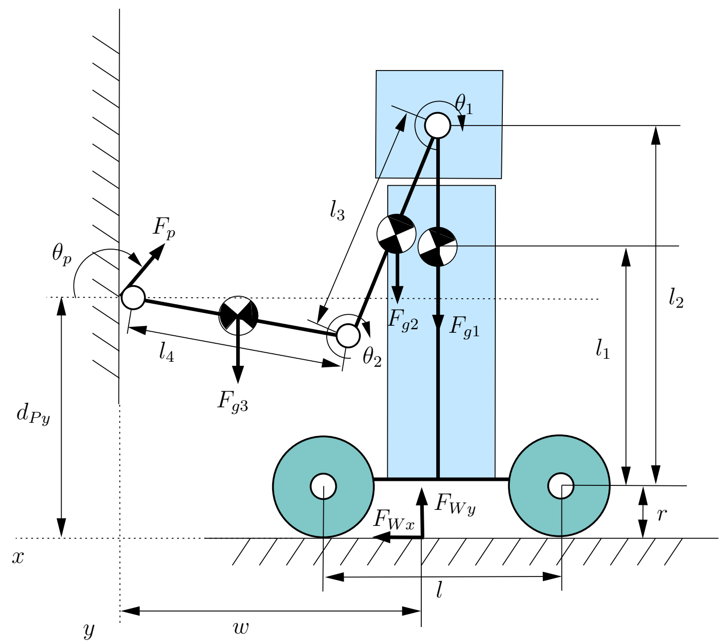


Figure 5.2. Free body diagram of the platform with limited postural control in a pushing task.

The maximum force magnitude of F_p in a given direction, θ_p, is computed by determining forces at the end effector that do not violate static stability. The analysis assumes a static coefficient of friction between the wheels and the ground of $\mu = 0.5$. Therefore, the model presumes that wheel slip occurs when,

$$|F_{Wx}| > \mu F_{Wy} \tag{5.1}$$

For some end effector forces and positions, static equilibrium is maintained for all values of F_p. In these situations, the maximum pushing forces at the endpoint is limited by joint torque limits.

Using Figure 5.1, we can find all of the moments that act around the wheel given the following values:

- w: The x-coordinate of the wheel;
- c: The x-coordinate of the center of mass of the robot body;
- s_x, s_y: The x, and y-coordinates of the robots shoulder (the first joint of the arm).

Also, assume the height at which the end effector makes contact with object, dP_y, is given. For static equilibrium, the net moment around the wheel must be zero.

$$\begin{aligned} m_1 g(w-c) + m_2 g(w - 0.5(s_x + d_3\cos(\theta_1))) \\ +m_3 g(w - ((s_x + d_3\cos(\theta_1) + 0.5 d_4 \cos(\theta_1+\theta_2)))) \\ -d_{Py} F_p \cos(\theta_p) - w F_p \sin(\theta_p) &= 0, \end{aligned} \tag{5.2}$$

where g is the acceleration of gravity, $m_1 g(w-c)$, $m_2 g(w - 0.5(s_x + d_3\cos(\theta_1)))$, and $m_3 g(w - ((s_x + d_3\cos(\theta_1) + 0.5 d_4 \cos(\theta_1+\theta_2))))$ are the moments about the wheel due to the mass of the body, first link of the arm, and second link of the arm, respectively. The load at the end effector generates moments $-d_{Py} F_p \cos(\theta_p)$, and $-w F_p \sin(\theta_p) = 0$ about the wheel.

Solving for F_p yields:

$$\begin{aligned} F_p &= (g(m_1(w-c) + m_2(w - 0.5(S_x + d_3 cos(\theta_1))) \\ &\quad +m_3(w - (S_x + d_3\cos(\theta_1) + 0.5 d_4\cos(\theta_1+\theta_2))))) \\ &\quad /(d_{Py}\cos(p) + w\sin(\theta_p)). \end{aligned} \tag{5.3}$$

Values of w, c, s_x, and s_y are computed for each type of robot. These also serve as the free variables in a pushing controller for achieving a desired F_p.

Since the robot is in static equilibrium, $|F_{Wx}| = F_p \cos(\theta_p)|$, and $|F_{Wy}| = m_t g - F_p \sin(\theta_p)$, where m_t is the total mass of the robot body and both links of the arms. Using Equation 5.1, we find that the maximum value of F_p that does not violate static friction constraints on the ground is:

$$F_p = \frac{\mu m_t g}{|\cos(\theta_p)| + \mu \sin(\theta_p)}. \tag{5.4}$$

If $|\cos(\theta_p)| + \mu \sin(\theta_p) \leq 0$, then static friction is not violated for any value of F_p.

5.1.1 Tipping Conditions for Statically Stable Platform

The tipping condition is determined using the definition of the zero moment point (ZMP). In static equilibrium w, the ZMP is the location of the ground reaction force, such that there is no net moment on the robot. For any force on the end effector, the platform will not fall over if it can position itself such that the ZMP is between the front and back wheels.

For the statically stable platform, w will always be the position of the ZMP. Let $\underline{w}$ be the minimum possible position along the x-axis at which the rightmost wheel can be positioned given the endpoint position of its arm (note that the origin is to the left of the base). Let $\overline{w}$ be the maximum possible position along the x-axis at which the left wheel can be positioned.

If the line of action of F_p at θ_p intersects the ground between $\underline{w}$ and $\overline{w}$, the platform will never tip over, since as F_p increases, the ZMP will tend toward the location where the line of action of F_p intersects the ground. Since the platform can place its support region over this point, it will not tip over no matter how large F_p becomes.

If the line of action of F_p intersects the ground to the left of the potential support region, then the maximum value of F_p that will not cause the platform to tip over is the value that causes the ZMP to be at the left edge of the potential support region. We can find this value of F_p using Equation 5.3 with:

$$w = \underline{w} \tag{5.5}$$

$$c = \underline{w} + l/2 \tag{5.6}$$

Similarly, if the line of action of F_p projects to the right of the possible support region, we can find the maximum possible value for F_p using Equation 5.3 with:

$$w = \overline{w} \tag{5.7}$$

$$c = \overline{w} - l/2 \tag{5.8}$$

In all of these situations we set s_x and s_y as follows:

$$s_x = c \tag{5.9}$$

$$s_y = d_2 + r - d_{Py} \tag{5.10}$$

5.1.2 Tipping Conditions for Dynamically Stable Platform

The dynamically stable platform has an extra degree of freedom θ_t, the angle between the body of the robot and the ground, as shown in Figure 5.1. In order for the dynamically stable platform to be in static equilibrium, the moment around the wheels must be zero. For a given value of θ_p, we can find the maximum value of F_p by solving for all of the terms in Equation 5.3 in terms of w and θ_t, and then numerically searching over the feasible combinations of w and θ_t, choosing the combination that maximizes F_p. For the dynamically stable platform:

$$\underline{w} = -d_2 \cos(\theta_t) - \sqrt{(d_3 + d_4)^2 - (d_2 \sin(\theta_t) - d_{Py} + r)^2}, \tag{5.11}$$

and

$$\overline{w} = -(\underline{d_{Px}} + l/2) \tag{5.12}$$

Due to physical and control constraints on the angle between the platform and the ground, we assume $\pi/4 \leq \theta_t \leq 3\pi/4$. For given values of w and d_{Py}, θ_t must also obey the following constraints which ensure that the body of the platform is not too close to the end effector and that the point where the end effector is to make contact with the manipulated object is within the workspace of the manipulator:

$$\begin{aligned} \theta_t \quad \leq \quad & \cos^{-1}\left(\frac{d_2^2 + w^2 + (d_{Py} - r)^2 - (d_3 + d_4)^2}{2d_2\sqrt{w^2 + (d_{Py} - r)^2}}\right) & (5.13) \\ & + \sin^{-1}\left(\frac{d_{Py} - r}{\sqrt{w^2 + (d_{Py} - r)^2}}\right) & (5.14) \end{aligned}$$

and

$$\theta_t \geq \tan^{-1}\left(\frac{l/2}{h + r}\right) + \cos^{-1}\left(\frac{w - \underline{d_{Px}}}{\sqrt{(h + r)^2 + (l/2)^2}}\right) \tag{5.15}$$

We can now solve for the variables in Equation 5.3:

$$c = w + d_1 \cos(\theta_t), \tag{5.16}$$

$$s_x = w + d_2 \cos(\theta_t), \tag{5.17}$$

and

$$s_y = r + d_2 \sin(\theta_t) - d_{Py} \tag{5.18}$$

Using Equations 5.3 and 5.11–5.18 , it is straight forward to compute the maximum value of F_p that can be resisted by the platform in static equilibrium for a given θ_p.

Figure 5.3 shows the predicted force advantage of using whole body postural control to generate force at the end effectors of the uBot-4 for two different postures

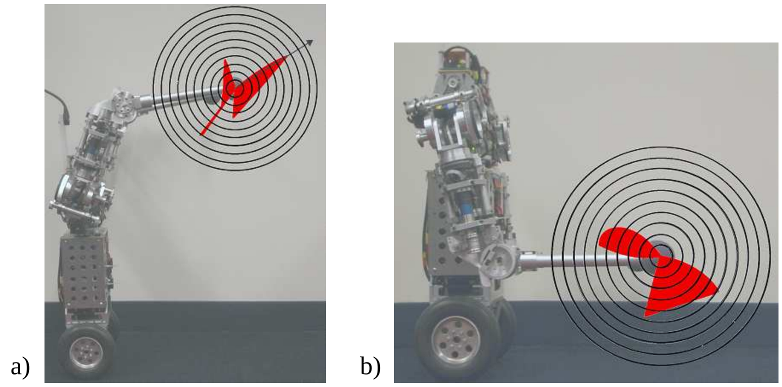
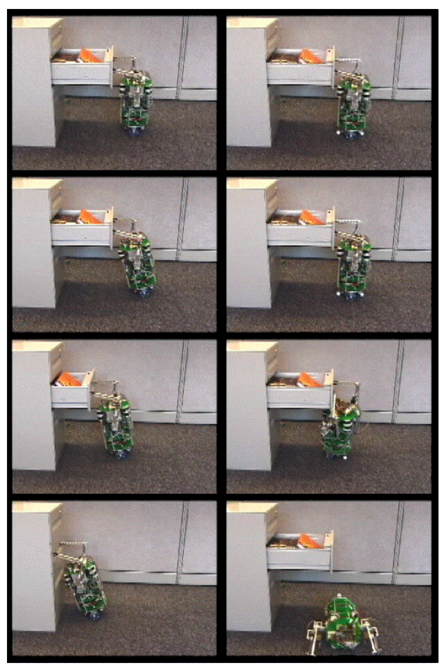


Figure 5.3. Panel a) illustrates the increase in forces that can be applied to the environment using whole body postural control for $d_{Py} = 0.75$m and $F_p = 50$N. The whole body platform can apply ≈ 15N more in the direction indicated by the arrow than its statically stable equivalent. Panel b) shows the results for $d_P = 0.19$m and maximum $F_p = 50$N. Each concentric circle corresponds to a $2N$ increment.



uBot–3 vs. uBot–2

Figure 5.4. uBot-3 illustrating a pushing task high overhead. The dynamically stable platform (left) and statically stable platform (right) pushing a drawer filled with books.

of the robot. This advantage is as great as 35N in some directions and for some configurations of the uBot-4.

Figure 5.4 demonstrates the advantage a dynamically stable platform with postural control has over a statically stable robot with the same mass and footprint in a drawer pushing tasks. The dynamically stable robot successfully leverages its high center of mass to close the same drawer without falling down.

5.2 Knuckle Walking

Kuindersma's [35] experiment regarding knuckle walking is a good example of an application that was facilitated by the morphological and computational structure designed into the uBot concept. Although the uBot-5 has wheels instead of legs, a hybrid knuckle walking strategy using both wheels and arms is a natural mode of locomotion for the robot in the 3-point stance. This is due to the large reachable ground plane. This skill is defined by constraints that require at least one stable 3-point stance to be maintained always using appropriate control tasks.

To optimize a 3-point stance, a controller must position the center of gravity of the robot near the ZMP of the support polygon. Kuindersma formulated knuckle walking as a sequence of control tasks (including 3- to 4- and 4- to 3-point transitions and impedance controlled motor units) with a subordinate chassis drive to control and actuate the wheels.

Two behaviors that are particularly relevant to mobility and locomotion are the ability to translate and rotate the robot. The development of translate and rotate gaits can employ Q-learning in the abstract predicate space [28, 46, 35]. After the translate and rotate controllers are developed, they can be incorporated as abstract actions for subsequent programming tasks.

The development of behaviors is expected to simplify future programming tasks because programmers (humans and those derived from an automated learning technique) no longer have to reason exclusively in high-dimensional configuration spaces. The navigation policy for knuckle walking shows how the translate and rotate behavior significantly reduces the dimensionality of the state space by carving it up in a relevant manner. The resulting policy over the abstract state space is defined abstract skills: one that engages sequential limb movements to change the heading of the platform and another to produce translations in the (x,y) plane as illustrated in Figure 5.5, where the state is $(\mathbf{p}_{rot}, \mathbf{p}_{trans})$.

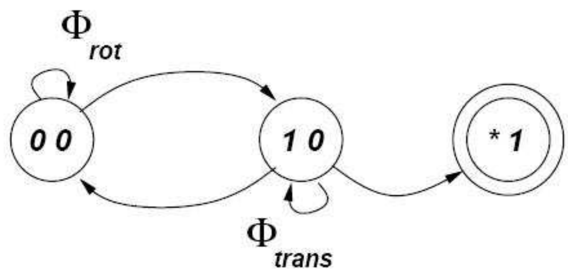


Figure 5.5. Walking policy for activating rotate gait whenever the robot is not aligned with reference direction. Otherwise forward walking motion is selected.

Knuckle walking is defined by alternating 3-point stances that result in translation or rotation across the floor (at first ignoring postures that allow arms to be braced against walls). The constraint that expresses postural stability only requires at least one arm to be in a stable 3-point stance.

Kuindersma's learning experiment used single-step tabular Q-learning [11] and represented the state of the system as vector of "n" Boolean predicates, $\mathbf{p}$, with the $\mathbf{p}_i$ element asserting 1 if controller i is converged, and 0 otherwise (see Equation 4.21). The state description was derived from the set of control actions listed in Table 5.1, where the state is described by an 11-dimensional state representation.

The total number of possible states given the 11-variable representation was $2^{11} = 2048$. To eliminate many of these states, logical constraints on the action available to the robot were imposed. Those included only considering concurrent actions that included up to three controllers, preventing concurrent actions that were contradictory, and defining a stability criterion. The stability criterion stated that

Table 5.1. Primitive control actions available to the robot in the knuckle walking task.

Label	Controller Description
1	Stabilize 4-point stance
2	Stabilize left 3-point stance
3	Stabilize right 3-point stance
4	Forward
5	Reverse
6	Rotate clockwise
7	Rotate counterclockwise
8	Raise left endpoint
9	Raise right endpoint
10	Place left endpoint
11	Place right endpoint

at least one stability controller must be converged at all times. For example, when in a 4-point posture and executing the "forward" action, the robot will roll forward until its COM projection strays too far from the ZMP, at which time the forward controller "quiesces."

Assuming the robot begins in a stable 4-point stance, there were 25 possible actions and a total of 144 reachable states that satisfied the postural stability constraints [35]. Kuindersma's results demonstrate that this approach supports autonomous learning techniques that learn knuckle walking gaits efficiently using this state and action definition. Policies in this space are sequences of primitive control actions and controllers from the postural stability control suite.

5.3 Personal Robotics and Healthcare

The success of the uBot platform is measured in part in terms of its ability to catalyze new research about dexterity and control in whole-body mobile manipulators. This contribution relates to new applications and functionality for robots. A personal assistant robot for eldercare applications is one such emerging role for dexterous WB-MMs. As demographic bubbles threaten to overburden the healthcare system, one solution may be to develop a robot that can allow a patient to remain independent longer and increase their quality of life by supporting telemedicine providers. In the manufacturing domain, a dexterous WBMM may decrease integration and application development costs of widespread flexible automation and supply chain handling in medium to small businesses. A typical industrial robot requires custom single purpose programming, safety cages, detailed material flow planning and synchronization. Moreover, some tasks require highly specialized sensors that must have controlled environments in order to perform reliably. Thus, there is a need for affordable, easily programmable, intelligent, and safe robots capable of collaborating with human workers to increase efficiency, reduce repetitive stress injuries, and lower costs. Hotzone and HAZMAT disaster relief and other-world exploration are particularly dangerous environments for humans. However, the risks can be mitigated by robots that can go where people go, deploy networked sensors, and perform manipulation tasks. A dexterous WBMM may also enable first responders to remotely enter dangerous areas, increase situational awareness, lead victims to safe areas, perform triage operations and assist those in need.

A personal robot offers the potential to further the productivity and quality of life in society. As baby-boomers approach the age of 65 approximately 70 million new clients will enter retirement. Shortly after that, many of these clients will require eldercare services without which, individuals in this population will likely require assisted living facilities or nursing care. In order to mitigate rising costs reduce the burden on centralized medical providers and community services, a robot that can perform assistance in various human environments may provide a solution. This type of robot can help an elder person stay at home longer, assist with chores, provide a link to medical professionals and family members, and greatly improve their standard of living.

A family-client interface was implemented for a configuration of the uBot that introduced a touchscreen, a shoulder mounted camera, and an audio interface including speakers and a microphone. Video conferencing via standard Internet telephony enables users to project their face and speak through the robot. Moreover, the user can act through the robot as a remotely embodied avatar; creating a cognitive focal point for the client, sensing the local environment, and performing work. This enriches the social experience of the client by allowing remote family members to visit a loved one and communicate naturally, help with chores, and be physically present in the client-side environment.

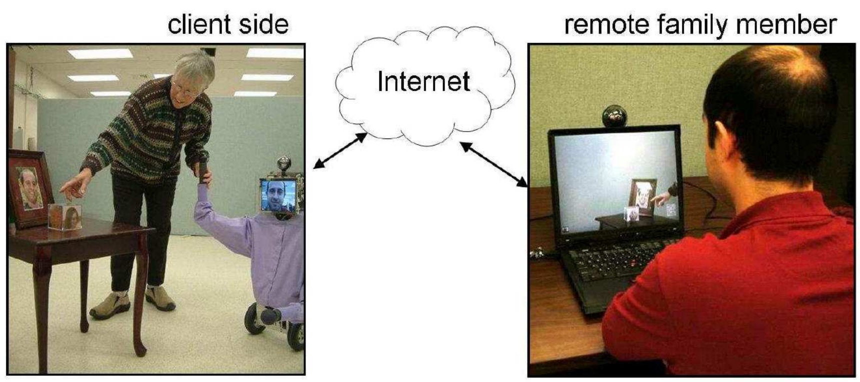


Figure 5.6. A client gives a tour of the assisted care facility. The visitor and the host share a videophone conversation while interacting and moving about the facility.

Applications that can monitor the well-being of the patient, assess their health, remind them to take their medications, and perform routine chores have been prototyped using the uBot-4 and uBot-5 [12]. In Figure 5.6 uBot-4 demonstrates the capabilities of the Skype videoconferencing interface and illustrates a doctor could conduct a virtual house call on a patient in his or her home. In this configuration, an on-board webcam provides the remote teleoperator with a perspective on the client-side environment that includes the robot's bimanual workspace. A joystick interface allows the operator to translate and to rotate the base, position the joints in the upper body, and trigger sequential behaviors such as postural transitions and grasping motions.

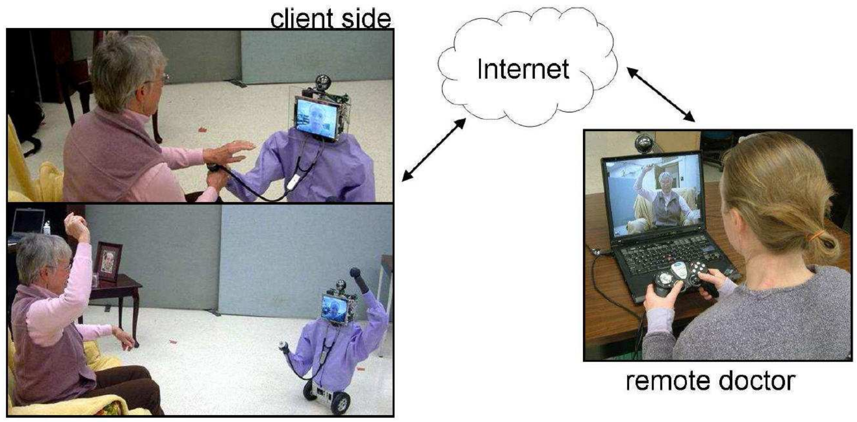


Figure 5.7. A three-question stroke diagnosis through video phone with motor tasks that are demonstrated by the uBot.

In addition, a mobile manipulator could deploy sensors and record telemetry in precise locations in the environment. To investigate these applications, the uBot-5 has acted as a client to a sensor array consisting of Pan-Tilt-Zoom (PTZ) cameras. The large scale network provides localization and path planning support, enabling the uBot-5 to act as the physical embodiment for network services that can do work. When the network detects unexpected situations, the uBot can be automatically deployed to intervene.

The ability to relay communication and deploy sensors as Internet services also enables new possibilities for remote service providers as well. Remote telepresence efficiently addresses the upcoming shortage of caregivers, and saves elders from difficult and time consuming transportation. Such technology may also improve workforce fluidity in this challenging demographic. Consequently, personal robots may provide some capabilities that mitigates the excess demand for more doctors, nurses, and technicians. Addressing eldercare needs requires developing solutions that can provide short term support for high demand situations. Moreover, these personal robots can be re-purposed after the demand passes.

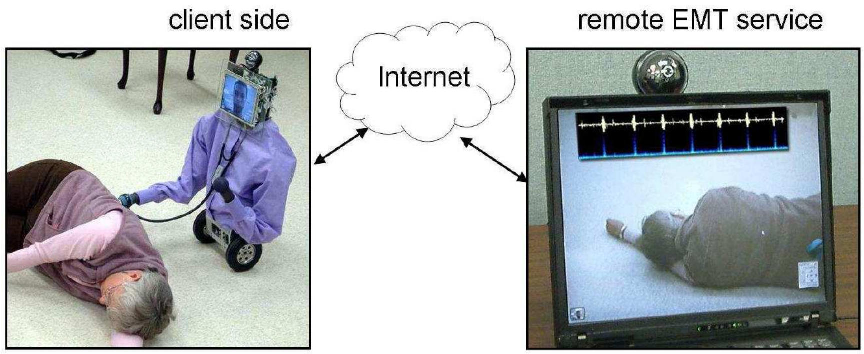


Figure 5.8. A fall is detected, the system places a call to Emergency Medical Technicians (EMT). Using the uBot, the EMT can attempt to rouse the client, or in this case, apply a digital stethoscope. The digital stethoscope relays heart rate and respiration telemetry to responders.

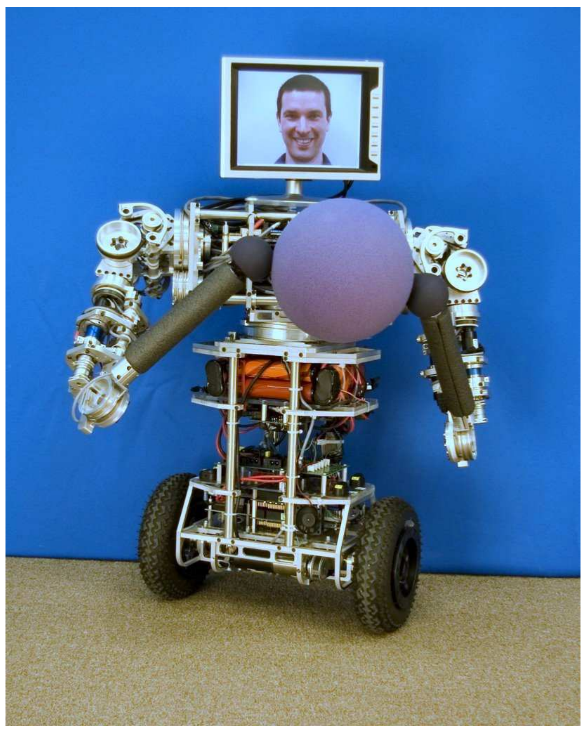


Figure 5.9. uBot-5 ready for work.

When the uBot-5 is utilized as a personal assistant type device, the robot is controlled to pickup, process, and replace all sorts of objects including soiled garments, toys, and tableware. Figure 5.9 shows the platform reaching for a ball placed on the ground. The body and mobility design enables the robot to manage tight corridors and shared spaces with humans such as kitchens and hallways. The robot's interfaces for teleoperation and video conferencing allow people to interact though the robot and operate the hardware remotely. Ultimately, this robot should be able to be deployed in a unique home environment, not require extensive custom programming, safely navigate autonomously, and perform useful work.

5.4 Mobile Dexterous Social

In a collaboration with the MIT Media Lab, the UMASS uBot was extended to incorporate a socially expressive head and force sensing hands to create the Mobile, Dexterous, and Social (MDS) platform. The MDS head consists of 20+ degree-of-freedom to generate facial expressions and the 5-DOF hands are used to create gestural actions and manipulate small objects. Figure 5.10, shows a solid model visualization of the integrated concept.

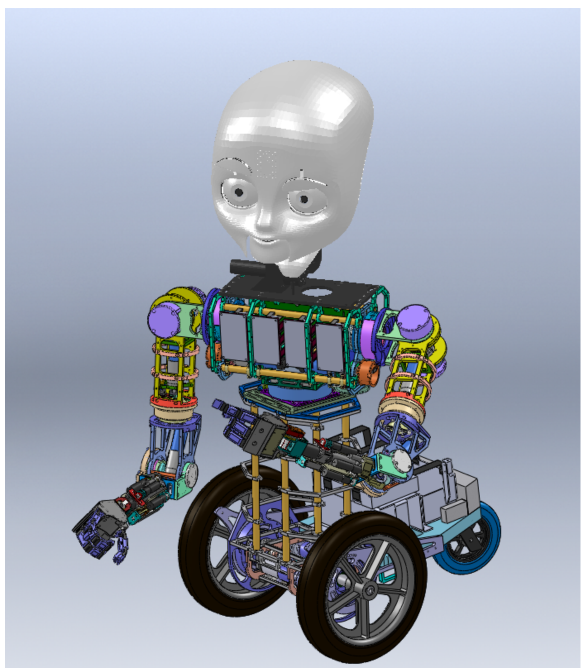


Figure 5.10. Mobile Social Dexterous (MDS) CAD rendering based on the uBot platform.

The uBot platform was chosen for this project based on the ability to support aspects of human-robot interaction critical to the MDS and emergency response applications. The uBot platform presents a familiar human-like morphology and supports gestural communication with dexterous bimanual manipulators.

5.5 Summary and Conclusions

This thesis deals with issues central to extending our understanding and implementation of dexterity in robot mobility and manipulation. The approach chosen in this study centers around a hardware and software co-development, resulting in the construction of a dynamically stable, bimanual, whole body mobile manipulator, the uBot-5. The platform successfully pairs motor flexibility and performance with a hierarchical embedded control framework that insulates programmers from the nitty-gritty details of sophisticated machines and provides a convenient applications development environment for constructing dexterous machines.

Historically, robotics hardware and control have suppressed dynamic effects when they can cause the robot to fall over. Conversely, biological systems have evolved to mitigate this risk behaviorally in order to exploit the performance advantages that dynamics provides. This interplay of mobility and dexterity requires a relatively well-developed vestibular system, balance control, and knowledge of the mechanical properties of objects in the environment. This work demonstrates a mastery of mobility and dexterity through several sequential programs including bracing to catch a fall, standing up from a prone position, and knuckle walking. Furthermore, these behaviors illustrate that reflexive behavior can increase the efficiency of the construction of control programs and improve the robustness of these programs to unanticipated circumstances.

The postural stability control suite supports dexterous manipulation whereby certain configurations, from the set of postural modes including lying prone, knuckle walking, and balancing, present advantages in particular contexts. While prone, the robot can reach objects low to the ground or under obstructions such as tables and bed frames. Knuckle walking provides mobility in the presence of challenging or slippery terrain. When stationary, only one arm is required to support the robot and thus single-handed manipulation tasks can still be achieved. The balancing mode completely frees the arms and further enhances the dexterity of the robot. Transitions between the modes include push-up, set-down, and bracing for a fall. This progression was inspired by developmental pressures that seek to increase efficiency, dexterity, and safety. Moreover, postural stability extends the range of stable mobility configurations and permits secondary manipulation objectives through a wide range of operating conditions.

The uBot-5 platform extends the state of the art in mobile manipulation and presents many new opportunities for future robotics research.

BIBLIOGRAPHY

[1] Barrett Technology - Advanced Robotic Arm and Hand Systems - WAM Arm and BarrettHand. `http://www.barretttechnology.com/robot/`, December 2006.

[2] HITACHI : Mechanical Engineering Research Laboratory : Hitachi's robot EMIEW. `http://www.hqrd.hitachi.co.jp/merle/emiew.cfm`, December 2006.

[3] robo3. `http://www.robo3.com/robot/robot\_r3.html`, December 2006.

[4] RobotCub - An Open Framework for Research in Embodied Cognition. `http://www.robotcub.org/`, 2006.

[5] TOYOTA.CO.JP -Toyota Partner Robot-. `http://www.toyota.co.jp/en/special/robot/`, December 2006.

[6] VECNA Technologies : Robotics. `http://www.vecnarobotics.com/`, December 2006.

[7] Ambrose, Robert O., Savely, Robert T., Goza, S Micael, Strawser, Philip, Diftler, Myron A., Spain, Ivan, and Radford, Nicolaus. Mobile Manipulation using NASA's Robonaut. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation* (New Orleans, LA, April 2004), pp. 2104–2109.

[8] Asfour, Tamim, Berns, Karsten, and Dillmann, Rüdiger. The Humanoid Robot ARMAR: Design and Control. In *The first IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS)* (2000).

[9] Braitenberg, V. *Vehicles: Experiments in Synthetic Psychology.* MIT Press, 1984.

[10] Brooks, Rodney, Aryananda, Lijin, Edsinger, Aaron, Fitzpatrick, Paul, Kemp, Charles, O'Reilly, Una-May, Torres-Jara, Eduardo, Varshavskaya, Paulina, and Webber, Jeff. Sensing and Manipulating Build-For-Human Environments. *International Journal of Humanoid Robotics 1*, 1 (March 2004), 1–28.

[11] C., Watkins. *Learning from Delayed Rewards.* PhD thesis, University of Cambridge, England, May 1989.

[12] Deegan, Patrick, Grupen, Roderic, Hanson, Allen, Horrell, Emily, Ou, Shichao, Riseman, Edward, Sen, Shiraj, Thibodeau, Bryan, Williams, Adam, and Xie, Dan. Mobile Manipulators for Assisted Living in Residential Settings. *Journal of Autonomous Robotics, Special Issue on Socially Assistive Robotics 24*, 2 (February 2008).

[13] Diftler, M. A., Ambrose, R. O., Goza, S. M., Tyree, K. S., and Huber, E. L. Robonaut Mobile Autonomy: Initial Experiments. In *IEEE International Conference on Robotics and Automation* (International Conference on Robotics and Automation Barcelona, Spain, April 2005).

[14] Edsinger-Gonzales, Aaron, and Weber, Jeff. Domo: A Force Sensing Humanoid Robot for Manipulation Research. *International Journal of Humanoid Robotics 1* (2004), 273–291.

[15] Garcia, Luiz-Marcos, Oliveira, Antonio A. F., Grupen, Roderic A., Wheeler, David S., and Fagg, Andrew H. Tracing Patterns and Attention: Humanoid Robot Cognition. *IEEE Intelligent Systems 15*, 4 (2000), 70–77.

[16] Goza, S. M., Ambrose, R. O., Diftler, M. A., and Spain, I. M. Telepresence Control of the NASA/DARPA Robonaut on a Mobility Platform. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems* (2004).

[17] Grasser, Felix, D'Arrigo, Aldo, Colombi, Silvio, and Rufer, Alfred C. JOE: A Mobile, Inverted Pendulum. *IEEE Transactions on Industrial Electronics 49*, 1 (February 2002), 107–114.

[18] Han, J.D., Zeng, S.Q., Tham, K.Y., Badgero, M., and Weng, J.Y. Dav: A Humanoid Robot Platform for Autonomous Mental Development. In *IEEE International Conference of Development and Learning* (2002).

[19] Hannigan, Edward P. Endpoint Force Sensing for Mobility and Grasping. Master's thesis, University of Massachusetts Amherst, 2008.

[20] Harada, Kensuke, Kajita, Shuuji, Kaneko, Kenji, and Hirukawa, Hirohisa. Pushing Manipulation by Humanoid considering Two-Kinds of ZMPs. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation* (Taipei, Taiwan, September 2003), pp. 1627–1632.

[21] Hart, S., and Grupen, R. Natural Task Decomposition with Intrinisic Potential Fields. In *International Conference on Robotics and Systems (IROS)* (San Diego, California, October-November 2007).

[22] Hart, S., Sen, S., Ou, S., and Grupen, R. The Control Basis API - A Layered Software Architecture for Autonomous Robot Learning. In *Workshop on Software Development and Integration in Robotics, IEEE Conference on Robotics and Automation (ICRA)* (Kobe, Japan, May 2009).

[23] Hirai, Kazuo, Hirose, Masato, Haikawa, Yuji, and Takenaka, Toru. The Development of Honda Humanoid Robot. In *International Conference on Robotics and Automation* (Leuven, belgium, May 1998).

[24] Hogan, Neville. Impedance Control: An Approach to Manipulation. Part I - Theory Part II- Implementation, Part III- Applications. *J. Dynamic Systems, Measurement and Control, March 1985 107* (1985), 1–24.

[25] Holmberg, Robert, and Khatib, Oussama. Development and Control of a Holonomic Mobile Robot for Mobile Manipulation Tasks. *The International Journal of Robotics Research 19*, 11 (2000).

[26] Huber, M., and Grupen, R. A. Learning to Coordinate Controllers Reinforcement Learning on a Control Basis. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI-97)* (San Francisco, Aug. 23–29 1997), Morgan Kaufmann Publishers, pp. 1366–1371.

[27] Huber, M., MacDonald, W.S., and A., Grupen R. A Control Basis for Multilegged Walking. In *Proceedings of the International Conference on Robotics and Automation* (Minneapolis, MN, April 1996), vol. 4, pp. 2988–2993.

[28] Huber, Manfred. *A Hybrid Architecture for Adaptive Robot Control*. PhD thesis, University of Massachusetts Amherst, 2000.

[29] I., Connolly C., and A., Grupen R. Nonholonomic Path Planning Using Harmonic Functions. Tech. Rep. 94-50, UMass Computer Science, 1994.

[30] J., Coelho. *Multifingered Grasping: Grasp Reflexes and Control Context*. PhD thesis, University of Massachusetts Amherst, September 2001.

[31] Kaneko, Kenji, Kanehiro, Fumio, Kajita, Shuuji, Hirukawa, Hirohisa, Kawasaki, Toshikazu, Hirata, Masaru, Akachi, Kazuhiko, and Isozumi, Takakatsu. Humanoid Robot HRP-2. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation* (New Orleans, LA, April 2004), pp. 1083–1090.

[32] Katz, Dov, Horrell, Emily, Yang, Yuandong, Burns, Brendan, Buckley, Thomas, Grishkan, Anna, Zhylkovskyy, Volodymyr, Brock, Oliver, and Learned-Miller, Erik. The UMass Mobile Manipulator UMan: An Experimental Platform for Autonomous Mobile Manipulation. In *Robotics: Science and Systems - Workshop on Manipulation for Human Environments* (2006).

[33] Khatib, O., Sentis, L., Park, J., and Warren, J. Whole-Body Dynamic Behavior and Control of Human-Like Robots. *International Journal of Humanoid Robotics 1*, 1 (2004).

[34] Koditschek, D.E., and Rimon, E. Robot Navigation Functions on Manifolds with Boundary. *Advances in Applied Mathematics 11*, 4 (1990), 412–442.

[35] Kuindersma, Scott, Hannigan, Edward, Ruiken, Dirk, and Grupen, Roderic. Dexterous Mobility with the uBot-5 Mobile Manipulator. In *14th International Conference on Advanced Robotics (ICAR)* (Munich, Germany, June 2009).

[36] Kuroki, Yoshihiro, Fukushima, Tetsuharu, Nagasaka, Ken'ichiro, Moridaira, Tomohisa, Doi, Toshi T., and Yamaguchi, Jin'ichi. A Small Biped Entertainment Robot Exploring Human-Robot Interactive Applications. In *The 12th IEEE International Workshop on Robot and Human Interactive Communication* (2003), pp. 303–308.

[37] Latash, M. L., and Turvey, M. T., Eds. *Dexterity and Its Development.* Lawrence Erlbaum Associates, Mahwah, New Jersey, 1996, ch. On Dexterity and Its Development by N.A. Bernstein.

[38] Lauwers, Tom, Kantor, George, and Hollis, Ralph. One is Enough! In *12th International Symposium on Robotics Research* (San Francisco, October 2005).

[39] McGann, Conor, Berger, Eric, Bohren, Jonathan, Chitta, Sachin, Gerkey, Brian, Glaser, Stuart, Marthi, Bhaskara, Meeussen, Wim, Pratkanis, Tony, Marder-Eppstein, Eitan, and Wise, Melonee. Model-based, Hierarchical Control of a Mobile Manipulation Platform. In *ICAPS Workshop on Planning and Plan Execution for Real-World System* (Thessaloniki, Greece, 2009).

[40] Messner, Bill, and Tilbury, Dawn. Control Tutorials for Matlab: Modeling an Inverted Pendulum. `http://www.engin.umich.edu/group/ctm/examples/pend/invpen.html`, April 1997.

[41] Nakamura, Y., and Hanafusa, H. Inverse Kinematic Solutions With Singularity Robustness for Robot Manipulator Control. *Journal of Dynamic Systems, Measurement, and Control 108*, 33 (1986), 163–171. American Society of Mechanical Engineers.

[42] Nasir, Ahmad Nor Kasruddin, Ahmad, Mohd. Ashraf, and Rahma, Mohd Fua'ad. Performance Comparison Between LQR and PID Controller For An Inverted Pendulum System. In *International Conference of Power Control and Optimization* (Chiang Mai, Thailand, July 2008).

[43] Nguyen, Hoa G., Morrell, John, Mullens, Katherine, Burmeister, Aaron, Miles, Susan, Farrington, Nathan, Thomas, Kari, and Gage, Doublas W. Segway Robotic Mobility Platform. In *SPIE Proc. 5609: Mobile Robots XVII* (Philadelphia, PA, October 2004).

[44] Odashima, Tadashi, Onishi, Masaki, Tahara, Kenji, Takagi, Kentaro, Asano, Fumihiko, Kato, Yo, Nakashima, Hiromichi, Kobayashi, Yuichi, Mukai, Toshiharu, Luo, Zhiwei, and Hosoe, Shigeyuki. A Soft Human-Interactive Robot RI-MAN. In *Video Digest of IEEE/RSJ International Conference on Robots and Intelligent Systems* (Beijing, October 2006).

[45] Platt, R., Fagg, A. H., and Grupen, R. Nullspace Composition of Control Laws for Grasping. In *International Conference on Intelligent Robots and Systems (IROS)* (Laussane, Switzerland, 2002), IEEE/RSJ.

[46] Platt, Robert. *Learning and Generalizing Control Based Grasping and Manipulation Skills*. PhD thesis, Department of Computer Science, University of Massachusetts Amherst, 2006.

[47] Pratt, G. *Exploiting Inherent Robustness and Natural Dynamics in the Control of Bipedal Walking Robots*. Ph.d. thesis, Computer Science Department, Massachusetts Institute of Technology, 2000.

[48] Pratt, Gill A., and Williamson, Matthew M. Series Elastic Actuators. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (Pittsburg, PA, July 1995), vol. 1, pp. 399–406.

[49] Robinson, David W. *Design and Analysis of Series Elasticity in Closed-loop Actuator Force Control*. Phd dissertation, MIT, June 2000.

[50] Rosenstein, Michael, Platt, Robert, Deegan, Patrick, Sweeney, John, Brock, Oliver, Fagg, Andrew, and Grupen, Roderic. Haptic Coupling of Dexterous Manipulation and Dynamic Mobility. In *In Video Proceedings of the IEEE-RAS/RSJ International Conference on Humanoid Robots* (Los Angeles, USA, November 2004).

[51] Sakagami, Yoshiaki, Watanabe, Ryujin, Aoyama, Chiaki, Matsunaga, Shinichi, Higaki, Nobuo, and Fujimura, Kikuo. The Intelligent ASIMO: System Overview and Integration. In *Proceedings of the 2002 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems* (Lausanne, Switzerland, October 2002), pp. 2478–2483.

[52] Saxena, Ashutosh, Driemeyer, Justin, Kearns, Justin, Osondu, Chioma, and Ng, Andrew Y. Learning to Grasp Novel Objects using Vision. In *To appear in: 10th International Symposium on Experimental Robotics (ISER)* (2006).

[53] Segway Inc. Robotic Mobility Platform Models. `http://www.segway.com/products/rmp/models.html`, August 2006.

[54] Sweeney, J., Brunette, TJ, Yang, Y., and Grupen, R. Coordinated Teams of Reactive Mobile Platforms. In *Proceedings of the IEEE Conference on Robotics and Automation* (Washington, D.C., May 2002).

[55] Takenaka, Toru. The Control System for the Honda Humanoid Robot. *Age and Ageing 36-S2* (2006).

[56] Takubo, Tomohito, Inoue, Kenji, and Arai, Tatsuo. Pushing an Object Considering the Hand Reflect Forces by Humanoid Robot in Dynamic Walking. In *Proceedings of the 2005 IEEE International Confrence on Robotics and Automation* (Barcelona, Spain, April 2005), pp. 1718–1723.

[57] Takubo, Tomohito, Inoue, Kenji, Sakata, Kotaro, Mae, Yasushi, and Arai, Tatsuo. Mobile Manipulation of Humanoid Robots-Control Method for CoM Position with External Force. In *Proceedings of the 2004 IEEE/RSJ International*

Conference on Intelligent Robots and Systems (Sendai, Japan, September 2004), pp. 1180–1185.

[58] Thibodeau, Bryan J., Deegan, Patrick, and Grupen, Roderic. Static Analysis of Contact Forces With a Mobile Manipulator. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation* (Orlando, FL, May 2006).

[59] Torres-Jara, Eduardo. Obrero: A Platform for Sensitive Manipulation. In *5th IEEE-RAS International Conference on Humanoid Robots* (2005).

[60] Wyrobek, K., Berger, E., der Loos, H.F.M. Van, and Salisbury, K. Towards a Personal Robotics Development Platform: Rationale and Design of an Intrinsically Safe Personal Robot. In *International Conference on Robotics and Automation (ICRA)* (2008).

[61] Yoshida, Haruyuki, Inoue, Kenji, Arai, Tatsuo, and Mae, Yasushi. Mobile Manipulation of Humanoid Robots-Optimal Posture for Generating Large Force Based on Statics. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation* (Washington, DC, 2002), pp. 2271–2276.

[62] Yoshikawa, T. Analysis and Control of Robot Manipulators with Redundancy. In *Robotics Research: The First International Symposium* (1984), pp. 735–747.