ENDPOINT FORCE SENSING FOR MOBILE MANIPULATORS

A Thesis Presented

by

EDWARD P. HANNIGAN

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

MASTER OF SCIENCE

September 2008

Computer Science

CONTENTS

Page

CHAPTER

APPENDICES

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

CHAPTER 1

INTRODUCTION: MOBILE MANIPULATORS

Applications of robotics involve exploration in space and the oceans, logistics and supply chains to get products to the right place at the right time, agriculture and mining, healthcare and telemedicine, etc. The list of potential applications of robotics may have no limits. Mobile manipulators will be entering controlled smart-rooms, hospitals, offices, and even residential homes as soon as they are able to do useful work in the environment reliably and at an affordable price.

A mobile manipulator is a robot with manipulation capabilities that is not fixed in place in the environment. It is mobile, so it can navigate through the environment and is able to manipulate the environment by moving objects around. The first task for these manipulators to interact with real-world objects is usually to grasp or push them. Manipulation can be done in environments designed for humans and includes tasks such as pushing elevator buttons, toggling light switches, or turning knobs on household appliances. Robots can use non-contact and contact-based sensors. These include cameras, laser range finders, and 6-axis load cells. The focus of this thesis is on using and evaluating force feedback on a new mobile manipulator by sensing motor currents. Applications are developed based on sensing joint torques with motor currents to show the usefulness for both mobility and grasping tasks.

1.1 Force Feedback in Personal Robotics

Personal robotics aims to make robots that perform tasks in the home. There are now personal robots available that can help in cleaning, can comfort us in the form of teddy bears, act as companions or pets, and allow virtual visits through an embodied interface. Many of them are still expensive for many consumers and do not yet do enough to justify the price. Computational power has been increasing exponentially as Moore's Law predicted and is expected to continue for at least a decade[13][14]. This combination of decrease in cost and increase in processing power is likely to be one of the key ingredients for many robotics applications to become more widespread. The type of force sensing can heavily influence price but it is possible that less pricey sensors can accomplish some of the same tasks.

A simple form of a mobile manipulator, which is more mobile than manipulator, is the Aibo that was made by Sony. It is a personal robotic pet dog that is able to walk, use a camera for vision, recognize spoken commands, and is considered to be autonomous because it can learn based on external stimuli from the owner and environment. Owners can also program in new behaviors. Some universities use these robots for research because they are relatively inexpensive research platforms. In the RoboCup autonomous soccer competitions Aibos were programmed to play soccer, which involves sensing opponents and the ball, planning, and taking actions such as moving or kicking the ball[19]. There are touch sensors on the feet to detect contact with the ground and 18 of 20 joints have force sensing.

There are therapeutic robots whose value is to provide comfort for children or patients. A personal robot like this is being developed by MIT called the Huggable robot, which is a teddy bear[17]. It is a social robot that interacts with the person in front of it. While it is limited in the ways it can currently interact, it is complex and contains full body sensor skin sensitive to touch, tensiometers in its legs and feet, cameras, speakers, microphone, and can move its head. Technology like this could let patients tell doctors where their pain is and teach children other languages. Maja J. Matarić studies how social robots with personality can help humans to recover from strokes through human robot interaction (HRI)[18]. The focus of robots like these is on social interaction but there can be physical interaction too.

There is expected to be a larger number of eldercare clients from the generation of baby boomers in need of caregivers over the next decade that could put strain on our centralized medical providers[1][6]. Mobile manipulators in homes of the elderly could help fill the shortage in eldercare and promote the safety and well being of the elderly. For instance, robots can help promote health among the elders by giving reminders to take medications and providing the remote interface of physicians and family members. Remote presence avoids costly and time-consuming transportation that could be difficult for elders with limited mobility. The robot may be able to increase their independence allowing them to live longer with greater quality of life and thus delaying or offering alternatives to nursing homes. Family members could have a remote presence in an embodied interface to check-in and interact with them. The robot has the ability to do useful work in the environment such as applying medical instrumentation that could monitor an elder's health.

There are many research areas in personal robotics that overlap with all of robotics and are not limited to, but include, navigation, embodied interfaces, human robot interaction, and mobile manipulation. Standardized access to force feedback from cost-effective sensors will improve the function of currently available commercial robots and it is a critically enabling technology for realizing new applications like healthcare within the personal robotics market.

1.2 Force Feedback Considerations

Force feedback allows interaction force with the environment to be measured or controlled. Sensors and actuators are chosen to make sure the robot has the required physical hardware to generate and measure forces. Strength of actuators and sensitivity to torque or force are related. For instance, gear trains on motors will decrease speed to provide more torque but decreased too much, it will lessen the ability to sense torques by observing motor currents. Joint torque sensing methods by motor current sensing, strain gauges, and series elastic actuators will be examined in the next chapter.

Electronics include the circuit boards to power the actuators including H-bridges to control DC motors and boards that sense other signals such as force/torque sensors, current, position, and temperature sensors. All these can contribute to estimates of contact forces and provide feedback for force control tasks. Electronics involves microcontrollers or other hardware such as field programmable grate arrays (FPGAs) that may handle low-level motor control. This hardware is often custom designed for particular robots to meet required specifications. Signals typically require some filtering to gain useful measurements or be useful for joint torque control that may be used for force control.

Contact interaction-based navigation is done with the biped robot Bonten-Maru II using 6-axis force/torque sensors on end-effectors[21]. It has been demonstrated to navigate using a motion algorithm consisting of searching tasks, self-localization, correction of locomotion direction, and obstacle avoidance. Motivation for this work was that contact based sensing is important to robots to support perceptual-guided navigation with sensors like vision or laser range finders. Non-contact approaches using vision and laser range finders is used by many autonomous mobile robots for simultaneous localization and mapping (SLAM) for navigation[8].

1.3 Introducing the uBot-5

The uBot-5 platform shown in Figure 1.1 is a dynamically balancing mobile manipulator that has the potential to perform many tasks in a variety of environments. It is a differential drive mobile robot that balances on two wheels with a trunk rota-

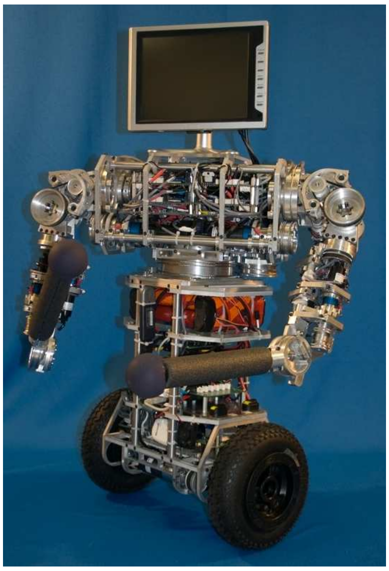


Figure 1.1. Picture of the uBot-5 platform

tion and two 4 degree of freedom arms. This is a good platform to use in tasks where a robot interacts closely with people since it has low input impedance by virtue of the cart-pole mobility system. It will move away if someone inadvertently bumps into it to maintain balance. The platform can accommodate mass placed high, which helps with balancing and ability to exert more force pushing and pulling forces than an equivalent statically stable robot of the same mass[2]. The smaller footprint has the benefit of being able to travel into tighter spaces. This robot must always consider postural stability as a component of every task, it must brace against environmental if stability is marginal, and right itself if it should fall—safely and routinely. The end-effectors can sense forces or deflections from the environment with compliant arms. Performing work in the environment involves forming grasps and to be able to control forces on objects. The spherical rubber end-effectors used by the uBot-5 are well suited for these types of tasks provided there is useful force-feedback for these types of tasks.

The uBot-5 has 11 degrees of freedom and the axis of rotation for each joint is shown in Figure 1.2. The arm is comprised of four revolute joints. From proximal to distal, they are the tilt, pan, twist, and elbow joints. The axis of rotation is directly through the center of each of the cylinders. The arms are actually larger and more complex but are shown in Figure 1.2 as cylinders to emphasize the kinematics of the robot.

All degrees of freedom are actuated by coreless DC motors that have an ironless core. Benefits of coreless motors include low inertia allowing faster acceleration, high efficiency, and no cogging. No cogging is desirable because there is no magnetic field acting on the iron laminations inside the motor so there is no torque ripple, allowing the motor to operate smooth even at low speeds.

The compute architecture on the uBot-5 is shown in Figure 1.3. A custom-made 12-channel FPGA based servo motor controller provides real-time guarantees on position and velocity embedded PID controllers that run at over 2 kHz. Quadrature

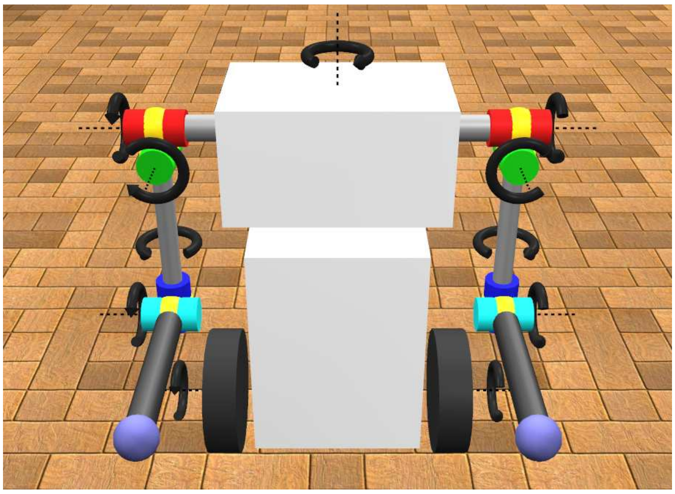


Figure 1.2. Rendering of the uBot-5 shows the revolute joints in the arms

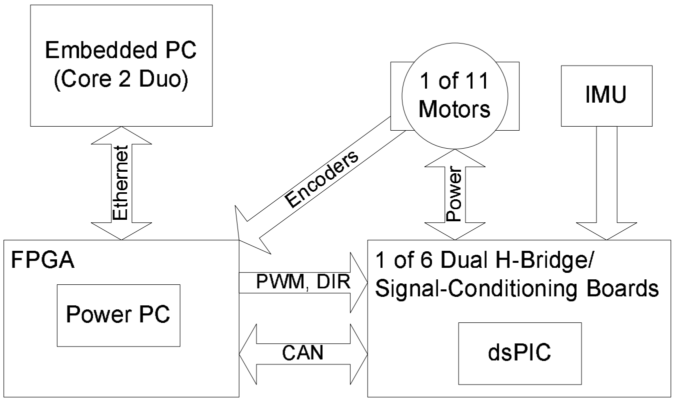


Figure 1.3. Compute architecture for uBot-5

encoders are connected as inputs to the FPGA for position feedback and computing motor velocities. An embedded PowerPC provides low-latency raw Ethernet connectivity with an onboard Intel Core 2 Duo. Raw Ethernet communication allows streaming data with Ethernet frames in addition to sending reliable data in the same frame. Closed-loop application-level controllers connected via Ethernet are able to run at over 500 Hz.

H-bridge amplifiers and signal conditioning for additional sensors are provided on six separate boards connected to the FPGA motor controller via CAN. Motor currents, accelerometer and gyroscope readings, and strain measurements from programmable gain amplifiers are updated on the FPGA and the PowerPC will read the latest contents within a quarter of a millisecond after a packet over CAN is sent. In the fabric of the FPGA, there is hardware dedicated to polling two networks of five SPI to CAN converters and placing the contents of new packets into dual ported memory that can be read via the OPB bus by the Power PC.

Presumably, the robot will be moving around in an environment where it does not have a complete or accurate map. Environments with people are generally changing rapidly and the robot must accommodate changes such as furniture moving around, unexpected new packages, and pets. Pushing, grasping, and righting from prone require the uBot-5 to measure forces and displacements. Robots like the uBot-5 can posture their arms to sense objects for manipulation or for mobility (as humans do when groping through a dark room). Force feedback is indispensable and complements other sensors like vision and laser scanning.

External loads on end-effectors are commonly measured with 6-axis load cells but are not ideal for the uBot-5. These types of sensors introduce significant mass at the end of the manipulator and are themselves mechanically delicate. This robot is small so it has low potential energy, but falls will likely damage endpoint load cells before other parts of the robot.

In the future, it is likely that we will need additional actuators and sensors for a manipulator at the end-effector to do more dexterous manipulation of objects in the environment. We could choose to use manipulators like these to achieve some of the same goals in this research, such as grasping objects. Again, this presents some of the same issues when specifying technologies for endpoint force sensing as we are addressing in this thesis.

Rather than adding more sensors, my goal in this thesis is to evaluate methods for using the force sensing that is already in the uBot-5. In particular, joint torques are sensed using motor currents and deflections are measured in compliant manipulators. In the next chapter, we see how joint torques are estimated and later how the endpoint force is found for the observed joint torques. The accuracy of sensing endpoint forces is tested over several configurations for an arm, with applied reference forces of different magnitudes and directions. Implicit force control using motor currents will be useful making the end-effectors compliant. The arms are kinematically conditioned for the force and velocity sensing for mobility experiments that follows. The uBot-5 demonstrates that it is able to navigate a maze using compliant end-effectors for sensing contacts. It can also autonomously grasp and lift certain objects. The limitations encountered during development of these applications leads to some recommendations for future hardware revisions of the platform.

CHAPTER 2

METHODS FOR ENDPOINT FORCE SENSING

2.1 Strain Gauges

Strain gauges can be glued to beams that bend as they are strained. They have thin foil patterns inside an insulating flexible backing and change resistance as the foil is deformed. Temperature affects the resistance of strain gauges but the effect of temperature can be compensated for by using 2 strain gauges in series. Sensitivity can be improved by using 4 of strain gauges in a configuration known as a Wheatstone bridge. Strain gauges may only change in resistance by about 1 % so the change in voltage is usually in the range of tens of millivolts. Additional hardware such as programmable gain amplifiers are needed to amplify the small change in voltage. Wires and connectors in series with the sensors will affect the quality of the measured signal. There is required calibration to relate a difference in voltage to joint torque. Strain gauges can be permanently damaged if they are strained too much. However, actual joint torque after the motor gearbox is measured so joint torque measurements are not affected by inefficiencies in the gearbox or motor.

2.2 Series Elastic Actuators

The research on Domo by Aaron Edsinger and Jeff Weber shows that series elastic actuators have many benefits including desirable joint torque sensing properties[7]. Series elastic actuators use an elastic spring element inline with the motor output. The force causing deflection in the spring can be measured by measuring the displacement in the spring. Joint torque can be found using Hooke's law for a spring (F = -Kx where K is a spring constant and x is the spring displacement). A potentiometer or Hall-effect sensor can be used to measure the deflection in the spring. In addition to measuring the torque more directly than other joint torque sensing methods the spring helps to add shock protection to motor gearboxes. Force control performance is improved and the effects of backlash, torque ripple, and friction are filtered by the elastic element[20]. High bandwidth force control approaches are prone to problems of contact instability with hard surfaces, noise, and low power density. Series elastic actuators trade off achievable bandwidth for gains in stable and low noise force control. The hardware required for this method of joint torque sensing is simple and low cost[7]. On the other hand, while just a series element such as a spring sounds simple to add, there is difficulty in constraining an elastic element to be elastic in only one degree of freedom (springs can deform in more than one direction) and may require components with very low friction such as ball screws and precision linear bearings[12]. There is increased complexity and bulk in the mechanical design.

2.3 Motor Currents and Modeling

Joint torques can be estimated in the uBot-5 by measuring motor currents and using motor torque constants specified in the datasheet of the motors. Motor torque is equal to the motor torque constant times current drawn by the motor. Actual joint torque can be calculated by knowing the speed reductions and the efficiency of the speed reductions. This is perhaps the simplest method of getting joint torques and motor current is readily available from most motor controllers so it is likely to not require additional hardware.

A previous study on motor current sensing along with other force sensing techniques have found that motor current sensing is only accurate when an arm is actively applying force [10]. A certain threshold of force needed to be exerted and Cartesian velocity needed to be reversed to reduce the load to get accurate force measurements from joint torques. The reason motor current sensing did not work when force was being applied to the end-effector was because of stiction effects in the gear train.

John W.L Simpson describes a sensorless force estimation technique capable of dealing with non-ideal transmission, friction, motor cogging, and other effects using only motor current and positions[16]. The torque is assumed to be accelerating the mechanical components, due to friction, overcoming a number of position dependent torques caused by the drive train and motor, and torque derived from operation of the end-effector. An accurate system model was developed to separate the applied torque component from the rest and includes inertia, viscous and coulomb friction, and position dependent torque components. Gears, belt teeth, and shaft eccentricities are a few examples that cause position dependent torque variations.

2.4 Joint Torque Sensing with Motor Current

Current sensing is perhaps one of the easiest ways to obtain force feedback on the end-effector of a robotic arm. It is available on many motor controllers because it is a way to protect the motors from overheating. However, current sensing provided with motor drivers typically has low accuracy and is noisy. Moreover, to measure endpoint loads using this method, low gear ratios and backdrivable actuators are required. The uBot-5 has been designed to have relatively low gear and pulley ratios so that torques due to external loads will be observable as motor currents.

Each uBot-5 joint is actuated with a brushed DC motor. Gearboxes and pulley reductions with belts or cables reduces the speed and increases the torque at each joint. The motor is always able apply torque to the load so the following formula works even with high gear ratios. The following formula relates the torque delivered to the load as a function of motor and transmission parameters at zero (locked rotor) velocities:

$$\tau_{Load} = K_t I R_G R_P \eta \tag{2.1}$$

where,

K - motor torque constant (Nm/A)
I - motor current (A)
R_G - gearhead reduction
R_P - pully reduction
η - efficiency of gearhead

The uBot-5 incorporates significant reductions in the transmission so it is not clear that current sensing techniques will work well for estimating external loads. With partially backdrivable joints like those on the uBot-5, torque applied to the joints can be computed in much the same way. When torque is applied to a partially backdrivable joint the motor will sense a fraction of applied torque due to speed reductions and any loss due to inefficiency because of the gearheads. Therefore, when joint velocity is zero, an external load generates a joint torque and with prior knowledge of motor parameters and measurements of motor current, we can estimate the external load as:

$$\tilde{\tau}_{Load} = \frac{K I R_G R_P}{\eta} \tag{2.2}$$

2.4.1 Joint Torque Error Analysis

The uBot-5 uses the integrated current sensing built into the LMD18200 H-bridge amplifiers that drive the motors to measure currents. The chip sources a current proportional to the current flowing through the motor which is 377±13.8% μA/A at 25° C and is 377±20.5% μA/A over the full operating temperature range. Current sourced by the LMD18200 flows through a resistor with 1% tolerance. By Ohm's law (V = IR) the voltage across this resistor over the operating temperature range is only accurate to within (20.5%)(1.01) = 20.7%. This voltage is read by a 12-bit analog to digital converter so a 1-bit error in sensed current corresponds to 1.6 mA error in measured current. The accuracy of the ADC is significantly less important considering that a motor drawing a typical current of 1 A cannot be determined with more accuracy than 207 mA. The H-bridges can sustain motor currents of up to 3 A (continuous) and it is not uncommon for the tilt motor to draw 2 Amps or more. Motors for elbow, twist, and pan are only rated for handling 1.4 A (continuous) and the tilt motor can handle 5 A (continuous).

The exact efficiency that torque is transferred from the joint to the motor or vice versa is not known. The manufacturer of the motors supplies rough estimates for gearhead efficiencies. Efficiency of the gearheads could be different from what the manufacturer says because of cumulative wear. We assume that that efficiency for transmitting torque in both directions is the same. All the gearheads of the motors used throughout the tests conducted are 70% efficient. There are also unmodeled losses in efficiency in the belt and cable drives.

In order to determine how accurately joint torques can be sensed in the uBot-5, experiments were performed on separate joints. These experiments were performed directly on the robot to get the most accurate results possible and to see the difference between computed joint torques and applied joint torques. Motors in position mode were given a fixed reference position. For each joint, known reference forces from 1 N to 15 N were applied to arms configurations where the motor being tested draws little current due to gravity. This approach allows evaluation over the largest range in current. The joint was preloaded initially by applying force in the direction the loads were going to pull to load the gearbox and belts/pulleys in the direction of the test. After removing this force, the joint is expected to return to the same position. In this state, the current drawn by the motor can be tared (re-calibrated to zero). There is significant joint friction in the bearings, gearboxes, and pulleys that causes the joint to come to rest at different positions and consume different amounts of current due to stiction. The approach taken to tare motor current affects the calculated joint torque that the load applies.

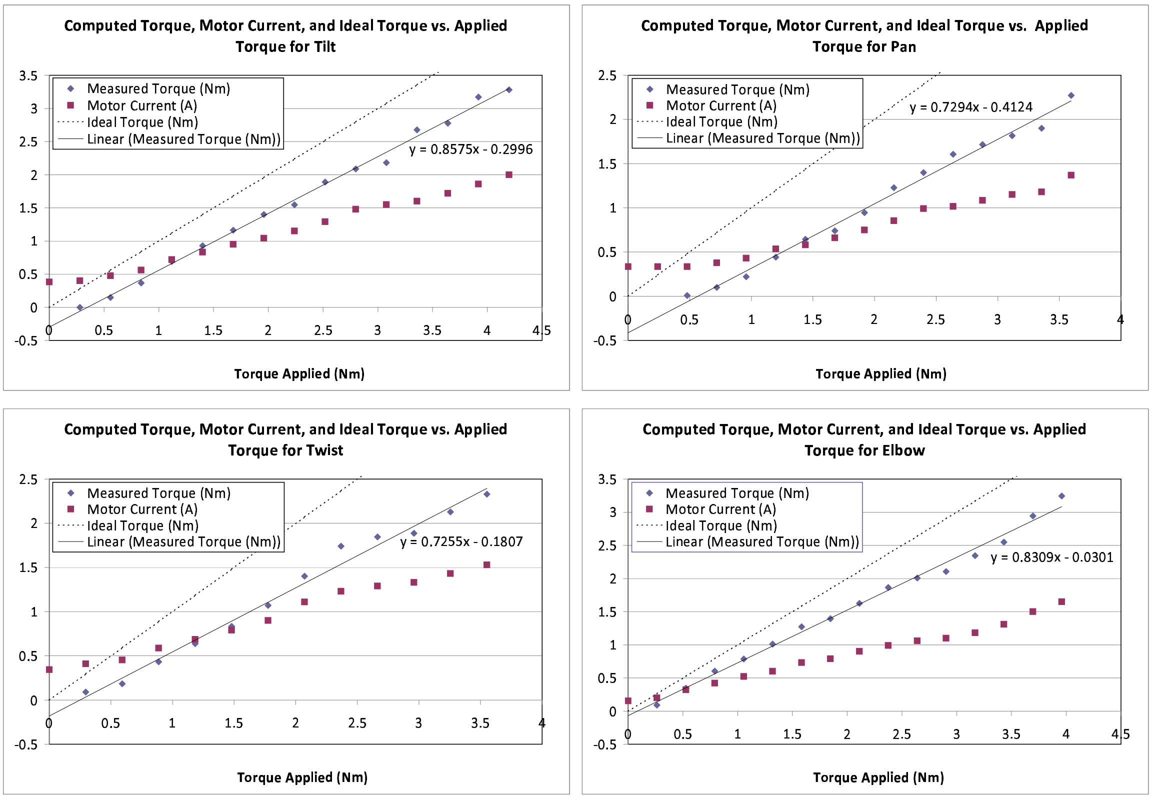


Figure 2.1. Results of applying joint torques to individual joints

In these tests, torque was measured soon after it was applied to the joint and quickly released to prevent large temperature rises in the motor or H-bridge. When applying larger forces to joints the current sensed would suddenly jump after a short amount of time. After applying a torque that causes an increase of current of about 1.5 A the motor would give a higher current reading of a couple hundred milliAmps more a few seconds after the force was applied. This is probably due to a rise in temperature of the H-bridge chip or the motor. Friction was ignored in computing joint torques so any losses in torque due to bearings, cables, or belts are ignored and this is likely significant.

Figure 2.1 shows graphs of computed torque, motor current, and ideal torque vs. applied torque for each motor in one of the arms. Motor current increases as larger loads are applied to the joints so the uBot-5 motors in the arms are backdrivable and external loads can be estimated. Computed torque is lower than applied torque so the system is lossy. The slope of a line of best fit for computed torque in the each graph can be used as an estimate for the efficiency of the backdrivable system. This slope can be used to scale sensed joint torques to more accurately estimate external loads applied to the joints.

The inefficiencies in backdriving the joints are quite large. Backdriving the tilt and elbow joints which use cable drives show 10% or more efficiency than backdriving the pan and twist joints which use belts. It looks like cable drives might be more efficient but there are still other factors such as friction and stiction. We know that stiction can prevent accurate joint torque sensing when backdriving motors so this may be skewing the results in the graphs.

2.5 Proprioceptive Feedback

In some cases, motor currents provide useful forms of force feedback, however, regular monitoring of motor current also has other important uses. Proprioception is the sense of the relative position of neighboring parts of the body and the feedback is provided solely on the status of the body internally. Animals and humans are quite good at judging and managing joint loads in their bodies. Robots, on the other hand, do not do as well. They need to be robust to last through harsh conditions and deliver high performance. There are other important research topics involving the use of motor current sensing in robotics.

2.5.1 Load Distribution

It is important that motor loads are distributed in a way to avoid thermal limits. Motors are able to deliver higher torques for short amounts of time and loads on motors should be balanced to get the motors to last as long as possible. DC Motors have bearings, brushes, and commutators that wear with use and it is desirable to distribute wear evenly. Sharing the load and having stronger motors do more work is a way of load balancing. Moreover, heat is dissipated more quickly when more motors are dissipating the energy. Actuators from the entire robot should participate in both mobility and manipulation tasks. Monitoring current use is a good way of assessing load distribution in a mobile manipulator.

2.5.2 Energy Efficiency

When the robot is idle but powered up it is best to have actuators drawing low current. Sometimes it is possible to exploit the structure of a robot by having limbs in configurations where virtually no motor torque is required. The legs in the human skeleton are able bear the weight of the human body without muscle effort. Robots should be able to identify similar low current configurations. Tasks can be optimized to consume less energy by moving arms through different trajectories. If the robot repeats motions often then perhaps there is a similar, more efficient motion that has the same result. Modeling how much current will be drawn is more difficult but sensing motor currents is easy and accurate.

2.5.3 Safety Thermal Protection

Current sensing can be used to protect the motors from overheating. A simple approach to protecting the motor would be to set a fixed current limit that is always safe. However, it is useful and safe to allow higher currents for short durations. Current sensing and motor temperature along with a more detailed motor model can be valuable in getting higher performance safely. Motor casing temperature may change relatively slow compared to the temperature rise of the motor windings. Motor current can be integrated over time, taking the thermal characteristics of the motor into account, to model winding temperature inside the motor. A robot can take action when approaching thermal limits of the motor by setting down a heavy object or just “rest” to allow thermal energy to dissipate.

CHAPTER 3

ENDPOINT FORCES

3.1 Relation of Endpoint Force and Joint Torques

The Jacobian is a kinematic mapping that transforms joint velocities to Cartesian velocities.

$$d\boldsymbol{x} = J d\boldsymbol{\theta} \tag{3.1}$$

The vector $\boldsymbol{x}$ contains translational components and $\boldsymbol{\theta}$ is a vector of joint configuration variables. The uBot-5 arms have four joint configuration variables and the endpoint moves in $\Re^3$.

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = J_{3\times 4} \begin{bmatrix} \Delta\theta_1 \\ \Delta\theta_2 \\ \Delta\theta_3 \\ \Delta\theta_4 \end{bmatrix}$$

In much the same way, Cartesian endpoint forces are mapped by the transpose of the Jacobian to joint torques.

$$\boldsymbol{\tau} = J^T \boldsymbol{f} \tag{3.2}$$

Motor currents can be used to control forces at the end-effector. This area of work is known as implicit force control[16]. The motor torque has to overcome friction as well as other torques and motor current due to force (and possibly torque) at the end-effector is relatively small. We look at the problem without taking into account the other torques. Individual torques on the joints can be computed to apply a force on the end-effector with Equation 3.2.

Given accurate joint torques due to gravity, the end-effector can be controlled similarly to if it were weightless, by applying torque that counters gravity and additional torques from the formula above. There are important factors of motor torque not being taken into account such as the torque accelerating the arm and friction.

The relation in Equation 3.2 can be inverted to map observed joint torques to forces on the kinematic device:

$$\boldsymbol{f} = (J^T)^{\#}\boldsymbol{\tau} \tag{3.3}$$

For the uBot-5 arm, the transformation from torque to force is redundant. There is more than one set of torques for the joints that will result in the endpoint force. There are possibly many generalized inverses to a non-square matrix. We employ the Moore-Penrose generalized inverse of the Jacobian transpose as in [10]. There is an assumption that there are no moments about the end-effector. With this pseudoinverse, $\boldsymbol{f} = (J^T)^{\#}\boldsymbol{\tau}$ is the least squares force that accounts for the observed torques through the non-unique mapping $\boldsymbol{\tau} = J^T\boldsymbol{f}$.

For an $m \times n$ matrix A where rank(A) = n and $m > n$ the Moore-Penrose pseudoinverse of A is defined as:

$$A^{\#} = [A^T A]^{-1} A^T (m > n) \tag{3.4}$$

When $m > n$, the pseudoinverse exists only when the $A^T A$ has an inverse. The matrix $A^T A$ is invertible if and only if rank($A^T A$) = n. The rank of matrix A is equivalent to the rank of $A^T A$ because the rank of a matrix and its corresponding Gram matrix are equal.

3.2 Reducing Motor Current from Stiction

Stiction in the joints is due to friction in the motors, gearboxes, pulleys, and bearings coming to rest. The joints are unable to move until enough torque is generated to overcome static friction. Stiction causes extra current to be drawn because the joint comes to rest sooner than if there was no friction. Stiction significantly affects the results in nearly all the experiments performed and particularly in sensing small endpoint forces. We attempt to negate the affect of stiction to improve our results, which works with the motors in position mode. Assumptions are made that external torque remains constant throughout the procedure, the effects of stiction are identical in nearly the same joint configuration, and that a position error causes the motor to apply torque proportional to the position error. Position error is defined as the desired position minus the actual motor position. The procedure described can be performed for each arm to minimize motor current drawn from stiction.

Each joint can be driven one at a time, slowly in each direction, from rest until the joint moves some small amount (0.003 radians worked well). Actual position is monitored while desired position is either increased or decreased at a constant rate to cause a small change in actual position. When the desired change in actual position occurs the position error is recorded for each direction. The two position errors we record are ones that just exceed the effects of stiction in both directions. Stiction effects are the same in both directions so just as much torque had to be applied to overcome stiction and cause the joint to move. We have a low position error that moves the joint in one direction when setting the desired joint position to the actual position plus the low position error and a high position error that can move the joint in the other direction when setting the desired joint position to the actual position plus the high position error. Setting the desired position for the joint to be the actual position plus a position error in between the low and high position errors should cause no movement since otherwise we would have a higher low position error or a lower high position error. The static friction torque that is overcome is equal and in opposite directions so setting a desired position error of the actual position plus the average of the low and high position errors will have the motor apply minimum torque from stiction.

Reducing motor current from stiction is important because motor currents can be tared after running a routine on the arms in the same way every time. Without it, motor currents can be drawing different amounts of current because the joint can come to rest in slightly a different way. In the same configuration, the motor currents should always be the same after running this routine multiple times, which is important for repeatability. It is useful to run this routine before reading motor currents to get more accurate motor currents due to gravity. When additional force is applied and the arm is in position mode, this routine can also be used to get more accurate motor currents that can be compared to tared motor currents to compute differences in motor currents.

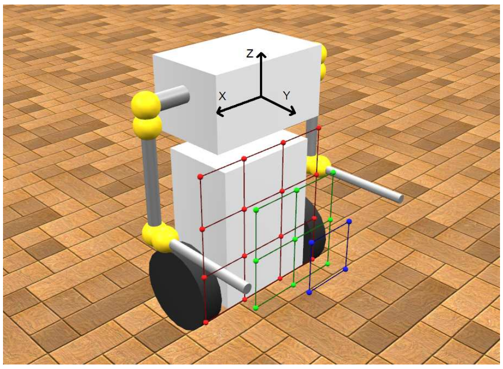


Figure 3.1. Cartesian grid locations chosen for applying endpoint reference forces

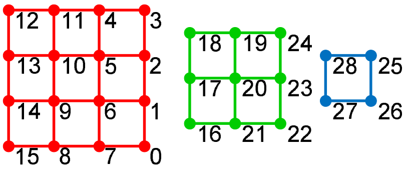


Figure 3.2. Assigned grid location labels

3.3 Empirical Analysis

To evaluate the accuracy in both magnitude and direction of the sensed endpoint forces over various arm configurations, reference forces were applied to the end-effector for different joint configurations. A custom setup using pulleys, string, and a set of reference weights was used to apply known force to the endpoint of the right arm in six directions. The force was applied straight in ±X, ±Y, and ±Z directions as accurately as possible. Arm configurations were chosen that positioned the endpoint on a regularly spaced Cartesian coordinate grid as shown in Figure 3.1 and with assigned grid labels shown in Figure 3.2 to identify the sampled locations. All endpoint positions on the grid were restricted to being in front of the robot and below the shoulder joint because this is where the arms are expected to be placed in most manipulation and mobility tasks.

The arms are redundant so endpoint positions are reachable by more than one joint angle configuration. An endpoint position controller, described in Appendix B, moves the endeffector to the grid location. Before running this controller, the arm was manual positioned in a configuration where the endpoint was close to the desired grid location. In general, configurations were usually picked that had more proximal links to the body heading downward. It was important to apply the reference force by being able to pull straight on the end-effector with the string, which sometimes limited the acceptable configurations of the arm. In these cases, the arm configuration was chosen with the proximal links in the arms facing outwards. Joint configurations of the arm were saved so that the six applied reference loads to the endpoint, at the same grid location, used the same desired joint positions.

Direction error can be calculated by computing the angle between the computed force vector and the reference force vector. Computing direction error like this results in loss of structural error. We choose to look at errors in altitude and azimuth so structural errors are still visible. Altitude and azimuth are commonly used to measure the position of objects in the sky where altitude measures the angle between the object and the horizon and azimuth is the measure of the angle east of north. We use a similar coordinate system shown in Figure 3.3. As an example, the direction of a force along the positive X-axis can be represented as altitude of 0° and an azimuth

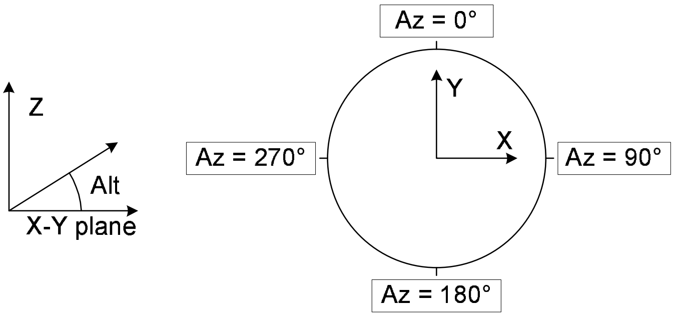


Figure 3.3. Altitude and Azimuth

of 90°. A direction of a force along the positive Z-axis has an altitude of 90° and an altitude of $-90°$ for a force in the direction of the negative Z-axis but azimuth provides no additional information on both of these direction. When we compare computed endpoint forces to reference forces, we assume a reference azimuth of 0° in the positive and negative Z directions.

Reference weights of 2 N, 4 N, and 8 N were the chosen reference loads for these experiments. These light weights show the sensitivity to forces at the end-effector. Although they are light, the routine described in the previous section to reduce stiction could not run because one of the joints did not have enough torque for 3 reference forces applied with the 8 N weight. Tables 3.2, 3.4, and 3.6 show the direction error in altitude (ϕ) and azimuth (θ) between computed force and applied reference force. Tables 3.1, 3.3, and 3.5 show magnitudes of computed force applied to the endpoint.

Computed magnitude is summarized by the average, median, and standard deviation shown in the tables for each direction that force is applied. The computed force for 8 N reference forces had average magnitudes all below the applied force, which was not seen in the case of applying 2 N and 4 N reference forces. It is not obvious why this would be but there may be relatively more noise with less force applied. The average of the average 6 magnitudes are 2.06 N, 3.89 N, and 7.44 N for the 2 N, 4 N, and 8 N reference forces respectively. Looking at the standard deviations, we see there is relatively more precision of computed forces but the average magnitudes show accuracy is lower for increased reference load.

Average direction error in altitude and azimuth generally decreases with increased reference force and standard deviation generally increases but usually not by a factor of 2 even with twice the reference force implying increased precision. Altitude error for reference forces applied in the positive and negative Z directions have the same sign in each column since altitude cannot be less than -90° or greater than 90°. The average for the columns are expected to be zero except for ϕ_{Err} in Z directions which would always require computed force is equal to the reference force.

The graphs in this section show magnitude and direction errors for computed force applied to the end-effector. There is almost 90 degrees of error in direction for some of the 2 N tests. Results are better with 4 N reference forces applied probably due to increased force so we examine some of the worst cases of these by retesting certain joint configurations. There is worst direction error is for grid locations 13, 14, 17, and 27 for 4 N loads. The original results are listed in a new way so that they can easily be compared with the new results for magnitude and direction errors. In the repeated tests the arm was left in the configuration before currents were tared for at least 30 seconds. Maximum direction error went from over 43 degrees to less than 16 degrees and maximum magnitude error went from 1.87 N to 0.67 N. The difference in these additional tests is believed to be due to temperature changes of motors or the H-bridges. The arm configurations for these repeated tests have the more proximal links stretched further outwards so the configuration is significantly different then the previous configuration. Additional accuracy is believed to be gained by letting the temperature of H-bridges come to a constant temperature since the accuracy of current sensing is affected by temperature. Different motor currents from different configurations would cause changes in temperature to the H-bridges.

Table 3.1. Computed magnitude for 2 N reference loads

Grid Label	(2, 0, 0)	(-2, 0, 0)	(0, 2, 0)	(0, -2, 0)	(0, 0, 2)	(0, 0, -2)
0	2.20	1.67	2.13	2.00	2.14	1.74
1	2.28	1.90	1.70	1.74	2.04	1.68
2	1.83	1.79	2.01	1.70	1.94	2.18
3	1.91	1.48	1.94	1.44	1.63	2.22
4	2.23	1.71	2.18	2.80	2.04	2.15
5	2.24	2.45	1.77	1.88	2.12	2.00
6	2.24	2.03	2.03	1.85	2.26	1.85
7	1.97	2.18	2.19	2.14	1.92	2.16
8	1.63	2.26	1.54	2.04	2.11	2.09
9	2.11	2.02	1.64	2.20	2.02	2.07
10	1.31	2.81	2.28	1.89	1.73	1.94
11	2.23	2.22	2.11	2.67	1.63	2.38
12	2.14	2.52	2.18	1.96	1.94	2.49
13	1.30	3.81	3.06	1.90	2.08	3.65
14	1.99	4.05	3.34	1.82	2.82	1.96
15	1.66	2.71	2.83	2.51	1.55	1.97
16	1.80	2.28	2.30	2.31	1.86	1.99
17	1.35	2.92	2.79	2.06	1.80	2.61
18	1.87	2.25	1.78	2.28	2.11	2.01
19	1.63	2.15	1.99	1.57	1.99	2.17
20	2.11	2.00	1.84	1.57	1.76	1.75
21	1.41	2.28	2.16	2.59	1.77	1.58
22	2.22	2.05	1.63	1.91	2.05	1.76
23	2.06	2.36	1.38	1.92	1.72	2.11
24	1.73	2.19	1.92	1.81	2.01	2.00
25	1.86	2.23	2.12	1.89	2.08	1.74
26	2.00	1.90	1.47	1.67	2.08	2.25
27	2.11	2.28	1.59	2.00	1.75	2.58
28	1.47	2.73	2.26	2.52	1.68	2.23
Average	1.89	2.32	2.07	2.02	1.95	2.11
Median	1.97	2.23	2.03	1.92	1.99	2.07
Stdv	0.31	0.56	0.46	0.34	0.25	0.39

Table 3.2. Direction error in altitude and azimuth for 2 N reference loads

	(2, 0, 0)		(-2, 0, 0)		(0, 2, 0)		(0, -2, 0)		(0, 0, 2)		(0, 0, -2)	
Grid Label	ϕ_{Err}	θ_{Err}	ϕ_{Err}	θ_{Err}	ϕ_{Err}	θ_{Err}	ϕ_{Err}	θ_{Err}	ϕ_{Err}	θ_{Err}	ϕ_{Err}	θ_{Err}
0	6.52	8.43	2.40	-15.62	4.58	4.58	3.73	6.34	-5.54	-20.22	19.52	75.72
1	0.75	4.28	4.83	-7.28	8.12	12.38	-1.32	15.31	-9.84	-90.00	10.84	21.45
2	2.19	2.82	-1.92	-12.25	1.43	9.46	-10.51	31.02	-10.09	-7.35	5.49	19.98
3	13.63	-20.88	-33.65	-39.75	5.62	11.66	-31.86	35.37	-38.25	40.17	7.70	42.40
4	0.26	-6.97	-1.68	4.01	0.00	11.36	-47.37	74.35	-5.68	-10.01	9.58	0.00
5	1.79	7.44	3.98	2.58	-2.91	-1.62	-7.95	12.70	-7.88	1.85	9.94	-41.63
6	-3.58	7.98	4.52	-3.96	5.65	6.81	-9.65	14.57	-10.80	3.90	0.00	35.54
7	4.37	7.31	0.79	2.63	2.62	1.57	-4.02	-1.08	-13.10	-26.57	14.63	-22.78
8	3.87	2.47	3.04	-1.27	7.09	-10.20	-2.81	-1.12	-12.50	-40.60	12.56	-23.70
9	-0.27	6.52	-3.97	-2.56	1.05	7.00	-5.22	2.88	-9.89	45.00	0.00	30.26
10	-2.19	-9.68	2.04	2.24	2.77	-0.25	1.82	40.70	-6.16	29.36	11.66	-27.18
11	8.25	0.00	9.33	7.09	9.82	5.79	-35.75	79.36	-6.35	49.09	9.11	23.96
12	5.63	0.27	3.41	9.61	6.06	-2.37	-10.29	9.56	-5.82	-61.70	7.27	-83.88
13	-31.02	-47.55	-14.59	20.30	-11.69	-22.19	5.13	2.73	-67.68	-48.17	34.45	-44.41
14	-55.50	-57.99	-23.42	20.00	-23.47	-19.26	12.70	13.33	-15.30	47.78	0.00	-28.61
15	-15.73	-11.94	-15.63	12.17	-14.32	-15.66	-8.71	-8.10	-15.99	-40.31	15.32	-34.62
16	-0.32	4.13	-1.01	3.26	-1.25	-7.99	-5.96	-3.25	-14.59	31.37	8.13	26.57
17	-19.02	-78.69	-2.94	12.89	-0.41	-16.03	-2.50	0.28	-36.34	-49.57	25.79	-33.97
18	1.84	10.47	0.25	-3.31	9.70	-0.98	-2.77	5.79	-9.67	-57.17	0.00	-14.04
19	6.34	-1.77	9.64	-1.62	6.64	1.74	-17.80	52.06	-5.75	78.69	0.00	26.57
20	-6.53	6.31	-1.15	14.18	-1.87	0.31	-9.53	4.83	-15.00	-38.09	8.67	-17.74
21	-9.39	-2.06	-5.79	6.06	0.80	-11.78	2.43	-10.70	-6.09	-80.54	12.92	37.87
22	6.99	5.22	-4.48	6.74	11.32	14.12	1.20	-2.40	-5.66	-37.41	6.11	33.69
23	1.39	-0.28	1.21	2.18	-0.83	20.41	-2.09	8.09	-13.85	3.99	14.80	30.96
24	0.33	-1.66	6.82	4.74	9.90	-1.52	-6.03	20.15	-9.91	36.53	5.73	16.70
25	2.16	-4.64	4.63	7.51	7.59	1.64	-22.39	13.24	-5.62	-6.71	8.70	-17.74
26	0.00	5.45	0.30	-0.60	7.82	11.46	-6.88	8.33	-17.84	9.16	19.57	-7.59
27	-3.80	16.88	-9.34	11.01	1.44	-1.44	-6.03	1.15	-13.73	-69.44	17.54	-44.48
28	-15.39	2.82	-5.89	7.41	-6.86	-15.28	-11.21	11.45	-8.85	-69.15	15.39	23.46
Avg	-3.33	-5.01	-2.35	2.36	1.60	-0.22	-8.33	15.07	-13.92	X	10.74	X
Median	0.33	2.47	0.25	3.26	2.62	0.31	-6.03	8.33	-9.91	X	9.58	X
Stdv	13.63	21.28	9.50	11.59	7.83	10.91	12.60	22.25	13.03	X	7.94	X

Table 3.3. Computed magnitude for 4 N reference loads

Grid Label	(4, 0, 0)	(-4, 0, 0)	(0, 4, 0)	(0, -4, 0)	(0, 0, 4)	(0, 0, -4)
0	3.59	4.12	3.72	4.28	3.72	3.59
1	4.54	4.22	3.35	3.65	3.62	3.83
2	3.71	4.54	3.55	3.83	4.34	3.98
3	3.62	3.00	4.54	3.74	3.81	3.75
4	4.48	3.59	5.10	3.37	4.00	3.51
5	3.64	4.33	3.86	4.10	4.03	3.84
6	4.33	4.24	4.37	3.75	3.85	3.77
7	3.43	3.56	3.48	3.26	3.50	3.04
8	3.54	4.28	4.32	3.69	3.44	4.17
9	4.40	3.99	3.71	4.16	4.04	4.34
10	3.69	4.71	4.15	3.83	3.92	3.72
11	4.80	3.81	5.26	3.41	3.46	4.30
12	3.93	4.38	3.88	3.12	3.31	4.27
13	3.17	5.80	3.79	3.17	3.35	5.70
14	3.10	5.87	4.66	3.42	4.13	4.30
15	3.57	3.62	4.12	3.65	3.96	3.64
16	3.97	3.71	3.72	3.16	3.77	3.88
17	3.14	5.05	3.84	3.32	3.64	4.69
18	3.97	4.02	3.77	3.96	3.90	4.02
19	4.09	3.94	5.34	3.80	3.68	4.14
20	4.65	4.43	4.04	3.67	3.86	3.63
21	3.36	3.27	3.42	4.07	3.93	3.67
22	4.67	3.39	3.82	4.08	3.42	3.24
23	4.06	3.60	4.19	3.98	3.44	3.64
24	3.59	3.82	3.53	3.49	3.78	3.72
25	3.81	3.69	3.07	3.54	3.66	3.61
26	3.55	4.29	3.45	2.46	3.69	4.14
27	4.40	3.94	3.35	4.09	3.93	4.70
28	3.71	3.96	3.49	4.72	4.14	4.11
Average	3.88	4.11	3.96	3.68	3.77	3.96
Median	3.71	3.99	3.82	3.69	3.78	3.84
Stdv	0.48	0.65	0.58	0.44	0.26	0.51

Table 3.4. Direction error in altitude and azimuth for 4 N reference loads

	(4, 0, 0)		(-4, 0, 0)		(0, 4, 0)		(0, -4, 0)		(0, 0, 4)		(0, 0, -4)	
Grid Label	ϕ_{Err}	θ_{Err}	ϕ_{Err}	θ_{Err}	ϕ_{Err}	θ_{Err}	ϕ_{Err}	θ_{Err}	ϕ_{Err}	θ_{Err}	ϕ_{Err}	θ_{Err}
0	3.51	0.00	-1.25	-0.83	2.77	2.62	2.54	-4.16	-5.94	-64.72	6.05	27.30
1	4.67	6.34	2.72	-0.14	3.59	4.64	-7.08	10.81	-14.16	-75.96	4.14	85.60
2	1.85	3.25	6.96	2.29	3.71	-0.32	-7.50	6.96	-8.71	20.66	15.24	-60.26
3	7.94	-3.68	-1.34	-9.21	-5.56	-2.41	-0.15	0.00	-5.87	20.56	5.92	-29.36
4	-1.79	8.22	-0.80	-4.31	-10.96	-13.87	11.47	10.83	-4.05	51.01	16.24	-16.59
5	2.83	2.68	3.58	-2.12	5.95	4.94	-5.32	1.12	-17.66	35.75	4.14	75.07
6	0.66	4.77	0.41	0.41	5.65	3.56	-3.67	3.68	-4.13	-13.50	7.23	-10.62
7	-2.00	-1.34	-3.06	-2.26	5.44	7.45	-9.53	7.50	-6.13	-70.02	6.58	68.20
8	-9.10	-2.62	-1.07	-1.20	3.05	-1.33	-4.20	9.39	-13.14	-70.14	10.51	-36.25
9	5.22	6.28	-4.31	0.86	2.78	2.63	-5.24	2.08	-16.68	6.51	9.54	-21.04
10	1.24	5.60	0.00	-2.31	4.70	-4.15	-1.20	11.90	-8.19	-53.13	11.90	-46.55
11	5.62	-0.24	1.20	-0.15	-5.35	-18.82	4.71	2.53	-15.13	31.73	5.53	-18.89
12	-0.73	5.84	3.27	7.09	5.32	0.45	3.12	-0.18	-11.80	-23.20	3.92	-46.27
13	-13.87	-14.08	-14.17	15.69	10.49	6.48	-23.03	38.32	-14.72	-68.82	23.93	-46.23
14	-25.20	-15.27	-13.90	13.18	7.52	9.58	-33.15	25.04	-5.64	-5.86	12.38	-18.63
15	-18.79	-10.91	-4.59	1.27	11.62	5.54	-11.70	5.61	-4.07	-27.47	4.25	-41.63
16	2.02	4.63	-2.16	2.32	7.41	7.16	-3.08	7.82	-7.23	12.41	8.23	-15.35
17	-0.55	-12.13	-3.41	14.37	4.93	6.46	-14.48	21.54	-27.11	-46.46	21.95	-28.77
18	1.73	11.34	1.43	-3.57	2.74	1.98	-8.28	8.67	-5.81	-18.92	0.00	-47.73
19	3.36	8.31	5.24	-1.90	-1.39	-15.21	-3.02	1.21	-4.22	30.96	7.97	-2.01
20	-0.12	10.15	-1.42	5.83	0.43	3.55	-8.62	6.17	-4.13	-60.26	11.27	-42.66
21	-7.35	0.69	7.91	-16.31	1.68	2.01	-3.10	-0.99	-4.09	73.74	5.98	14.74
22	4.91	6.30	-0.17	-4.74	2.10	5.26	-0.84	-0.70	-6.20	-26.57	7.80	63.43
23	4.66	5.25	-1.27	-4.14	-0.27	-6.85	-2.02	3.46	-6.18	-34.16	7.36	-33.69
24	8.33	-4.84	-0.75	-9.34	3.57	1.79	-6.09	13.84	-4.17	-72.76	7.28	-10.20
25	-1.80	7.09	1.71	-6.55	7.11	15.82	-3.24	6.82	-5.99	-54.46	7.39	13.39
26	0.81	6.14	1.34	11.44	3.99	4.83	-10.54	13.17	-4.22	-68.20	14.94	-9.11
27	1.56	7.58	2.33	16.55	2.91	7.72	-5.89	5.36	-10.83	-73.74	21.93	-33.42
28	3.86	8.09	7.69	7.61	4.60	-6.59	-4.86	3.29	-9.77	26.57	8.00	-3.88
Avg	-0.57	1.84	-0.27	1.03	3.12	1.20	-5.65	7.62	-8.83	X	9.57	X
Median	1.56	4.77	-0.17	-0.15	3.59	2.63	-4.86	6.17	-6.18	X	7.80	X
Stdv	7.65	7.32	5.03	7.86	4.64	7.61	8.23	8.78	5.52	X	5.79	X

Table 3.5. Computed magnitude for 8 N reference loads

Grid Label	(8, 0, 0)	(-8, 0, 0)	(0, 8, 0)	(0, -8, 0)	(0, 0, 8)	(0, 0, -8)
0	6.68	7.73	7.30	7.65	6.46	6.44
1	7.48	8.12	7.46	6.59	7.03	7.25
2	7.90	7.90	7.75	8.24	7.16	7.08
3	7.69	7.38	7.62	7.64	7.99	8.19
4	7.36	7.37	6.93	7.38	7.11	7.94
5	7.28	8.08	7.77	7.93	7.18	8.40
6	8.44	7.32	7.50	7.13	7.64	7.03
7	7.01	7.11	6.56	7.54	8.16	7.19
8	7.37	7.44	7.30	7.91	7.28	7.54
9	8.19	7.73	6.50	7.07	7.88	7.46
10	6.96	8.10	7.13	7.58	7.35	7.36
11	7.96	7.76	7.40	8.13	7.33	7.67
12	7.50	8.15	7.82	7.13	8.01	7.72
13	6.53	10.23	10.07	5.78	6.32	8.54
14	6.94	NA	9.32	6.10	7.08	6.70
15	7.05	7.49	6.42	6.59	7.01	6.63
16	8.33	6.79	7.20	7.31	7.86	7.05
17	6.86	NA	8.85	6.15	6.89	6.81
18	7.62	7.07	6.71	7.67	7.17	7.57
19	7.73	7.48	7.33	7.97	7.13	7.88
20	8.11	7.09	7.25	7.33	7.16	7.31
21	6.77	7.41	8.38	7.36	7.20	6.00
22	7.61	6.91	7.72	7.22	7.44	8.98
23	7.71	6.84	7.15	8.88	7.37	7.31
24	7.42	7.56	6.77	7.54	7.23	7.09
25	7.10	7.88	7.41	8.52	7.28	6.64
26	7.39	7.53	6.92	6.19	8.58	7.71
27	7.55	NA	7.26	7.82	7.06	7.61
28	7.13	7.88	8.20	7.37	7.45	6.80
Average	7.44	7.63	7.52	7.37	7.34	7.38
Median	7.42	7.51	7.33	7.38	7.23	7.31
Stdv	0.49	0.66	0.82	0.73	0.48	0.66

Table 3.6. Direction error in altitude and azimuth for 8 N reference loads

	(8, 0, 0)		(-8, 0, 0)		(0, 8, 0)		(0, -8, 0)		(0, 0, 8)		(0, 0, -8)	
Grid Label	ϕ_{Err}	θ_{Err}	ϕ_{Err}	θ_{Err}	ϕ_{Err}	θ_{Err}	ϕ_{Err}	θ_{Err}	ϕ_{Err}	θ_{Err}	ϕ_{Err}	θ_{Err}
0	-8.52	-2.17	-0.07	1.04	0.55	0.00	4.95	-1.80	-19.22	-60.89	5.53	51.63
1	-2.68	0.46	0.64	2.40	-1.61	5.31	2.78	-2.00	-16.23	-78.23	5.21	71.27
2	-1.23	1.31	0.73	1.45	-2.14	-1.48	-1.81	-0.56	-4.28	-59.42	10.56	48.73
3	-3.06	3.13	-0.54	-1.09	3.39	1.58	-0.90	-3.53	0.00	90.00	0.00	0.00
4	-1.25	3.27	-0.31	6.00	0.33	9.88	1.79	0.70	0.00	-16.70	10.38	-18.18
5	-1.57	1.42	1.42	1.85	-0.44	0.44	0.36	4.70	-3.02	-78.96	2.80	19.65
6	-2.10	0.95	-0.47	8.09	-5.20	-2.46	-2.65	1.37	-5.08	1.68	5.30	21.80
7	-1.31	-0.33	1.21	0.24	0.70	4.11	0.68	1.60	-6.35	8.22	6.05	-33.91
8	-0.70	3.27	0.39	0.15	-2.75	0.08	0.80	4.13	-5.20	-79.48	5.90	-35.36
9	-2.66	4.28	-2.67	4.91	-0.71	2.91	-3.24	3.66	-8.17	13.54	0.00	-12.53
10	-3.46	-2.31	0.92	1.56	0.64	0.16	1.44	6.14	-2.99	39.29	0.00	-35.84
11	1.66	-0.72	2.44	4.51	2.17	2.56	0.99	3.53	-5.19	63.05	10.15	-8.49
12	1.38	-4.36	2.46	4.93	6.24	-2.73	0.96	0.64	-7.58	-4.48	13.71	-2.18
13	-13.46	-13.95	-9.40	14.86	-10.41	-17.63	-10.16	23.32	-22.71	-53.65	9.21	-45.00
14	-5.29	7.81	NA	NA	-13.72	-11.15	-11.63	12.57	0.00	-54.46	3.13	43.78
15	-4.15	1.14	-5.44	7.48	-5.27	-1.52	-3.74	4.18	-4.33	0.00	0.00	-14.04
16	5.03	10.42	5.41	-2.71	-0.32	0.32	2.67	-0.24	-2.89	-18.43	3.05	-26.57
17	-4.60	-5.12	NA	NA	-0.52	-11.01	-5.51	20.57	-6.91	-37.01	11.64	-40.60
18	-0.75	2.93	0.41	1.62	-0.51	1.96	0.67	5.01	-8.57	15.95	4.17	-48.62
19	-1.85	-2.30	1.23	0.92	2.82	-2.51	0.07	0.29	-4.29	35.17	10.01	-15.55
20	-2.05	4.53	-0.40	3.23	0.79	0.79	-1.64	1.88	-6.06	5.78	6.00	-25.56
21	-4.15	-8.69	0.77	1.55	-4.79	-6.05	6.32	0.86	-4.27	84.05	5.73	53.70
22	1.81	4.90	3.73	-3.33	0.89	1.71	1.83	-1.11	0.00	-15.95	17.17	-22.60
23	-1.41	4.76	-0.34	0.00	-2.32	-1.12	3.74	-7.78	-2.99	30.07	0.00	7.43
24	0.31	1.24	0.15	-1.06	-1.35	4.07	-2.36	0.76	-3.01	-19.23	22.72	-22.27
25	-2.26	-0.08	0.22	0.07	-0.39	0.77	-1.08	0.81	0.00	-78.69	8.33	-20.14
26	-0.23	2.09	-0.68	5.03	-0.33	1.74	0.56	6.13	-12.08	11.06	10.12	0.84
27	-7.31	0.76	NA	NA	-4.03	0.16	1.17	0.29	-4.31	-80.18	16.41	-24.89
28	-4.50	0.48	-4.37	7.32	-1.12	-6.66	-3.03	7.41	-9.40	13.81	3.11	70.97
Avg	-2.43	0.66	-0.09	2.45	-1.36	-0.89	-0.55	3.22	-6.04	X	7.12	X
Median	-2.05	1.14	0.30	1.59	-0.51	0.16	0.56	1.37	-4.33	X	5.90	X
Stdv	3.47	4.73	2.89	3.92	3.91	5.44	3.88	6.42	5.55	X	5.65	X

Table 3.7. Direction error of original 4 N experiments

Grid Location	(4, 0, 0)	(-4, 0, 0)	(0, 4, 0)	(0, -4, 0)	(0, 0, 4)	(0, 0, -4)
13	19.65	21.03	12.32	43.77	14.83	23.94
14	29.18	19.06	12.16	40.69	5.45	12.61
17	12.15	14.76	8.12	25.76	27.26	21.86
27	7.74	16.70	8.24	7.96	11.00	22.00
Average	17.18	17.89	10.21	29.55	14.64	20.10
Median	15.90	17.88	10.20	33.23	12.92	21.93
Stdv	9.39	2.74	2.35	16.40	9.26	5.08
Max	29.18	21.03	12.32	43.77	27.26	23.94

Table 3.8. Direction error of repeated 4 N experiments

Grid Location	(4, 0, 0)	(-4, 0, 0)	(0, 4, 0)	(0, -4, 0)	(0, 0, 4)	(0, 0, -4)
13	6.21	3.93	4.97	5.36	3.22	5.99
14	4.02	11.77	8.07	12.39	0.71	9.29
17	2.08	6.59	3.67	12.34	3.93	15.05
27	6.79	6.47	4.26	5.53	2.74	8.68
Average	4.78	7.19	5.24	8.91	2.65	9.75
Median	5.12	6.53	4.62	8.94	2.98	8.99
Stdv	2.16	3.29	1.96	4.00	1.38	3.81
Max	6.79	11.77	8.07	12.39	3.93	15.05

Table 3.9. Magnitude error of original 4 N experiments

Grid Location	(4, 0, 0)	(-4, 0, 0)	(0, 4, 0)	(0, -4, 0)	(0, 0, 4)	(0, 0, -4)
13	-0.83	1.80	-0.21	-0.83	-0.65	1.70
14	-0.90	1.87	0.66	-0.58	0.13	0.30
17	-0.43	-0.38	0.12	-0.35	-0.04	-0.36
27	-0.03	-0.29	-0.28	-0.84	-0.23	-0.12
Average	-0.55	0.75	0.07	-0.65	-0.20	0.38
Median	-0.63	0.76	-0.04	-0.71	-0.14	0.09
Stdv	0.40	1.25	0.43	0.23	0.34	0.92
Max	0.90	1.87	0.66	0.84	0.65	1.70

Table 3.10. Magnitude error of repeated 4 N experiments

Grid Location	(4, 0, 0)	(-4, 0, 0)	(0, 4, 0)	(0, -4, 0)	(0, 0, 4)	(0, 0, -4)
13	0.21	-0.56	-0.28	-0.09	-0.59	-0.28
14	0.06	-0.05	0.04	-0.02	-0.67	0.25
17	-0.11	-0.28	-0.02	-0.59	0.07	-0.40
27	0.25	-0.34	0.26	-0.51	-0.35	-0.21
Average	0.10	-0.31	0.00	-0.30	-0.39	-0.16
Median	0.14	-0.31	0.01	-0.30	-0.47	-0.25
Stdv	0.16	0.21	0.22	0.29	0.33	0.28
Max	0.25	0.56	0.28	0.59	0.67	0.40

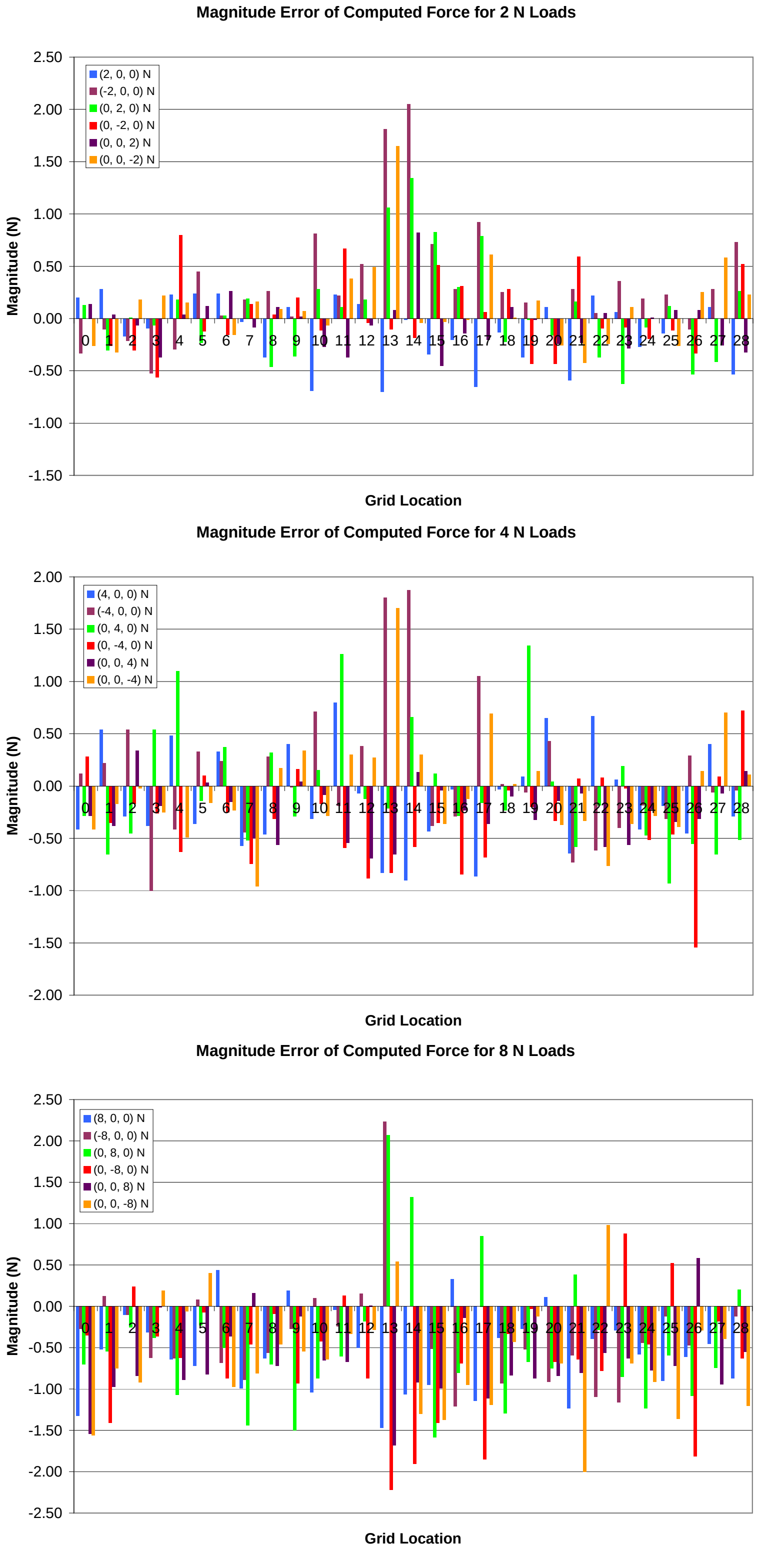
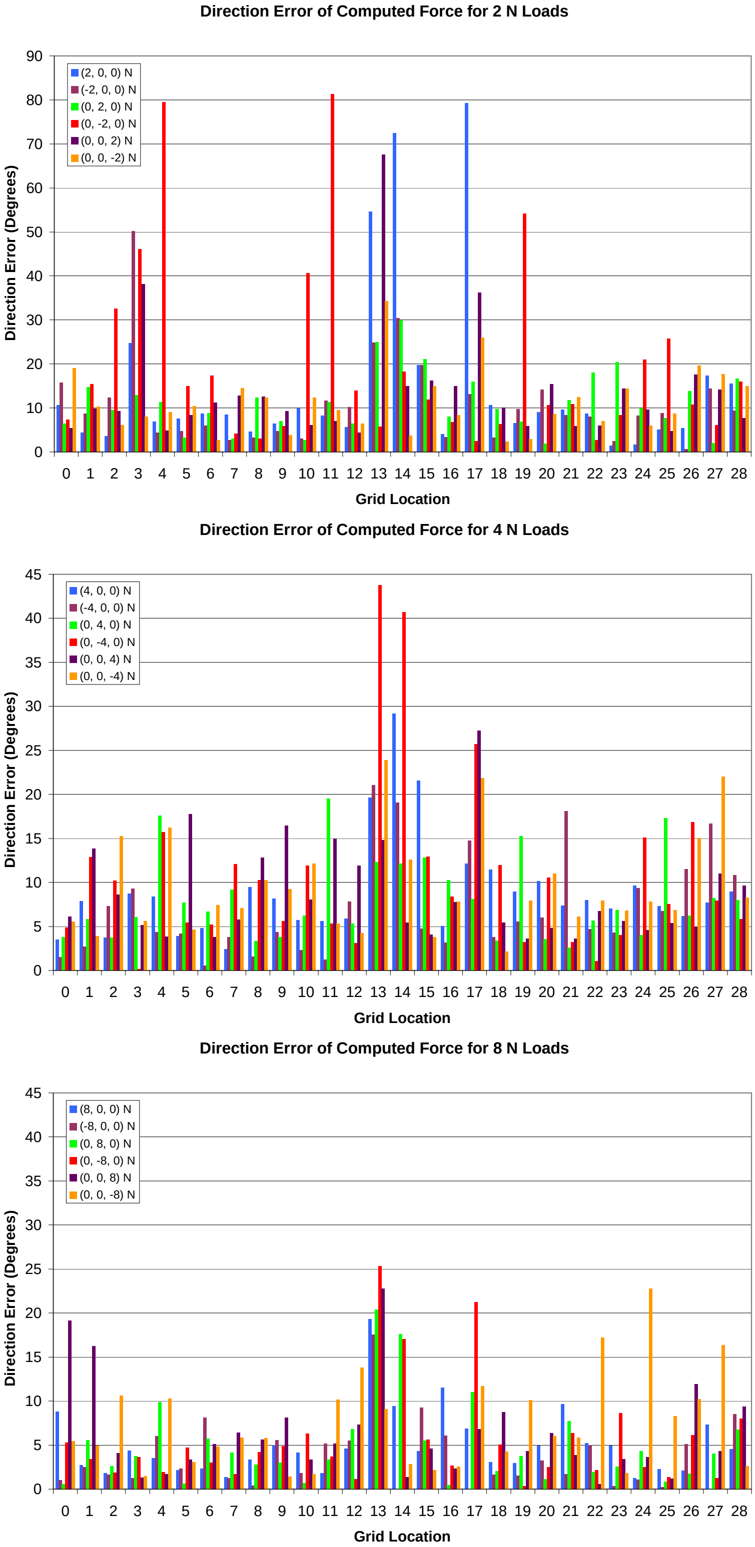


Figure 3.4. Comparison in magnitude error of computed force with different magnitudes of applied force


Direction Error of Computed Force for 2 N Loads

Direction Error of Computed Force for 4 N Loads

Direction Error of Computed Force for 8 N Loads

Figure 3.5. Comparison in direction error of computed force with different magnitudes of applied force

CHAPTER 4

KINEMATIC CONDITIONING OF ARMS

In order to detect forces and/or deflections using a kinematic device with many degrees of freedom, care must be taken to posture the device appropriately. The kinematic configuration of a robot arm has significant affects on the range and precision of these observations. The uBot-5 will be using this feedback for the end-effectors in later experiments so we kinematically condition the arms for good sensitivity to forces or velocities. Contact forces can be estimated from motor current measurements and knowledge of kinematics, or from deflections in the mechanism with known controlled impedances. There is limited resolution in sensing joint torques with motor currents and the accuracy of sensing motor current decreases with increased motor current. Encoders are equally sensitive to sensing joint displacements. The analysis of kinematic conditioning that we use assumes that each joint has equal sensitivity in sensing joint torques and displacements. This simplification gives insight into good configurations of the arms for sensing with the end-effector without taking into account individual joint sensitivity to speed or torque for the joints or knowing motor currents that counter torques on the joints due to gravity.

The analysis for the sensitivity of endpoint forces to joint torques involves looking at kinematic conditioning ellipsoids[9]. The joint torques and the kinematics determines the ability of the endpoint to exert forces. We can estimate the relative amplification of a mapping from joint torque to endpoint force by evaluating the affect of applying the manipulator Jacobian to the unit hypersphere in torque space to see how it is deformed in the output force ellipsoid. Directions in which the endpoint is poor at generating forces require more joint torques, which are ideal directions for sensing forces.

The shape of the force ellipsoid is described by the eigenvectors and eigenvalues of the matrix $(JJ^T)^{-1}$. The principal axes of the force ellipsoid are defined by the eigenvectors and relative amplification of forces in the directions of the eigenvectors are the corresponding eigenvalues. There is an inverse relation between the ability to generate endpoint forces and velocity. Directions that are good at generating endpoint forces are poor at generating endpoint velocities and directions that are good at generating endpoint velocities are poor at generating endpoint forces. Likewise, the force sensitivity and velocity sensitivity has an inverse relation. Just as the force ellipsoid is defined, a velocity ellipsoid can also be defined. The velocity ellipsoid is defined by eigenvectors and eigenvalues of the matrix JJ^T where the principal axes of the velocity ellipsoid are defined by the eigenvectors and relative amplification of velocities in the directions of the eigenvectors are the corresponding eigenvalues.

The relation between joint velocities and Cartesian velocities can be observed by the deformation of a hypersphere of joint velocities to a velocity ellipsoid.

4.1 Comparing Relative Sensitivities

There are different formulations to problems involving kinematic conditioning. Manipulability is a measure of the kinematic state of a mechanism and of being in an isotropic condition where the manipulator is able to apply velocities or forces equally well in any direction[11]. A scalar conditioning metric like manipulability can be evaluated for configurations to rank configurations for a given criterion.

Another way is to just rank configurations as better conditioned in a given direction by intersecting a conditioning ellipsoid with a vector in a given direction. Rank can increase for a given configuration and direction the further the intersection point is from the center of the ellipsoid. This is a direct way to compare the sensitivity of the endpoint to various forces or velocities.

We will be using the idea of "scoring" good sensing configurations for the end-effector to sense horizontal bumps. The uBot-5 can search mazes as Figure 4.1 shows and it is necessary/beneficial to have the arms in well-conditioned configurations where the end-effectors are able to effectively sense bumps and estimate contact normals from vertical walls as they are encountered. It is not critical that the arm configuration be sensitive to bumps from other directions so we can restrict attention to the sensitivity of detecting bumps within the horizontal plane.

4.2 Endpoint Sensitivity within Planes

There is special interest in finding the sensitivity of the endpoint to forces and velocities within planes and to ignore the sensitivity to force or velocity in directions

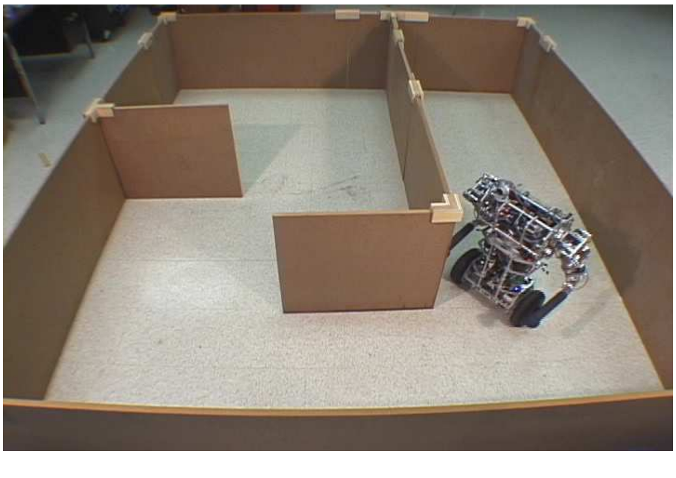


Figure 4.1. Picture of the uBot-5 navigating a maze motivating a need for kinematically conditioned arm configurations

outside the plane. Sensitivity to endpoint forces or velocities can be examined by looking at the ellipse that results from the intersection of a plane through the center of either the force or velocity ellipsoid. The intersecting points of the plane and the ellipsoid form an ellipse describing the magnitudes of forces or velocities able to be generated in directions within the plane. The major axis of the ellipse corresponds to the largest force/velocity in the plane that a unit-length input of joint torques/velocities can generate and the minor axis corresponds to the smallest force/velocity. Searching a set of arm configurations for the ellipse with the smallest major axis will find the best arm configuration for sensitivity to forces or velocities in directions within the plane.

To find the most sensitive configurations to forces or velocities within a plane the following procedure can be executed over a finite set of arm configurations:

1) Evaluate the Jacobian J
2) Compute eigenvalues and eigenvectors of JJ^T for comparing endpoint velocity sensitivity (or $(JJ^T)^{-1}$ for endpoint force sensitivity)
3) Calculate or estimate the length of the major axis of the ellipse formed when intersecting the plane with the ellipsoid defined by the above eigenvalues and eigenvectors

Actually computing the major axis of the ellipse formed when intersecting the plane with the ellipsoid would be best but for the examples shown later it was estimated. The major axis of the ellipse can be estimated with a general routine that computes the intersection point from a vector originating at the center of the ellipsoid. With this routine the length of the major axis of ellipse would be the intersection point with the longest distance in the plane to the center of the ellipsoid. A uniformly spread set of vectors in the plane can be picked and intersected with the ellipsoid. The vector that intersects the ellipsoid with the longest length can be estimated to be the length of the major axis. This estimate will get closer to the length of the major axis of the ellipse with larger sets of vectors.

4.3 Sensitivity in the Horizontal Plane

The sensitivity to horizontal forces and velocities at the end-effector is determined for a small part of the workspace of the uBot-5 arms where tilt and elbow configuration can vary but pan and twist are fixed. A few example configurations for the right arm are illustrated in Figure 4.2. Figure 4.3 shows θ_1 (tilt) and θ_2 (elbow) that can vary. This search space is chosen to find good arm configurations for the maze experiment. Since the forearm length from the elbow to the end-effector is less than the length of the upper arm and the elbow DOF is restricted to the range from -$\pi/2$ to $\pi/2$, inverse kinematics allows a unique mapping from Cartesian coordinates to the two joint angles. This is useful because an image can be created to visualize the sensitivity of the end-effector to horizontal forces or velocities. Images in Figure 4.4 show the sensitivity in the end-effector to horizontal forces and velocities. Each pixel is mapped by inverse kinematics from a (Y, Z) coordinate to a unique arm configuration consisting of θ_1 and θ_2. Intensity for the pixel is assigned to be proportional to a metric of the corresponding arm configuration or zero intensity if the arm configuration is unreachable.

Bitmaps in Figure 4.4 show metrics that were created with intensity proportional to the length of the major axes of the corresponding force/velocity ellipse formed by the intersection of a horizontal plane and the ellipsoid. So intensity is proportional to

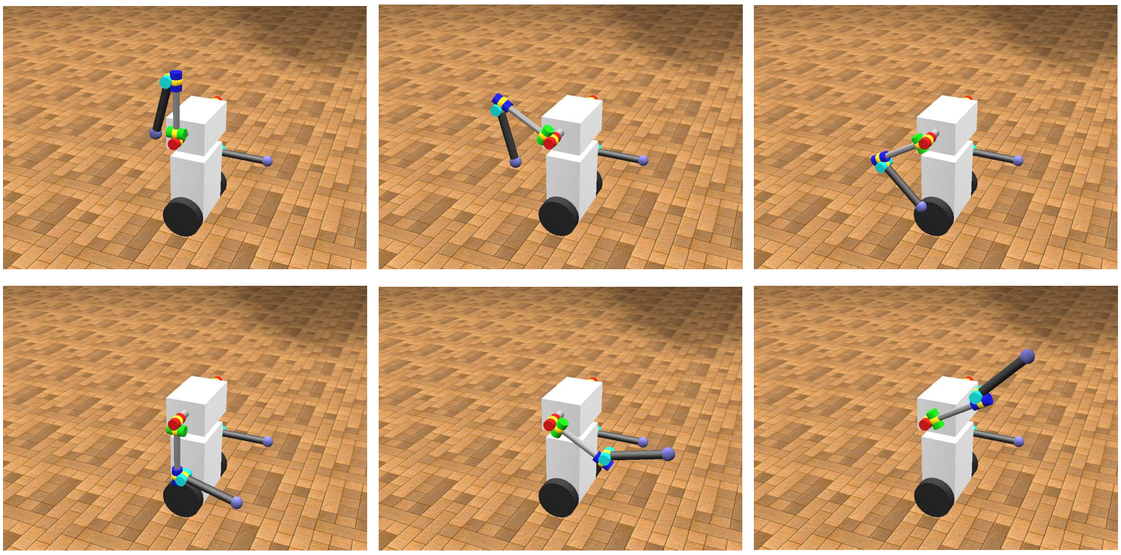


Figure 4.2. Examples of restricted joint configurations in the plane for the right arm where tilt and elbow configuration can vary but pan and twist are fixed

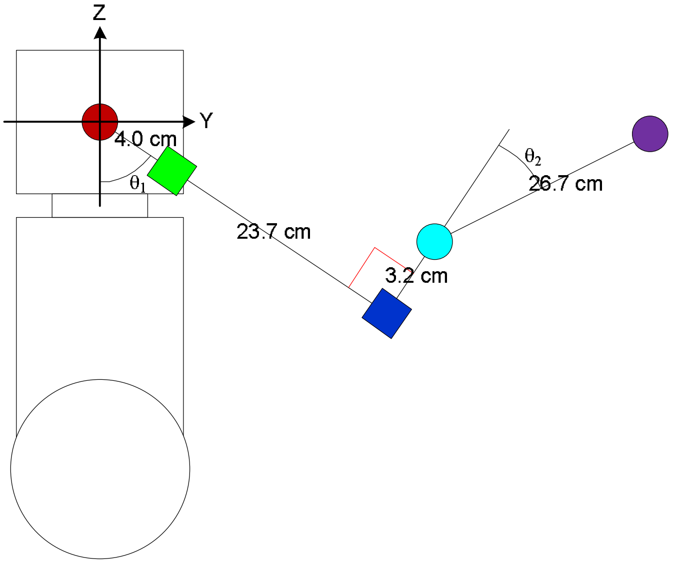


Figure 4.3. Restricted kinematics of the arm considered for sensitivity to horizontal forces and velocities

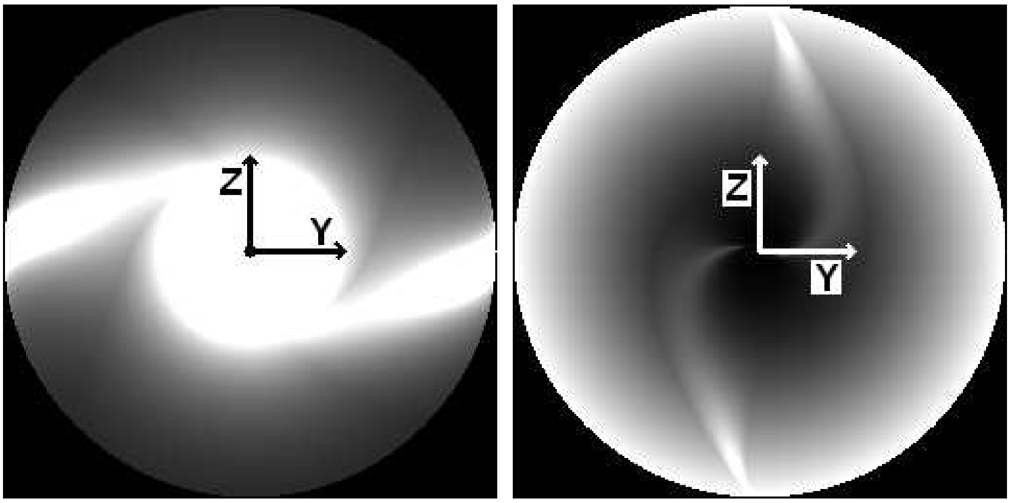


Figure 4.4. Sensitivity to horizontal endpoint forces (left) and velocities (right)

the maximum amount of force/Cartesian velocity that can be exerted by the endpoint from a unit vector of joint torques/joint velocities. It is scaled to best represent the range over all configurations considered. Brighter intensity corresponds to configurations with better maximum force or Cartesian velocity generation. Darker areas in both of the metrics correspond to better sensitivity.

Figure 4.4 shows the image on the left is darker above and below the center, so the end-effector in these configurations is only able to apply relatively small horizontally oriented forces. Configurations that generate relatively small horizontal endpoint forces imply that the configuration is better for sensing horizontally applied forces. We can see from the saturated white region that sensitivity to horizontally applied forces to the end-effector is worse when placed in front of or behind the shoulder. It is better the further above or below the shoulder it gets. The 4 darker spirals indicate particularly good configurations exist for sensing horizontal forces compared to nearby configurations.

The image to the right in Figure 4.4 shows that in general horizontal endpoint velocities are sensed better with the endpoint closer to the shoulder joint. There are two wide spirals where the sensitivity for sensing endpoint velocity is lower than other arm configurations where the endpoint is just as far from the center. Avoiding configurations where the end-effector is inside the spiral is better for sensing horizontal forces for a given distance of the end-effector from the shoulder.

The arm configuration that is best for sensing horizontal forces on the end-effector over the full range of motion in all 4 degrees of freedom in the arm is shown in Figure 4.5. The end-effector is very sensitive to horizontal forces with the arm stretched close to the ground or nearly straight upward. The optimum configuration is found by searching for the smallest major axis of all horizontal force ellipses of a discretized set of all joint configurations. The optimum configuration found is the one that can generate the least maximum horizontal force possible from the configuration.

This is an optimum configuration for sensing horizontal forces because any other configuration is able to generate more horizontal force, which would imply a less sensitive configuration to horizontal forces. The optimum configuration has a large level-arm. It takes less force with a larger lever arm to create the same joint torques so an outstretched arm is beneficial to sensing joint torques in the more proximal tilt and pan joints of the arm.

The best sensitivity to horizontal Cartesian velocities and the most sensitive configuration is shown in Figure 4.6. The velocity ellipsoid shows that the end-effector can produce large Cartesian velocities in the Z direction for unit length joint velocities but only small velocities in the X and Y directions. The major axis of ellipse resulting from the intersection of the velocity ellipsoid and a horizontal plane is the smallest for the discretized set of joint configurations. Small maximum horizontal Cartesian velocity generation corresponds to good sensitivity to horizontal Cartesian velocities. Small horizontal displacements of the end-effector will cause large joint angle displacements.

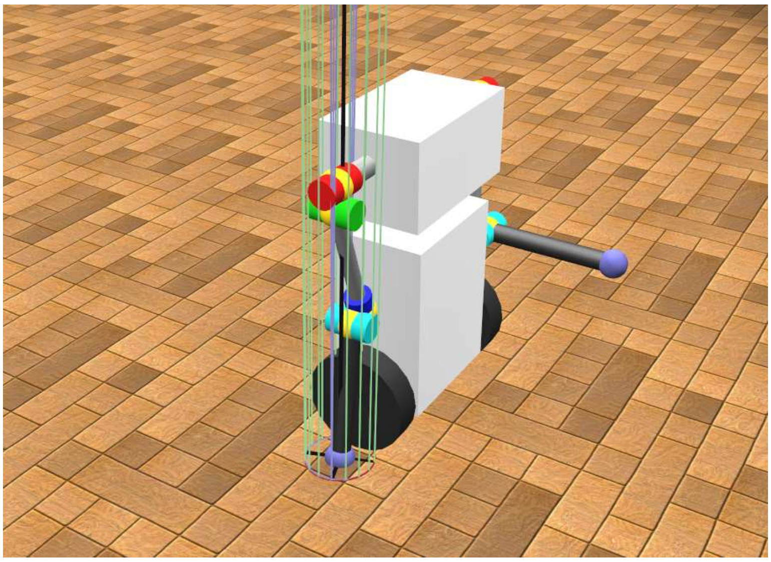


Figure 4.5. Arm configuration with best force sensitivity in the horizontal plane.

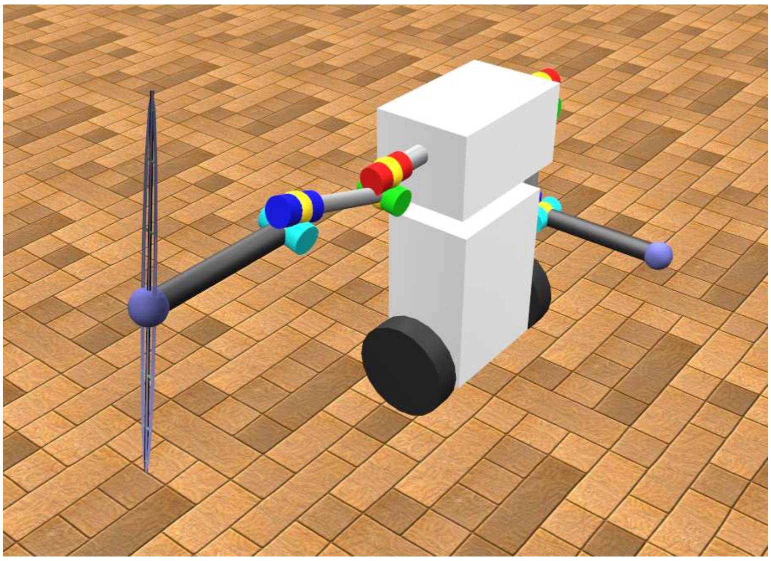


Figure 4.6. Arm configuration with best Cartesian velocity sensitivity in the horizontal plane.

CHAPTER 5

MOBILITY EXPERIMENTS

A mobile robot with arms that moves autonomously in a dynamic or unknown environment can be continuously sensing for forces or displacements of its end-effectors. Sensing like this with compliant arms can help prevent the robot from running into things causing damage to the environment or the robot. Once the presence of an unexpected entity is sensed, the robot can stop and engage in other behavior such as more closely examining it with other available sensors. In the mobility experiments, I conducted as part of the thesis, the uBot-5 navigates around obstacles using haptic feedback in a maze-like environment based on methods described in earlier chapters. These experiments are conducted in an indoor environment with a flat floor surface at a single elevation with obstacles high enough to be observed by contacts with the compliant arms. To navigate successfully, the robot still requires additional sensors in addition to endpoint forces or deflections. For instance, odometry is particularly important to detect obstructions that do not make contact with arm endpoints.

For these mobility demonstrations, the arms are put in a desired configuration and then endpoint contacts can be inferred to be from obstacles. The arms are placed in configurations that allow them to sense endpoint forces or deflections well. Mobility constraints are detected by noticing a change in the reaction force applied or a displacement in the endpoint of either arm. This experiment uses odometry to map mobility constraints sensed by arms and a harmonic function path planner to re-plan mobility strategies. Using a harmonic function, which is free of local minima, the robot moves in the direction of the steepest downhill gradient to generate collision-free paths that are guaranteed to reach the goal if a path exists. The uBot, therefore, uses as little environmental geometry as possible, extending the model only when the current path encounters an unmodeled obstacle.

5.1 Harmonic Function Path Planning

Harmonic functions are useful solutions to the path-planning problem for robotics. One of their most important properties is that they do not have local minima so following the gradient of this potential field will always lead to a reachable goal. Connolly, et al. describe the application of harmonic functions to the path-planning problem[4]. Harmonic functions are generally smooth and are complete for path planning (up to discretization error in the environment model) so it generates a path if one exists. Computing them is fast, allowing the environment model to be updated incrementally and works well even when the model is updated online. A signal that there is "no path" takes the same time required to find a path.

A function on the domain $\Omega \subset \Re^n$ is harmonic if it satisfies Laplace's Equation that requires that the trace of the Hession is zero (the sum of the second derivatives with respect to each variable in the domain is zero). For robot path planning the boundaries of this function consist of the boundary conditions at the edges of the environment, obstacles, and goals. The spontaneous creation of local minima within the harmonic function region is impossible and from anywhere in freespace there is a path to a goal[4]. Numerical relaxation methods are used to solve Laplace's Equations on a regular grid (i.e. Jacobi iteration, Gauss-Seidel iteration, or Successive Over-Relaxation (SOR)). SOR converges with fewer iterations than the other two methods so it was chosen in this application. SOR defines the following recurrence relation:

$$u^{k+1}(x_i, y_j) = u^k(x_i, y_j) + \frac{\omega}{4}(u^{k+1}(x_{i+1}, y_j) + u^{k+1}(x_{i-1}, y_j) +$$

$$u^k(x_i, y_{j+1}) + u^k(x_i, y_{j-1}) - 4u^k(x_i, y_j)),$$

where ω is the overrelaxation constant. In the above equation, ω was set to 1.9 in all the experiments. Grid potential values $u(x_i, y_j)$ marked as freespace get updated each k-th iteration until a minimum residual threshold is achieved.

Boundary conditions for obstacles can use Dirichlet or Neumann boundaries. A Dirichlet boundary holds the boundary at a fixed high potential so that a robot close to an obstacle will depart along the outward normal of the obstacle. Neumann boundaries establish a zero gradient normal to object boundaries so that no flow violates an obstacle boundary. Using Neumann boundaries results in the robot moving tangentially along obstacles. Dirichlet boundary conditions imply minimum hitting probability, which is quite useful in cases where there is significant ignorance about environmental geometry. Neumann boundaries do not have this property. Following closer to obstacles can be a useful exploratory behavior but is more likely to result in an undesired collision so Dirichlet boundary conditions were used in the experiments.

A starting location for the robot is estimated to the nearest grid point, which is kept at a high potential and is not freespace. The harmonic function is computed with SOR whenever new obstacles or goals are added. After the harmonic function has converged to within a given tolerance, bilinear interpolation is used to create a continuous function. A search is done with the computed harmonic function in the neighborhood from the current location of the robot to find the steepest "downhill" gradient, which becomes the new reference direction that the robot tries head in. If the robot heads in slightly the wrong direction the robot will just follow another streamline path to the goal.

5.2 Implementation

The environmental model used in the implementation consists of a 2-dimensional occupancy grid to store the harmonic potential field, a two bit label for each grid location indicating freespace, obstacle, or goal, the physical grid width and height, and the size of the grid array. The robot can start at any grid location not on the border and can start facing any direction. Given the command to begin, the robot will assign its current position and orientation as the origin and use odometry to measure/integrate position changes.

The local task space of the robot is all possible positions and orientations of the robot in the modeled area. Obstacles are configurations in this space that result in collision. When the robot senses a bump, obstacles are added to the occupancy grid. Obstacles boundaries are dialated so as to reduce the dimension of the robot to a point. Elliptical obstacles are appropriately dialated for the uBot-5 with the additional information of the estimated contact normal from the bump event.

Elliptical obstacles are added to the occupancy grid. The minor axis of the ellipse is along the contact normal and the major axis of the ellipse is tangential to the contact normal. Sizes of the major and minor axes were chosen to be smaller than the radius of the uBot-5. The major axis was chosen to be long enough so that bumps along straight walls overlapped enough so there was not a gap in between. The thickness, or the size of the minor axis, was large enough to avoid having the robot touch walls.

In the maze traversal demo, the uBot-5 was kept along a stream-line to the goal by higher-level controllers. A velocity mode controller was used move the uBot-5 with a reference velocity and turnrate. In general, the uBot-5 travels along a “stream-line” with constant desired speed towards the goal with a desired turnrate proportional to the error in heading. If the error in heading becomes too large then a zero reference velocity is given with a constant turnrate to turn in place until the error in heading is small again. When a bump is sensed, the robot backs up a fixed distance along the gradient towards the high potential before inserting the sensed obstacle. The uBot-5 may drift backwards to maintain balance because the end-effector presses harder on an object if it were to only turn in place. Control over backing up was preferable to having no control when an obstacle is encountered and the robot needs to turn. The environment could be delicate, so the end-effectors should not be pressing or sliding against it more then they have to. For smoother acceleration, desired reference speeds were ramped up linearly over time. More work is required improve path tracking accuracy.

The environment and state of the world, as the robot sees it, is displayed to a robot operator. Initially nothing is known of the environment but the user is able to select grid size/resolution of the harmonic function and starting position/orientation of the robot. Each grid point is colored to easily identify whether the area represented by the grid point contains free space, an obstacle, a start location, or a goal location. The start location can be anywhere in the environment and multiple goals can be placed. When the operator is ready to have the robot attempt to reach the goal, they click the “run” button.

5.2.1 Endpoint Sensing Configurations

The best arm configuration for sensing forces in the horizontal plane was found in Section 4.3. The domain over which a good bumper arm configuration was determined included tilt and elbow angles when pan and twist are fixed at zero. Further restricting the domain of arm configurations is required if the robot is to balance by requiring the endpoint to be at least a certain distance above the ground. In order to bump things while traveling forward the endpoint should be placed in front of the body so the endpoints hit an obstacle before the body. The result placed the arms quite a bit forward of the chassis so another constraint was used to restrict the endpoint to be

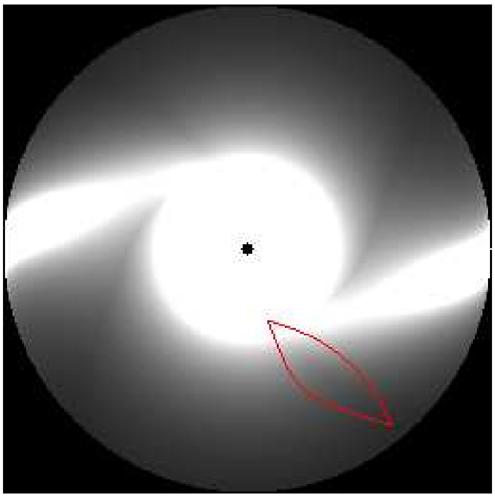


Figure 5.1. Highlighted region shows configurations good for sensing horizontal endpoint forces and velocities (Y-axis is to the right and the Z-axis is up

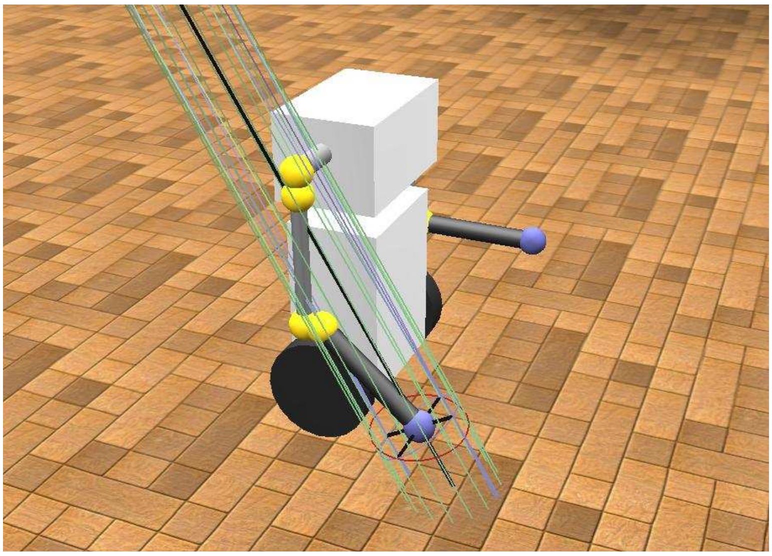


Figure 5.2. Arm configuration with best force sensitivity in the horizontal plane for bumping obstacles.

within a certain range of the body. The resulting configuration is shown in Figure 5.2.

In addition to sensing endpoint forces, endpoint position deflections can be used to detect bumps. A hybrid configuration can be found that is good at sensing both position deflections and forces in the horizontal plane. Figure 5.1 shows a highlighted area from Figure 4.5 that has a balanced sensitivity for both of the methods of sensing bumps. The area was highlighted by hand because it is difficult to quantitatively compare the force and velocity sensitivities because of the different units. The highlighted region contains the darker area, which is more sensitive to forces than nearby surrounding configurations. It is near the bottom for better force sensitivity, but at the same time, close to the center for better displacement sensitivity. It is uncertain where the best configuration really is, but I expect that the area illustrated strikes a good balance with respect to sensitivity to force and velocity.

5.2.2 End-Effector Compliance

The environment in which the robot operates can be damaged if it is encountered with stiff arms. Both arms, the robot, and environment will experience impact force when the end-effectors encounter an obstacle. End-effector compliance reduces impact force, which is directly proportional to mass and velocity of the robot:

$$f = ma = \frac{d}{dt}(mv)$$

The reaction forces are directly proportional to the amount of mass that is subject to the change in velocity—the goal is to minimize “the spring mass” as in the design of an automobile suspension. Impact force from a stiff arm would be influenced by the mass of the whole robot whereas with a compliant end-effector the impact force can be much less. A dynamically balancing robot can have the arms swing backwards while the base continues to move forward for some amount of time. This can reduce the effective mass involved in the impact since the mass of the arms can be much lower than that of the whole platform.

The way that compliance is added to the end-effector of the uBot-5 is by attaching virtual springs at the end-effector of each arm. In addition to the torque required to counter gravity and hold the arm in place, torque is added to each joint in the arm in order to apply an endpoint force. Using the formula relating endpoint forces to joint torques ($\tau = J^T f$) there is an attractive force that pulls the endpoint toward a desired Cartesian point. When the arms have no velocity or acceleration and the end-effector is compliant the joint torques required to counter gravitational force are just the current joint torques. Acceptable compliant arms were achieved, with the torques due to gravity kept constant arms, even though the torques to counter gravity change as the arm is displaced.

Additional endpoint force is added to the end-effector with the above formula to apply additional force from the virtual springs. The setup of springs used one radial spring in the horizontal plane and a nonlinear spring in the vertical direction with a higher spring constant for deflections in the downward direction. The virtual springs exert a force of -Kx where K is the spring constant and x is the endpoint displacement relative to the equilibrium point. The radial spring constant is low to have greater endpoint compliance in the horizontal plane. If the endpoint is above the desired endpoint position then a low spring constant is used for the vertical spring to keep it from floating up too high and if the endpoint is lower than the desired endpoint position then a high spring constant is used for the vertical spring to keep the end-effector from dropping due to gravity much lower than the desired endpoint position.

End-effector compliance required writing reference torques to the motor controller. The torque controller was written on top of a position mode PD controller using motor current. The position PD controller contains a damping term so no damper was added to the virtual springs to prevent oscillations. With too high of spring constants there were oscillations and a damper could probably help to eliminate them. The damper could exert a force to counter force from the spring that is velocity dependent.

5.3 Results

Figure 5.3 shows side-by-side video frames and the occupancy grid as it is being built while the uBot-5 navigates to a destination in the upper left corner of the maze. The grid is 75 × 75 and covers a 5 m × 5 m area. The green square shows the starting position, a single goal is shown by a red square, squares that contain known obstacles are colored black, and newly discovered obstacles are colored brown (in order for new obstacles to get inserted into the occupancy grid the robot travels backwards a fixed distance). A stream-line from the current position to the goal is shown in blue.

The obstacles inserted have approximately the correct contact direction. The endpoints were compliant and contact direction was found using endpoint displacements that appeared to be more accurate than sensing contact direction using endpoint force. Odometry was consistently accurate enough to get to the final room. Hitting walls with compliant endpoints did not cause any noticeable slipping in the wheels.

Arms were spread slightly outwards (twist was set to bring the endpoint further outwards by 0.1 radians) from an arm configuration chosen to be within the highlighted region shown in Figure 5.1. Moving the arms slightly outwards the robot is less likely to slide along the wall with the shoulder since the endpoint will hit first.

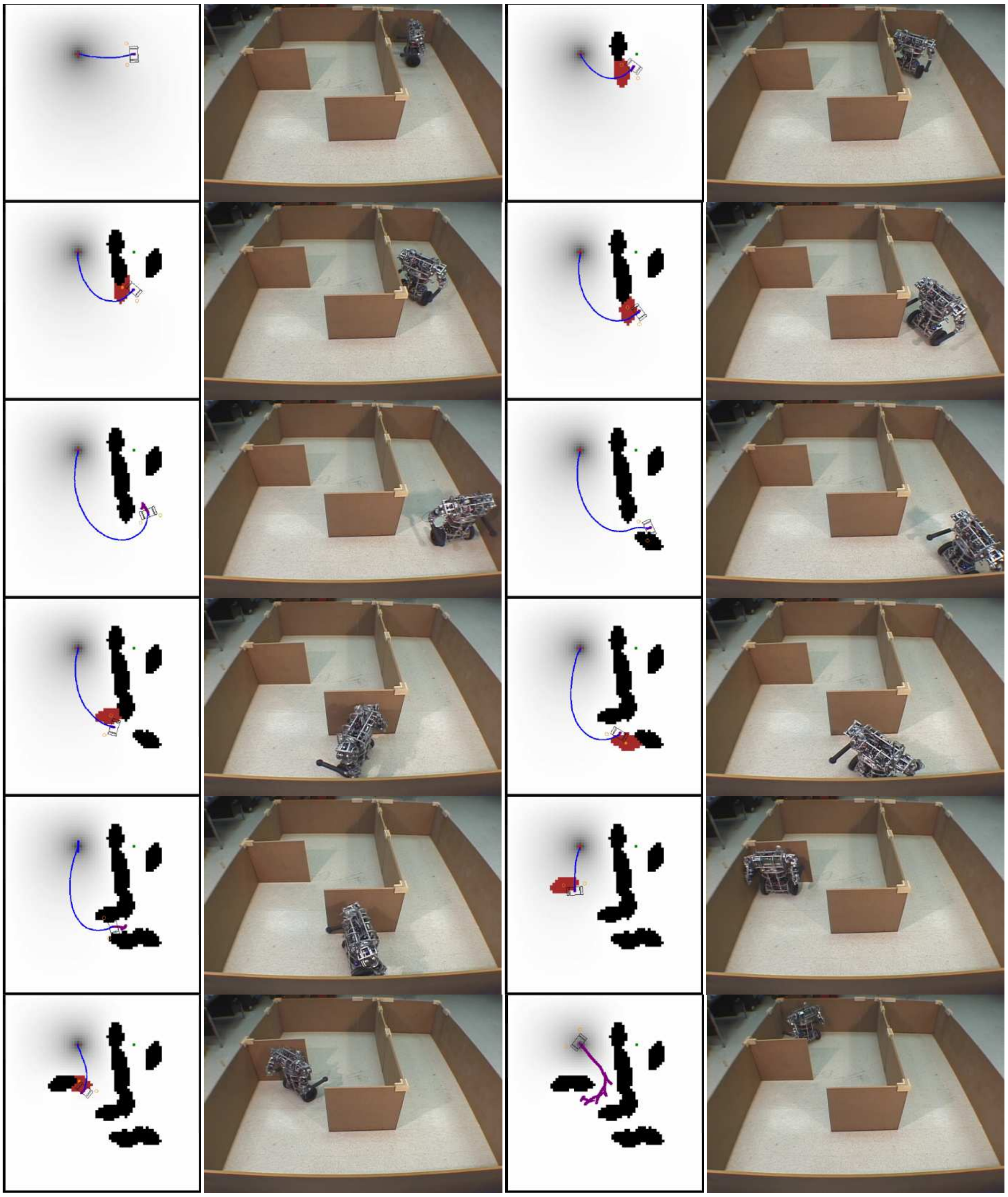


Figure 5.3. Constructed environment map and pictures of the maze demo

CHAPTER 6

EVALUATION OF GRASPING

The uBot-5 has spherical rubber end-effectors able to do a bimanual grasp on objects by closing its two arms and is able to lift objects due to friction caused by the grasp forces. It is difficult to teleoperate the robot to achieve the precise geometries required by grasping tasks. Autonomous grasping can be far better if sensor feedback is good since the robot can actively achieve the force conditions comprising a grasp while suppressing geometric disturbances. In this chapter, an autonomous grasping algorithm using contact normals computed from a probing technique is given and implemented on the uBot-5. The results of grasping have been successful, even with high friction at the end-effectors, for grasping objects with different diameters. We describe the details of how it works and the limitations encountered.

6.1 The Grasp Controller

The grasp controller developed works on convex objects where the geometry is not known (a priori) by moving the end-effectors at their initial heights in a horizontal plane towards force closure. A force closure grasp can be controlled to reject bounded disturbance forces to achieve a stable equilibrium. Equilibrium means that there are no net forces or moments on the object, there is no slippage between the end-effector and object, and external force on the object can be resisted. The grasp synthesis process brings the grasp configuration closer to the force closure condition. This process probes an object for contact normals and positions using the end-effectors

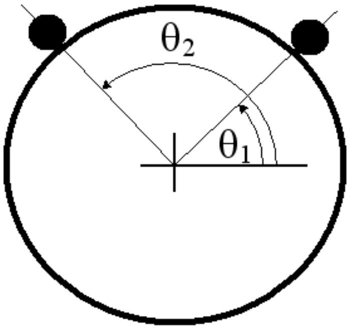


Figure 6.1. Grasp controller configuration

and then repositions the end-effectors for probing again if grasping the object would not result in the desired force closure grasp.

A contact wrench for a contact can exert force and moments at the point of contact. The grasp controller assumes point contacts without friction that apply unit magnitude forces along the inward normal. Consider a contact configuration in the plane parameterized by coordinates θ_1 and θ_2 as in Figure 6.1. The contact wrench $\boldsymbol{\omega}_i$ for contact point i is transformed into an object-frame wrench using the mapping:

$$\boldsymbol{\omega}_i = \begin{bmatrix} f_x \\ f_y \\ m_z \end{bmatrix} = \begin{bmatrix} -cos(\theta_i) \\ -sin(\theta_i) \\ 0 \end{bmatrix}.$$

The wrench residual $\boldsymbol{\rho}$ is the net wrench over k contacts:

$$\boldsymbol{\rho} = \frac{1}{k}\sum_{i=1}^{k}\boldsymbol{\omega}_i = \frac{1}{2}\begin{bmatrix} -cos(\theta_1) - cos(\theta_2) \\ -sin(\theta_1) - sin(\theta_2) \\ 0 \end{bmatrix}$$

We construct a grasp controller by descending the squared wrench residual. In some cases, this corresponds to grasps where contact forces can be controlled to yield force closure[15][3]. The grasp controller uses contact force feedback to compute differential displacements in the contact coordinates that minimize the squared wrench residual ϵ:

$$\begin{aligned}
\epsilon &= \boldsymbol{\rho}^T\boldsymbol{\rho} = \left(\frac{1}{k}\sum_{i=1}^{k}\boldsymbol{\omega}_i\right)^T\left(\frac{1}{k}\sum_{i=1}^{k}\boldsymbol{\omega}_i\right) \\
&= \frac{1}{2}\begin{bmatrix} -cos(\theta_1) - cos(\theta_2) \\ -sin(\theta_1) - sin(\theta_2) \\ 0 \end{bmatrix}^T \frac{1}{2}\begin{bmatrix} -cos(\theta_1) - cos(\theta_2) \\ -sin(\theta_1) - sin(\theta_2) \\ 0 \end{bmatrix} \\
&= \frac{1}{4}\left[(-cos(\theta_1) - cos(\theta_2))^2 + (-sin(\theta_1) - sin(\theta_2))^2\right] \\
&= \frac{1}{4}\left[(cos(\theta_1) + cos(\theta_2))^2 + (sin(\theta_1) + sin(\theta_2))^2\right]
\end{aligned}$$

The Jacobian of the force residual is a 1×2 matrix such that $\Delta\epsilon = J^F_{1\times 2}\Delta\boldsymbol{\theta}_{2\times 1}$ and is:

$$J^F = \begin{bmatrix} \frac{\partial\epsilon}{\partial\theta_1} & \frac{\partial\epsilon}{\partial\theta_2} \end{bmatrix}_{(1\times 2)}$$

where

$$\begin{aligned}\frac{\partial\epsilon}{\partial\theta_1} &= \frac{1}{4}\left[2(cos(\theta_1)+cos(\theta_2))(-sin(\theta_1))+2(sin(\theta_1)+sin(\theta_2))(cos(\theta_1))\right]\\ &= \frac{1}{2}\left[(sin(\theta_1)+sin(\theta_2))cos(\theta_1)-(cos(\theta_1)+cos(\theta_2))sin(\theta_1)\right]\end{aligned}$$

$$\begin{aligned}\frac{\partial\epsilon}{\partial\theta_2} &= \frac{1}{4}\left[2(cos(\theta_1)+cos(\theta_2))(-sin(\theta_2))+2(sin(\theta_1)+sin(\theta_2))(cos(\theta_2))\right]\\ &= \frac{1}{2}\left[(sin(\theta_1)+sin(\theta_2))cos(\theta_2)-(cos(\theta_1)+cos(\theta_2))sin(\theta_2)\right]\end{aligned}$$

We want to take steps in the direction that will minimize the squared residual force. This is done by finding the pseudoinverse of the Jacobian of the residual force, which for the 1×2 force residual Jacobian matrix is:

$$\begin{aligned}(J^F)^{\#} &= (J^F)^T\left[J^F(J^F)^T\right]^{-1}\\ &= \frac{1}{\begin{bmatrix}\frac{\partial\epsilon}{\partial\theta_1} & \frac{\partial\epsilon}{\partial\theta_2}\end{bmatrix}\begin{bmatrix}\frac{\partial\epsilon}{\partial\theta_1}\\ \frac{\partial\epsilon}{\partial\theta_2}\end{bmatrix}}\begin{bmatrix}\frac{\partial\epsilon}{\partial\theta_1}\\ \frac{\partial\epsilon}{\partial\theta_2}\end{bmatrix}\\ &= \frac{1}{\left[(\frac{\partial\epsilon}{\partial\theta_1})^2+(\frac{\partial\epsilon}{\partial\theta_2})^2\right]}\begin{bmatrix}\frac{\partial\epsilon}{\partial\theta_1}\\ \frac{\partial\epsilon}{\partial\theta_2}\end{bmatrix}\end{aligned}$$

New reference positions of the contacts are computed that move the endpoint in directions that reduce the remaining force residual using the formula $\Delta\boldsymbol{\theta}_{(2\times1)} = J^{\#}_{(2\times1)}\Delta\epsilon$. A small negative epsilon value is chosen appropriately to descend the squared force residual function at a rate that requires a reasonable number of probes. Direction is actually what is important in the grasp controller so that small tangential displacements to the contact normal in the direction that minimizes the remaining force residual is sufficient.

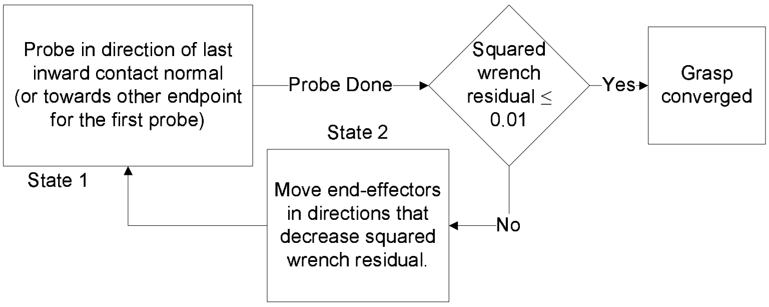

Probe in direction of last inward contact normal (or towards other endpoint for the first probe)
State 1
Probe Done
Squared wrench residual ≤ 0.01
State 2
Yes
Grasp converged
No
Move end-effectors in directions that decrease squared wrench residual.

Figure 6.2. Grasp controller flowchart

6.2 Implementation

6.2.1 Grasp Controller

A flowchart of the grasp controller is shown in Figure 6.2 and the motion of an end-effector in Figure 6.3. The grasp controller begins by simultaneously moving the end-effectors towards each other for probing an object. Probing moves the end-effector in the desired direction to get a contact normal and position. The calculations from the previous section are performed and if the squared wrench residual is small enough then the grasp has converged. If the grasp controller has not converged then the desired endpoint displacements are calculated and prescribed to endpoint position controllers. Once the displacements are close to being achieved, the grasp controller resumes probing in the direction of the last inward contact normal.

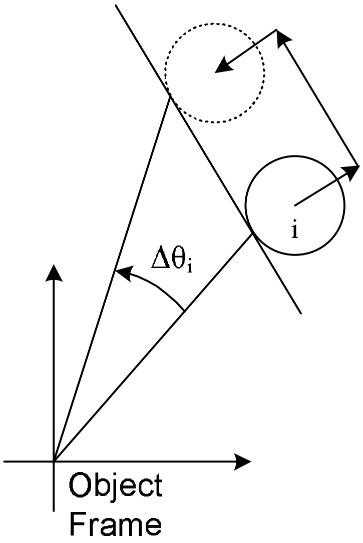


Figure 6.3. Differential displacements of an end-effector

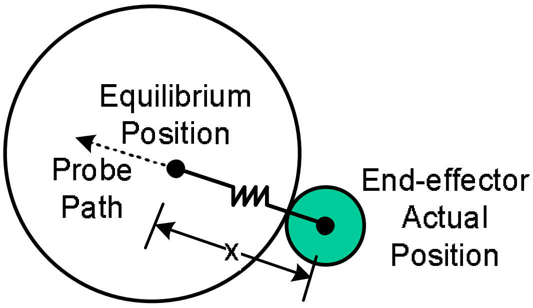


Figure 6.4. Probing using force control

6.2.2 Probing with Force Control

Implicit force control is used to probe for contact normals and positions. The technique is described and works for probing nearby objects for contact normals. Probing utilizes compliance in the end-effector similar to how it was used in the maze demo. The probing controller developed moves the end-effector along a probe path as shown in Figure 6.4 with a virtual spring by moving an equilibrium point at constant speed until the distance between the equilibrium point and actual position have exceeding some threshold, which was set to 2 cm. Accurate estimates of joint torques due to gravity are obtained with motor torque constants, motor current, and overcoming stiction as described in Section 3.2 close to where the probe is expected to make contact. In addition to applying constant joint torques to overcome gravity, additional torque is added to the joints to have the end-effector apply a force from the virtual spring. A virtual spring is attached from the actual end-effector position and an equilibrium point, applying force according to Hooke's Law where x is the distance between these points.

When the equilibrium point of the end-effector moves into an object's surface, reaction loads from the object cause deflection in the compliant controller. The force the end-effector exerts is proportional to the distance x in the direction from the actual effector position towards the equilibrium position. This force caused by the spring will have the end-effector seek minimum energy configurations. Friction between the end-effector and the object as well as stiction in the joints cause the end-effector to come to rest early. Direction of an estimated contact normal is towards the equilibrium point from the actual position but this is not accurate when there is friction since the end-effector comes to rest before reaching a minimum energy configuration.

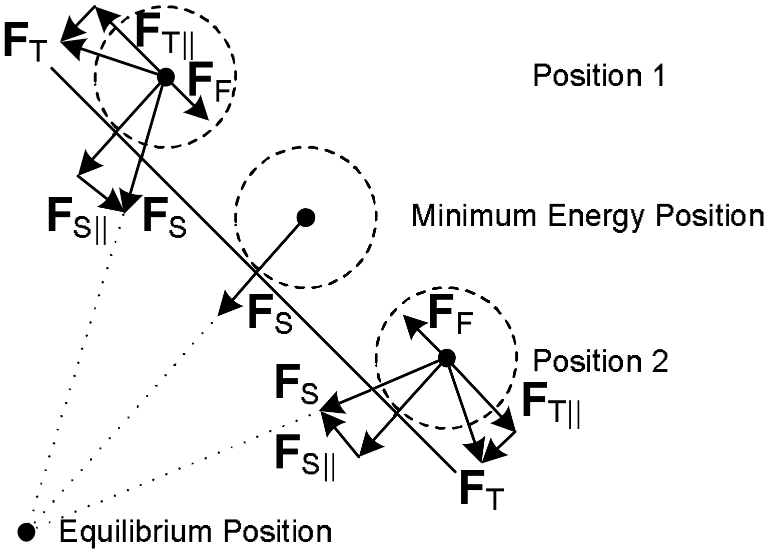


Figure 6.5. Probing technique that applies tangential force so the end-effector reaches two extreme positions

Fortunately, an active probing technique allows contact normal and position estimation even with friction. The technique expands on the idea of probing and seeking minimum energy configurations with the end-effector but since friction is significant we know the end-effector often cannot attain one. Applying force to the end-effector can cause it to slide, which can help move it towards or away from minimum energy configurations. Our technique applies sufficient force to the end-effector to cause it to slide along the object's surface that reach two extreme positions positions away from the minimum energy configuration as shown in Figure 6.5. Tangential force, relative to the vector from the position of the equilibrium point to the actual end-effector position, is applied to the end-effector to have the end-effector reach two extreme positions in the same way. The tangential force needs to overcome the force of friction between the end-effector and object and the stiction in the joints. The end-effector needs to settle (come to rest) after applying tangential force.

Tangential force applied to the end-effector ramps up and down linearly over a constant amount of time of about a second to a specified magnitude, which was 7 N, three times. Only two times are required but the first time tangential force is applied the end-effector it could move away from a minimum energy configuration. After each tangential force is applied in a direction, it is applied in the opposite direction. After the second and third tangential forces have been applied the actual position of the end-effector is recorded.

An estimated contact position is the average position of the recorded end-effector positions. The estimated contact normal is the normalized vector from the estimated contact position to the equilibrium point. After the probe controller computes both of these, the equilibrium point is moved in the direction of the actual position the end-effector started from until the actual position gets within a certain distance of the initial position.

6.3 Results

Grasping was performed successfully on a bucket about 8 inches in diameter that had a mass of 1.88 kg. This grasping experiment was performed with the uBot-5 not balancing, placed in a stand as rigidly as possible, and with the torso degree of freedom fixed. Pictures were taken of a successful grasp that started with the end-effectors a few of centimeters away from the surface of the bucket and the grasp controller probed the bucket until a grasp configuration with low enough squared residual force was found. To demonstrate the bucket could be picked up successfully, the end-effectors applied force on the bucket to squeeze it and then lifted it up several centimeters. Squeezing and lifting the bucket both used implicit force control of the end-effectors using the idea of virtual springs attached to the end-effector. The equilibrium position of the virtual spring for each end-effector was moved towards each other to squeeze the bucket and then raised to lift it.

Table 6.1 shows the grasp configuration of the end-effectors as they probed the bucket each time and the squared residual force. Pictures in Figure 6.6 show the results as probing proceeded. The first picture is the initial configuration where the

Table 6.1. Grasp configuration and squared wrench residual for probing a bucket

Probe Num	θ_1 (deg)	θ_2 (deg)	ϵ
1	201	327	0.204
2	212	327	0.286
3	197	332	0.151
4	190	337	0.083
5	188	338	0.068
6	188	341	0.056
7	182	347	0.019
8	182	350	0.013
9	178	350	0.006

end-effectors were close to the bucket but not touching it, the following 9 pictures show each step of the probing as contact was made with the bucket, and the final two each show the bucket lifted up.

6.4 Limitations of Grasping

There are several limitations that were encountered while grasping objects. Relatively large forces estimated to be between 5 to 10 N were required to be applied to the end-effector to probe objects for estimating contact normals. Force to overcome stiction in the joints was observed to be roughly 3 N. Excess force was required to cause movement in the end-effector in the desired direction and for the end-effector to slide on the surface of the object. The high force generated by probing for contact normals can cause the probed object to move, which is something that is not taken into account. The current probing method requires that objects remain fixed for probing them. This limits the usefulness of grasping most objects that the robot can lift since they will slide away.

The grasp controller implemented is not able to grasp while balancing. An inverted pendulum robot will naturally back away as it tries to maintain balance when exerting force by probing. The probing technique will need to be modified so it can work while balancing. Mobile manipulation should be able to be performed while balancing.

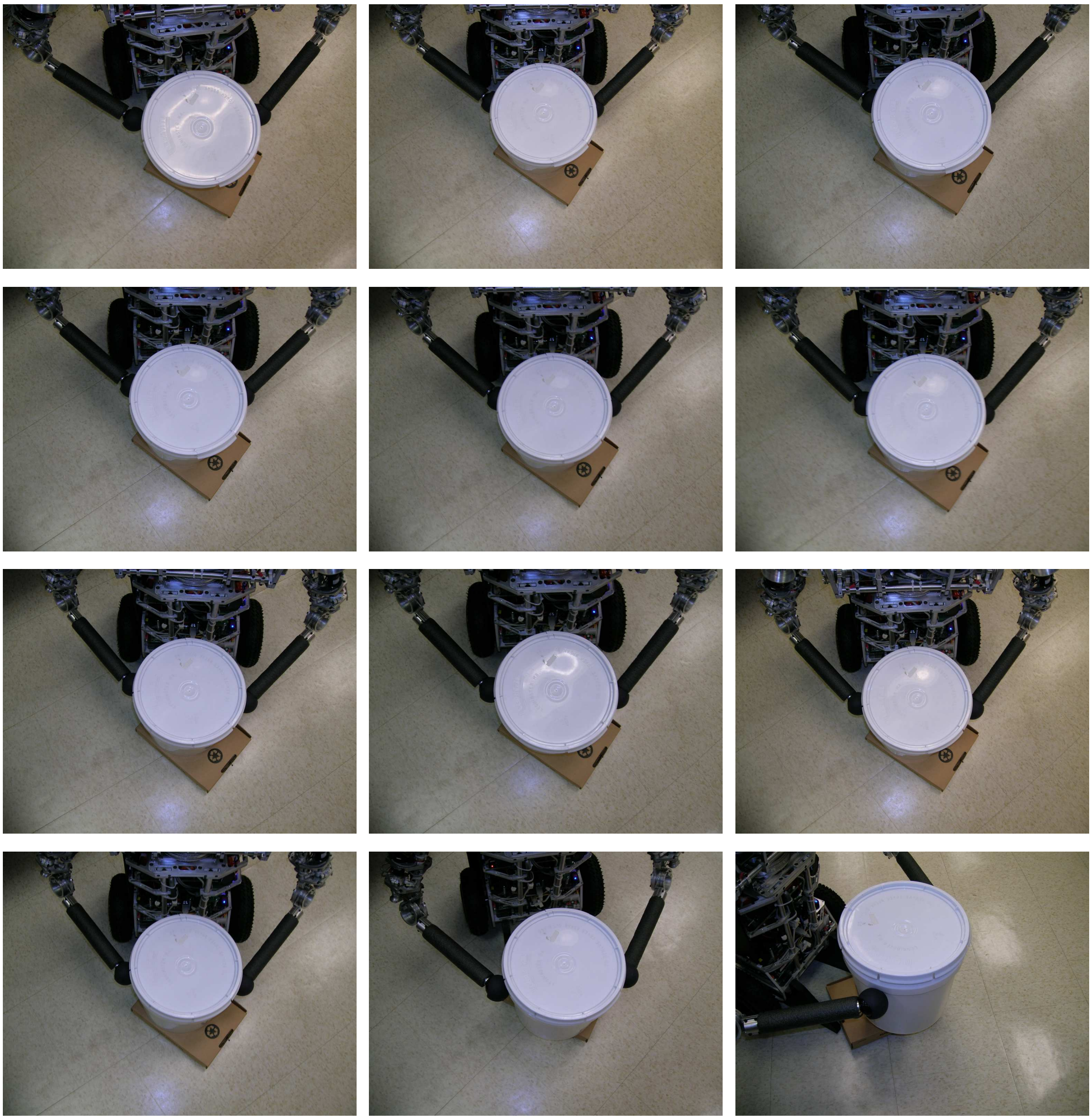


Figure 6.6. Pictures of grasping and lifting a bucket

CHAPTER 7

CONCLUSIONS AND FUTURE DIRECTIONS

The uBot-5 has been demonstrated to use endpoint force sensing in both mobility and manipulation tasks by using motor currents. A maze was navigated using the end-effectors to detect mobility constraints and cylindrical objects were lifted up after the uBot-5 formed stable grasps on them. Implicit force control was used for both demonstrations to control the impedance of the end-effectors that are expected to make contact with objects.

The end-effectors were made compliant to lessen impact forces imparted to the environment and allow sensing bumps by measuring deflections in the end-effectors so the uBot-5 to can traverse a maze by contact sensing. A reactive harmonic function path-planner was used to guide the robot in the right direction to the goal. Mobility constraints were detected by sensing contact normals of the end-effector by directly computing endpoint force applied to the end-effector or displacements of the end-effector.

Autonomous grasping has been demonstrated with the uBot-5 to grasp and lift buckets, which works with friction at the end-effectors. The grasping algorithm is general enough to form grasps on convex objects but the probing technique for contact normals will likely not work well with objects that are not cylindrical shaped. Limitations of grasping were mentioned in the last chapter.

Accuracy of computed endpoint force was found experimentally by placing one arm in various configurations and applying known reference forces of different magnitudes in different directions. Errors in computed magnitude and direction decrease with increased reference force. The tests with 8 N reference loads had the least maximum direction error of less than 30 degrees for all configurations and directions. Accuracy decreases significantly when switching to configurations where the change in motor current is high. This is probably caused by changes in temperature of the H-bridge amplifier, which has integrated current sensing and/or changes in motor temperature. Accuracy is gained by waiting for a long period of time (30 seconds or so) before motor current is tared, which should not be required.

There is significant stiction in the gear trains that causes accuracy problems in sensing endpoint forces. Stiction can be reduced using the method in Section 3.2. The routine takes a significant amount of time to “wiggle” the arm. Sensing joint torques after the gearbox would eliminate stiction affects from the gearbox that had to be dealt with by running this routine on the arms. Series elastic actuators seem to have desirable properties for joint torque sensors and look like they will be simpler than using strain gauges. The uBot-5 lacks real force control because joint torque loops are being closed using motor models and control of motor currents. Hardware revisions to the uBot-5 should consider these joint torque sensors.

Mobile manipulators should be able to react to external forces and sense forces that it applies. In the future, robots like the uBot-5 will probably need better force sensing than motor currents provide. Although the accuracy is low, it is sufficient for some mobility and manipulation tasks as demonstrated in this thesis. Strain gauges or series elastic actuators can have higher accuracy. This would increase the sensitivity to forces and help all force-domain tasks. Besides better hardware, research on better manipulation capabilities need to be developed. Finding force on parts of the arm other than on the endpoint given the joint torques is an interesting problem. Personal robots will need to be able to do enough tasks before they become more widespread.

APPENDIX A

EIGENVECTORS AND EIGENVALUES

An eigenvector of a linear transformation A is a non-zero vector $\boldsymbol{x}$ that satisfies the equation A$\boldsymbol{x} = \lambda\boldsymbol{x}$ for some scalar λ[5]. The value λ that satisfies this equation for an eigenvector is called an eigenvalue of A corresponding to eigenvector $\boldsymbol{x}$. A method to solve for the eigenvalues involves solving for the equation (A - λI)$\boldsymbol{x} = 0$. According to Cramer's rule, this system of equations can be solved for the non-trivial solutions by solving for when det(A - λI)$\boldsymbol{x} = 0$. The left side of this equation is known as the characteristic polynomial and the eigenvalues are the roots of the characteristic polynomial.

The roots (eigenvalues) of a second-degree characteristic polynomial are solved using the quadratic formula. Solving for the roots of a cubic polynomial is of particular interest because Cartesian space is 3-dimensional. Typical Jacobians are $3 \times n$ where n is the number of configuration variables so JJ^T is a 3×3 matrix. Closed form solutions exist to find roots of cubic polynomials. Once the eigenvalues are found the eigenvectors can be solved for by plugging in an eigenvalue λ into the equation $A\boldsymbol{x} = \lambda\boldsymbol{x}$. Solving this for the system of equations will yield as many eigenvectors as the multiplicity of the eigenvalue.

APPENDIX B

ENDPOINT POSITION CONTROLLER

An endpoint position controller takes as input a desired Cartesian position reference to achieve and a desired endpoint speed. The controller computes a gradient to follow to that leads towards the reference position. A normalized vector $\boldsymbol{d}$ is computed which is the direction the endpoint needs to move computed by subtracting the desired endpoint position from the current endpoint position. In order to approximately move the endpoint at a constant speed an endpoint displacement $\boldsymbol{x}$ is found which is $\boldsymbol{d}$ scaled by an amount to move in the direction for the servo-cycle computed by desired endpoint speed times the elapsed time since the last servo-cycle. The Jacobian relates joint angle rates of change to rates of change in the end-effector in Cartesian position. The pseudoinverse of the Jacobian multiplied by $\boldsymbol{x}$, which is a small endpoint displacement, gives joint angle displacements that moves the endpoint towards the desired position. The pseudoinverse is not unique for the uBot-5 because the manipulator is redundant. The method used to try and have the end-effector achieve the desired joint angle displacements is adding the resulting joint angle displacements to desired reference positions.

BIBLIOGRAPHY

[1] Adler, Jane. A robot may help improve your senior homelife. http://www.chicagotribune.com, March 7 2004.

[2] Bryan J. Thibodeau, Patrick Deegan, Roderic Grupen. Static analysis of contact forces with a mobile manipulator. In *Proceedings of IEEE International Conference on Robotics and Automation, Orlando, FL* (2006), pp. 4007–4012.

[3] Coelho, Jr. J. A., Grupen R. A. A control basis for learning multifingered grasps. *Journal of Robotic Systems 14* (1997), 545–557.

[4] Connolly, Christopher I., and Grupen, Roderic A. Applications of harmonic functions to robotics. Tech. Rep. UM-CS-1992-012, , 1994.

[5] Curtis, Charles W. *Linear Algebra An Introductory Approach*. Springer, 1984, ch. 7, p. 189.

[6] Deegan, Patrick, Grupen, Roderic A., Hanson, Allen, Horrell, Emily, Ou, Shichao, Riseman, Edward M., Sen, Shiraj, Thibodeau, Bryan J., Williams, Adam, and Xie, Dan. Mobile manipulators for assisted living in residential settings. *Auton. Robots 24*, 2 (2008), 179–192.

[7] Edsinger-Gonzales, Aaron, and Weber, Jeff. Domo: A force sensing humanoid robot for manipulation research. In *Proc. IEEE/RSJ International Journal of Humanoid Robotics* (2004).

[8] Fu, Sheng, Liu, Hui-ying, Gao, Lu-fang, and Gai, Yu-xian. Slam for mobile robots using laser range finder and monocular vision. In *Proceedings of the 14th International Conference on Mechatronics and Machine Vision in Practice* (2007), pp. 91–96.

[9] Grupen, R. *Robotics and the Developmental Organization of Robot Behavior*. MIT Press, Cambridge, MA, in press.

[10] Halmick, Daniel, Okon, Avi, and DiCicco, Matt. A comparison of force sensing techniques for planetary manipulation. Tech. rep., March 2006.

[11] Hart, Stephan, and Grupen, Roderic. Natural task decomposition with intrinsic potential fields. In *Proceedings of the 2007 International Conference on Robotics and Systems, San Diego, CA* (2007), pp. 2507–2512.

[12] Kumpf, Adam. Explorations in low-cost compliant robotics. Tech. Rep. MIT-CSAIL-TR-2007-006, January 2007.

[13] Markoff, John. It's moore's law, but another had the idea first. `http://www.nytimes.com`, April 18 2005.

[14] Moore, Gordon E. Cramming more components onto integrated circuits. In *Proceedings of the IEEE, January* (1998), pp. 82–85.

[15] Platt, Robert J. *Learning and Generalizing Control-Based Grasping and Manipulation Skills*. PhD thesis, University of Massachusetts Amherst, 1996.

[16] Simpson, John W.L, Cook, Chris D, and Li, Zheng. Sensorless force estimation for robots with friction. In *Proc. 2002 Australasian Conference on Robotics and Automation* (2002), pp. 94–99.

[17] Stiehl, Walter D., Lieberman, Jeff, Breazeal, Cynthia, Basel, Louis, Cooper, Roshni, Knight, Heather, Lalla, Levi, Maymin, Allan, and Purchase, Scott. The huggable: A therapeutic robotic companion for relational, affective touch. In *Proceedings of the Consumer and Networking Conference, 2006* (2006), pp. 1290–1291.

[18] Tapus, Adriana, Tapus, Christian, and Mataric, Maja J. User-robot personality matching and assistive robot behavior adaptation for post-stroke rehabilitation therapy. *Intelligent Service Robotics 1* (2008), 169–183.

[19] Veloso, Manuela, Lenser, Scott, Vail, Douglas, Rybski, Paul, Aiwazian, Nick, and Chernova, Sonia. Cmrobobits: Creating an intelligent aibo robot. In *Proceedings of the AAAI Spring Symposium on Accessible Hands-on Artificial Intelligence and Robotics Education, Palo Alto, CA* (2004), pp. 1–6.

[20] Williamson, Matthew M. Series elastic actuators. Master's thesis, Massachusetts Institute of Technology, February 1995.

[21] Yussof, Hanafiah, Yamano, Mitsuhiro, Ohka, Masahiro, and Nasu, Yasuo. Development of a contact interaction-based navigation strategy for a biped humanoid robot. In *Proceedings of the 2007 International Conference on Intelligent Robots and Systems, San Diego, CA* (2007), pp. 4241–4246.