A Planar Robot for High-Performance Manipulation

CMU-RI-TR-00-19

Arthur E. Quaid

Submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Robotics

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania

Abstract

A novel planar robot is used to demonstrate that large numbers of axes are not required for effective manipulation, and that there can be strong advantages to eliminating excess degrees-of-freedom. This robot combines sub-micron precision with a meter-sized workspace, motions with velocities above 1 m/s, and accelerations several times that of gravity. It combines planar stepping motor (Sawyer motor) actuation technology with a compact, high-precision, high-bandwidth position sensor developed at Carnegie Mellon University.

The actuator operation is experimentally characterized over the closed-loop operating regime using an automated experimental setup that incorporates a laser interferometer and load cell. Parametric models are constructed based on these experiments that allow for autonomous calibration of the actuators and sensors. This calibration enables on-site calibration, improving the precision of both positioning and force output without requiring any external measuring devices.

Coupled with software-based calibrated models and commutation, the stepping motor actuators can be operated as servo motors, simplifying the controller design and eliminating the underdamped oscillation modes that limit the robustness of open-loop stepping operation. PD and PID controllers are used to demonstrate the performance improvements from closed-loop operation, which include improved repeatability (to sub-micron levels), fast settling times (20 ms), and a four-fold decrease in energy usage.

The main application of interest for this robot is for conveyance and fine motion control of subproducts in a rapidly-deployable assembly system called minifactory. This system requires low-DOF robotic agents to cooperatively perform higher-DOF tasks, coordinating their actions through optical endpoint sensing, networked communications, or the task dynamics. It also requires each robotic agent to be trustworthy to ensure system reliability. The performance and robustness of closed-loop operation is demonstrated with a high-speed wall-following task, a high-precision visually guided positioning task, and dynamic visual tracking task.

A mobile vibratory parts feeding application is also presented. This application exploits the dynamic range of the planar robot, which allows for gross motion and precision positioning of the feeder as well as a high enough bandwidth to generate 30 Hz vibratory waveforms. It also is an application that makes good use of the limited travel rotational axis of the robot.

Acknowledgements

The work described in this dissertation is by far the hardest and longest I've ever worked on one set of problems. For a while, I felt like it might never come to completion. On the other hand, I never was in much of a rush to finish, probably because CMU was such a good place to be. Then suddenly, I'm packing up the truck and leaving town!

Thanks of course to Dr. Ralph Hollis for the original core ideas, technology, and motivation for this work, and for his guidance, support, advice, encouragement, and (in the finishing-up stages) patience. Thanks to Dr. Alfred Rizzi for teaching me so much about so much. Going back further, I had many interesting conversations with Prof. Yangsheng Xu regarding controls (as well as academic life in general) during his early involvement with this work.

Everyone in the MSL – Ralph, Al, Zack, Jay, Pat, Mike, Rich, Shinji, Jimmy, Greg, Ben, Alan – contributed to the incredible amount of hardware, software, and laboratory infrastructure necessary for this work. Zack Butler's efforts in keeping the sensing electronics functional and in helping me figure out hundreds of strange Matlab plots over the years are especially noteworthy.

Specific acknowledgements related to the document need to be made. Various figures in the text may have been generated, at least in part, by other lab members. Much of the text has been published previously as conference papers co-authored with Ralph Hollis and Al Rizzi, and a few snippets of their original writing may have survived revisions to make it into this document. Of particular note, Al Rizzi collaborated closely with me on the autonomous calibration and hybrid control chapters.

This work was supported in part by NSF grants DMI-9523156 and DMI-9527190, and the CMU Engineering Design Research Center. I was supported in part by a Lucent Technologies Foundation Fellowship. IBM donated the commercial planar linear motor equipment, interferometer systems, and much of our lab equipment.

Contents

List Of Tables

List Of Figures

Chapter 1

Introduction

```
     The secret of getting ahead
         is getting started.
Lucky Numbers 11, 16, 27, 28, 36, 37
```

Robot manipulators usually come in the form of serial-chain industrial robots, as reflected in the ISO 8373 definition [1] of a "Manipulating Industrial Robot:"

> An automatically controlled, reprogrammable, multipurpose manipulator programmable in three or more axes, which may be either fixed in place or mobile for use in industrial automation applications.

Although these robots are touted for their general purpose abilities, their use is often restricted to repetitive positioning operations, such as welding, painting, or pick-and-place operations using simple binary (open/close) grippers. For this reason, robot designs and performance specifications reflect an emphasis on the kinematic and dynamic requirements of tracking workspace trajectories and attach less importance to the details of interacting with the environment. While many automation systems have been designed around robots performing positioning tasks, this mode of operation is perhaps the least challenging and represents a barrier to wider application of robotic technologies.

The ISO definition above makes no requirements on the axes of a robot beyond their mere existence. Precision, speed, bandwidth, etc. are not considered, even though in practice the range of economically useful applications for a robot will depend on these qualities. Furthermore, manipulation encompasses much more than positioning operations. Large, heavy, stiff industrial robots are poor at tasks that require substantial interaction with their environment. How can we provide robots that have a richer set of manipulation capabilities? One approach is to add sensing

and actuation to the end of the robot that can approximate the abilities of the human hand [2, 3] . However, this approach tends to lead to systems that are too fragile, expensive, or slow to be used outside of research environments. Coarse-fine systems [4, 5] take a similar approach, adding high-performance but small workspace degrees-of-freedom (DOFs) that are redundant with the primary (large workspace) robot actuators. These systems improve performance, but the need for redundant DOFs increases system cost and complexity.

The need to put these devices on the end of the robot arm is, in large part, to compensate for the deficiencies and poor performance of the arm itself. It is understandably difficult to have rich interactions with the world when six or more actuators, bearings, transmissions, and links come between mechanical ground and the task of interest. What if most of this apparatus were eliminated, providing a much shorter, simpler, and precise connection? What types of manipulation would become possible?

This dissertation demonstrates that the *quality* of the degrees-of-freedom can be more important than the *quantity* of degrees-of-freedom when judging the manipulation capabilities of a robot. The moniker "manipulation robot" should mean more than the ability to move in some number of different directions. A true robot for manipulation should be able to perform a variety of manipulation tasks – precision positioning, fast positioning, tapping, sliding, poking, stretching, shaking, vibrating, pushing, throwing, etc. – with sufficient performance to be a viable replacement for special-purpose devices. Encapsulating these manipulation capabilities into a single device is useful not only for flexibility and re-use, but also for exposing the impoverished ways in which robots interact with their environments. Although only hinted at in this work, the eventual goal is that the availability of a wide variety of manipulation capabilities in a single robot will enable and inspire new models for how to think about, design, and program robotic manipulation tasks.

1.1 Low-DOF Robots

Robotic devices with less than six DOFs have found wide usage. The SCARA design [6] with four DOFs is commonly used for vertical placement operations. Simpler one- and two-DOF robots are common in fixed automation systems, although robotics researchers might think of them as mere "actuators." Though the SCARA might

have a higher-level programming language, in practice the two types of robots perform very similar tasks–moving between pre-programmed setpoints and waiting for some event (gripper open or close, photo detector on or off) to occur.

Researchers have also used low-DOF robots. Often, developments in control theory are applied to an arm with two revolute joints, especially where dynamic performance [7, 8] or environmental interaction [9] is required. These robots are not expected to have immediate application, but are useful in providing a non-trivial experimental testbed for research and teaching without introducing the complications that arise with longer serial chains. There has also been recent interest in robotic minimalism [10]. The minimalist approach [11] is to design manipulation devices and strategies that avoid using resources that had previously seemed necessary for a task. This approach includes systems with reduced actuation [12, 13, 14]. The planar robot of this dissertation is minimalist in that it has fewer DOFs than typical for manipulators for assembly.

1.2 Sawyer motor technology

There has been both industrial and academic interest in assembly workcells using low-DOF *Sawyer motor* based robots. Invented by Bruce Sawyer [15, 16, 17] in the late 1960's, Sawyer motors have multiple linear stepping motors combined into a single housing. Such a planar motor requires two parts: a moving *forcer*, and a stationary *platen* stator surface, similar to the rotor and stator in a rotary stepping motor.

The forcer (Figure 1.1) contains four linear motors and channels for providing an air bearing between itself and the platen. Each of the four motors comprise a stack of laminations, two coils, and permanent magnets, shown schematically in Figure 1.2. Each motor produces force along one direction. The motors operate on a flux-steering principle, with the coil currents acting to switch the permanent magnetic flux from one set of poles to the other. The poles with the most flux tend to align themselves with the platen teeth, so that by activating the poles in the proper order, a stepping motion is achieved. The currents in the motor coils can also be smoothly varied, making it possible to set the equilibrium position to any position relative to the teeth. In practice, the limits of this open-loop *microstepping*

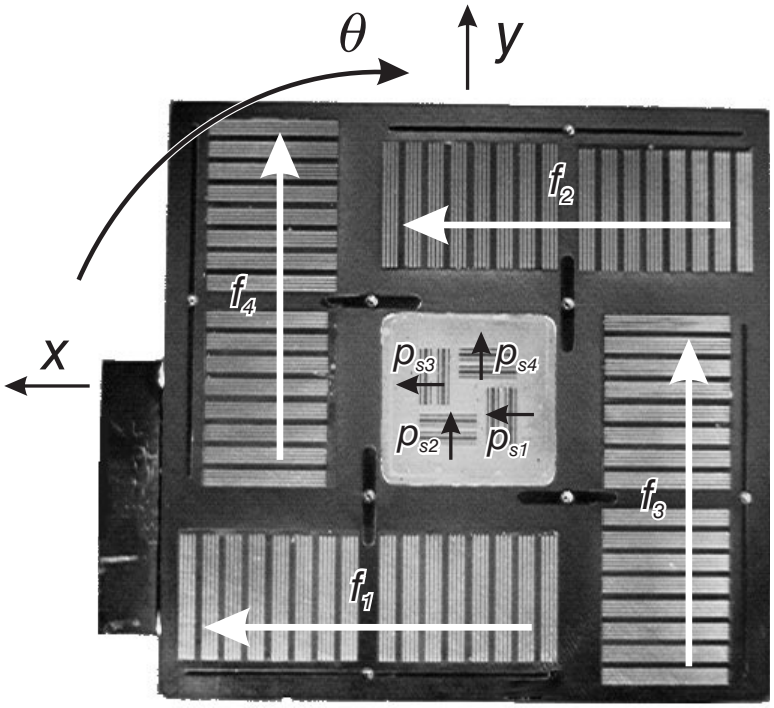


Figure 1.1: Bottom view of forcer (*normag1* system)

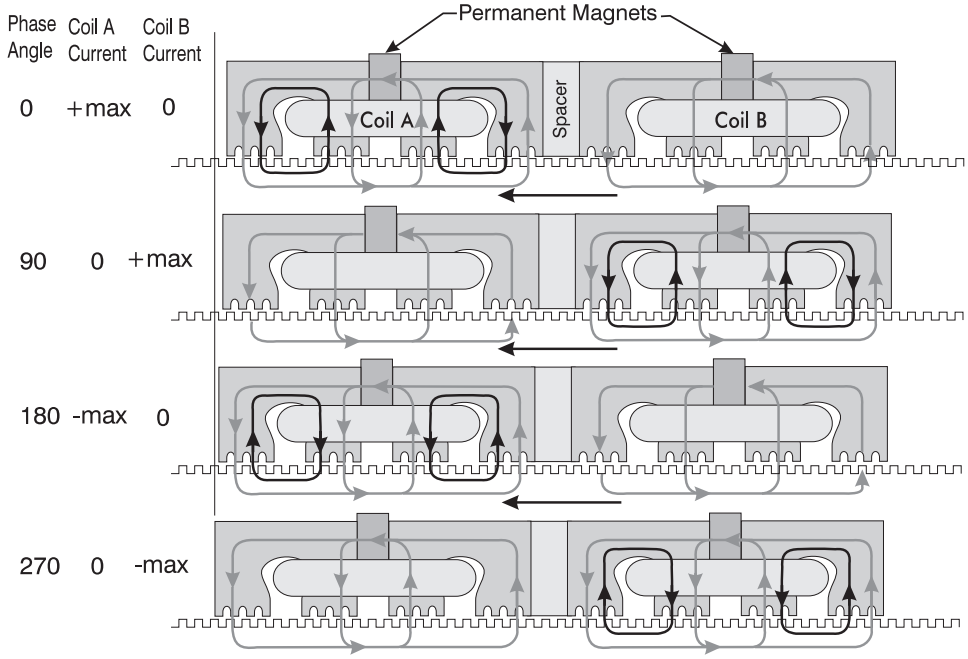


Figure 1.2: Basic linear motor operation: Current in the motor coils generates magnetic flux (dark flux path) that sums with the permanent magnet flux (light flux path) to produce forces.

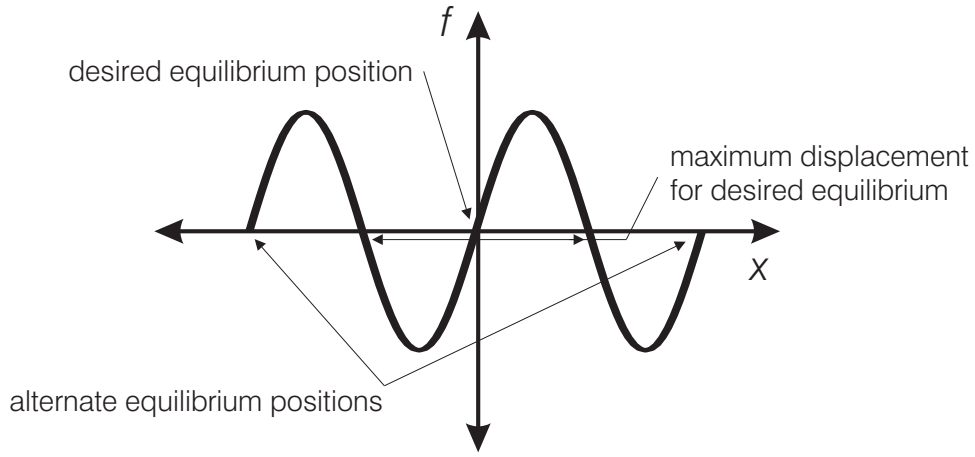


Figure 1.3: The open-loop force profile has a large stiffness near the equilibrium position but this equilibrium is only attractive for positions within $\pm\rho/2$.

operation are about 1/500th of a 1 mm tooth pitch, or about 2 μm. If the equilibrium position is fixed by sending appropriate constant currents in the coils and the forcer is displaced from the equilibrium position by an external device, it will resist with a force that to first order is a sinusoidal function of position, as shown in Figure 1.3. Note that the actuator only applies force when it is displaced from the equilibrium position, so that motions that require a significant fraction of the peak force will require displacements that move the system operating point closer to the maximum displacements shown in Figure 1.3. When floating on a platen, the air bearing is strongly pre-loaded by the permanent magnets of the motors, resulting in an extremely stiff, near-frictionless air bearing with a thickness of approximately 12 μm. A tether is required for supplying air and motor currents. Motor currents are typically supplied by pulse-width modulation (PWM) current-output amplifiers, with two channels required for each of the four linear motors.

The platen (Figure 1.4) is a passive steel surface with a two-dimensional waffle-iron pattern of teeth, with the gaps between teeth filled with epoxy to provide a smooth air bearing surface. The tooth pattern may be produced in the thin soft magnetic steel top sheet of the platen by chemical etching or machining techniques. To make it suitable for the thin air bearing, its surface must be locally flat to within a few microns over regions corresponding to the area covered by a forcer. Honeycomb material or a web of cast material is used to provide a stiff, stable structure to support the steel top sheet. The nominal tooth pitch, ρ, of most commercially

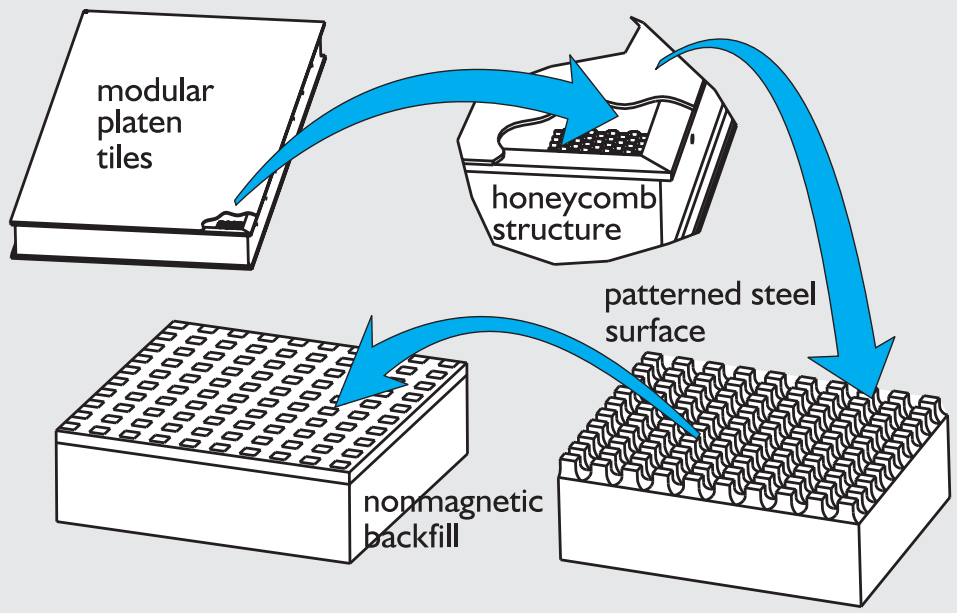


Figure 1.4: Platen structure and fabrication

available platens is 1.016 mm (0.04 inch), and they are available in a variety of sizes up to 0.9 by 1.3 m [18]. Forcers and platens are available commercially as individual components and as parts of integrated systems [19, 20, 21].

Planar motors are unique in combining high-speed and high-acceleration motions over a large-workspace, as well as exceptionally precise positioning capabilities. Northern Magnetics, Inc. reports [18] forcers with 60 N static force, better than 1 m/s peak velocity, 40 m/s^2 acceleration, and 3 μm positioning resolution. Note that the planar motor has 3-DOFs but only a single moving part, combining the high stiffness of parallel mechanisms with the large workspace of serial mechanisms. However, the open-loop stepping operation has limited the use of these actuators to positioning applications. Open-loop operation can be characterized as a very stiff, lightly damped, passive position control loop. This behavior is illustrated in the impulse response of Figure 1.5, where the system vibrates at approximately 60 Hz for a large fraction of a second. For a linear system, such dynamics affect performance but not stability. However, in this case, because of the force response of Figure 1.3, the passive stiffness of the open-loop stepping mode will be non-linear, and deviations from the equilibrium position larger than one-half of a tooth pitch will result in the system moving to the adjacent tooth. In the absence of sensing, this condition cannot be detected and the forcer position will have a steady-state

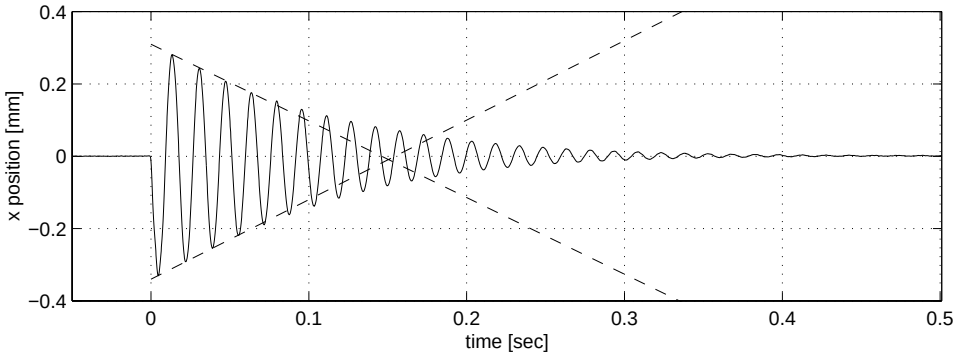


Figure 1.5: The impulse response under open-loop operation appears to have a pair of underdamped poles at a natural frequency of 60 Hz with a time constant of 150 ms.

error of one pitch. This occurrence is referred to as a *loss of synchrony*, and may also involve rotational errors or translational errors of more than one tooth pitch. For this reason, changes in the equilibrium position of the stepping mode must be made carefully to avoid exciting the underdamped dynamics and must be followed by appropriate delays to give them time to dissipate.

Recently, a high-precision, high-bandwidth position sensor [22] was developed at Carnegie Mellon University. Developed in the Microdynamic Systems Laboratory, this sensor [22], shown in Figure 1.6 comprises four segments, each capable of measuring position along one direction. This sensor takes advantage of the existing precision grid etched into the platen. As shown in Figure 1.7, a drive coil generates an alternating magnetic field that flows through the platen teeth and returns through two sets of sense coils. The relative magnitudes of the sense winding outputs vary based on which coil is aligned with a row platen teeth. The output of custom-designed demodulation circuitry is a pair of amplified sense coil amplitudes $\{v_{a,i}, v_{b,i}\}$ for each of the four sensor segments. Plotting a pair in an (x, y) plot yields a rounded diamond shape (Figure 1.9). Taking the arctangent function of this signal pair yields, to first order, the position of the sensor element, expressed as a *spatial phase*. The spatial phase, ϕ_s, is related to the position p_s by the proportional relationship $p_s = \phi_s \rho/(2\pi)$, where ρ is the pitch of the platen teeth. It is also necessary to read the sensor often enough that a tooth counter can be incremented or decremented whenever the spatial phase transitions from the upper to the lower range of the arctangent function or vice-versa. Using this method to compute the position, the noise level of a single axis of the integrated sensor was reportedly [22]

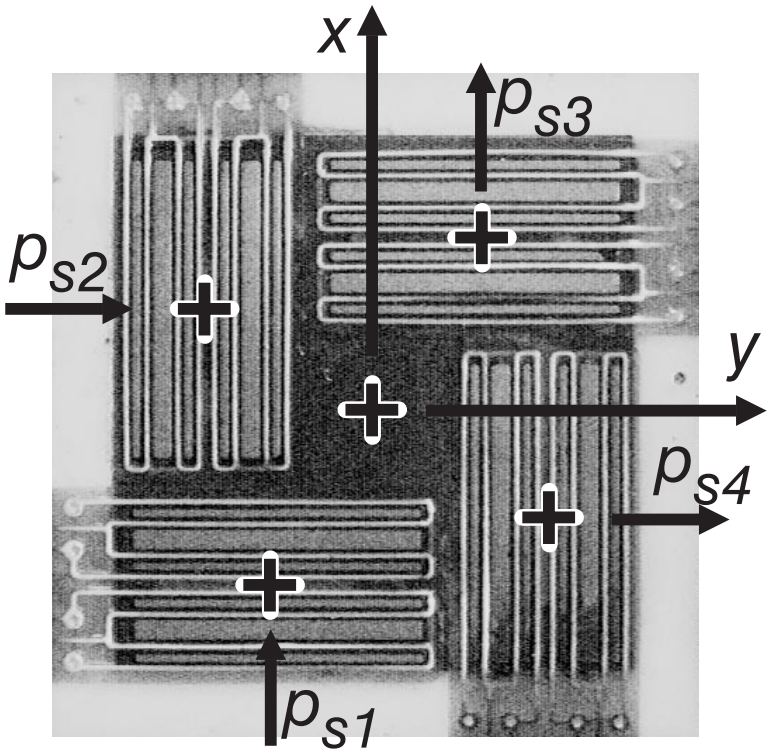


Figure 1.6: Platen sensor detail from *normag2* system

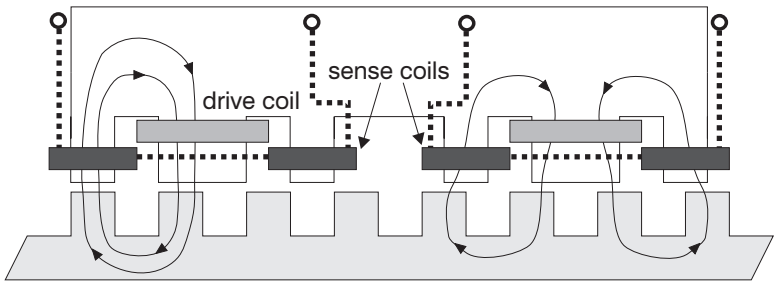


Figure 1.7: Schematic of a single platen sensor segment

dimension	*normag0*	*normag1*, *normag2*, *vole*
d_f [mm]	96	150
d_a [mm]	24.5	48.5
d_{wa} [mm]	31	31
d_s [mm]	n/a	9.7
d_{so} [mm]	n/a	15.8
d_{ws} [mm]	n/a	13.8

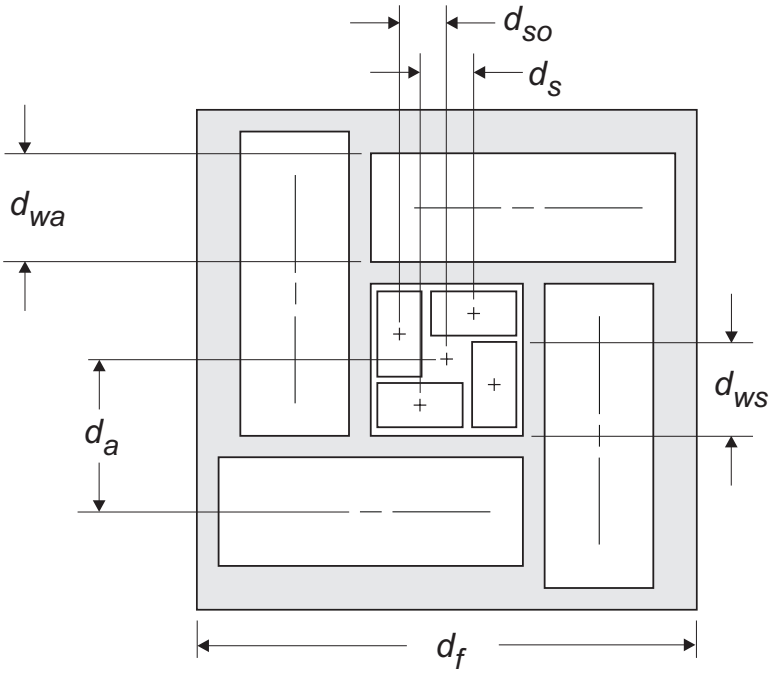


Figure 1.8: Forcer dimensions

0.3 μm (1σ). However, two of the four sensor segment output positions are averaged to compute the position of the forcer, which reduces the noise level to 0.2 μm ($1\ \sigma$) because the noise of each sensor is independent. The forcer skew angle is computed from the difference of either the x direction ($q_{\theta,x} = (p_{s1} - p_{s3})/d_s$) or y direction ($q_{\theta,y} = (p_{s2} - p_{s4})/d_s$) sensor segments divided by the distance between sensors, d_s in Figure 1.8. In practice, both differences are computed and averaged to reduce the noise level of the angle measurement. Characterizing the accuracy of the sensor is more complicated. The global accuracy will be affected by the accuracy of the platen grid and thermal deformations of the platen. Assuming the platen is thermally stable and the grid is to be used as the primary position reference in a system, the local accuracy is more important. The local accuracy will depend on local tooth non-uniformities, the errors that result from using an arctangent function to demodulate the rounded diamond output of the sensor, and the magnetic state of the platen material. The CMU sensor differs from other magnetic platen sensor designs [23, 16, 24, 25, 26, 27] in its small size, which eases integration and allows for high-frequency operation, its balanced construction, which makes it insensitive to bias fields from the actuators, and its simple mechanical design, which makes it easy to fabricate. Experiments using a laser interferometer [22] demonstrated accuracies below ± 10 μm after calibration using a Fourier error model.

Sawyer motors have received a lot of attention from academic and industrial researchers. The original invention was documented in a series of patents by Bruce Sawyer covering the linear actuator operating principle and the grouping of actuators in a single housing to enable planar motion [15, 28], integration of displacement, velocity, and acceleration feedback information [17], and trajectory design to minimize excitation of the open-loop dynamics [29]. Xynetics[1] licensed [30] and further developed this technology. The Xynetics group also had a number of technical publications discussing trajectory design and motion control implementation [31], constant velocity systems [32], performance specifications and accelerometer feedback for increasing damping [33], as well as system overviews in trade publications [34]. A later Xynetics patent covers a method for reducing the effects of magnetic hysteresis in the platen [35].

[1]Xynetics became a division of General Signal, Inc. in the early 1980's. General Signal was in turn acquired by SPX Corporation in October 1998.

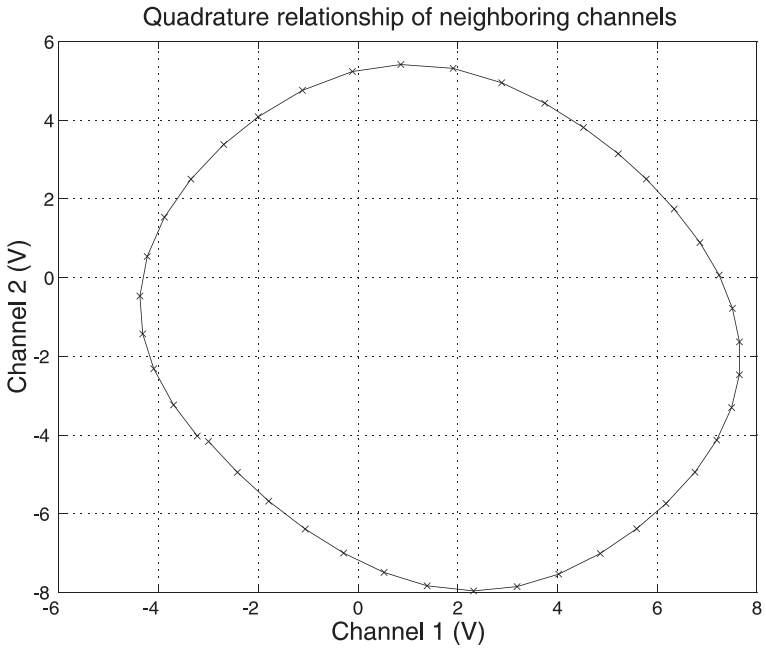


Figure 1.9: Platen sensor quadrature output

Early applications of Sawyer motors were limited to flat-bed plotters and wafer positioning stages [36, 37]. However, several groups began developing assembly workcells based on the technology. The RobotWorld concept was reportedly developed by Victor Scheinman as early as 1979 [21]. The Automatix[2] RobotWorld system [19] is designed to use planar motors for vision guided assembly. In this system, the platens form the ceiling of a workcell. Multiple motors augmented with z, θ axes are operated hanging down from the platen, along with 2-DOF motors that position mobile cameras for inspection and registration. Industrial applications include mechanical and electronic assembly [19, 21]. Academic researchers have also been attracted to the structured environment of RobotWorld, using it to develop DesignWorld, an automated circuit build and maintenance workcell [38], and modular manipulation workcells [39]. Recently an optical sensor [40, 41] with the trade name "GridTrac" was introduced as an option for the RobotWorld robot modules. This approach requires four sensor segments to be mounted on the outside of the forcer, increasing its effective footprint.

[2]Automatix was acquired by Yaskawa Electric Corporation, and the RobotWorld product line is currently distributed by Yaskawa's Motoman division.

Contemporaneous with the RobotWorld introduction, a similar assembly workcell was designed at AT&T [42] and Megamation, Incorporated[3] [43]. The Megamation systems differ from the RobotWorld systems in form more than function. The controller hardware is mounted above the platen to reduce floorspace requirements and allow the system to be mounted over a conveyor or other transfer line, making it easier to integrate the system into existing assembly lines. Megamation has patented a number of technologies related to planar motors, including their overall system design [44], an optical sensing technique that relies on an extra grid on the surface of the platen [45], a design for laminated platens that reduces eddy current damping effects [46], a method for detecting stall conditions without position sensors [47], a mechanism for reducing tether disturbances [48], a liquid-cooled platen [49], and a fiber optic encoder position sensor [50]. They also have patented devices with multiple planar motors for a large area coating system [51] and high force applications [52], as well as adding a coarse/fine planar motor system for improved resolution [53]. Applications for the Megamation system include thru-hole and surface-mount printed circuit board assembly [43], test and inspection of electronics, small mechanical assembly, laboratory automation [20], and fine-pitch surface mount placement [54, 55]. There are reportedly over 350 Megamation robot heads in use [56].

Planar motor systems for assembly were also under development internally at IBM as early as 1987 [57]. Early published results of this work involved platen sensors for planar motors based on both ac magnetic [24, 58] and optical [59, 27] operating principles. These sensing technologies were further refined and evaluated at Carnegie Mellon University [22, 60]. Prototypes of the CMU magnetic sensor were integrated into two commercial forcers and enabled the work described in this dissertation. Portions of this dissertation have been published in other forms, including the modeling of planar motors [61], demonstration of full 3-DOF closed-loop control of planar motors [62], and autonomous calibration of the platen sensor [63]. Also at CMU, a miniature, rapidly-deployable assembly system called *minifactory* [64, 65, 66, 67, 68, 69] is under development and uses planar robot couriers to provide both sub-product transportation and precision positioning. The couriers are also used for determining the configuration of the factory at startup [70, 71], and as mobile vibratory parts feeders [72, 14].

[3]Megamation is now a product line of Rankin Corporation.

The IBM sensor was also used in work at Columbia University and Polytechnic University, including preliminary experiments with 1-DOF of a planar motor under PD control [26, 73], and modeling the actuator and sensing outputs as a function of skew angle [74]. Further work related to planar motors includes an investigation of the inaccuracies resulting from thermal deformations of the platen [75, 73], and a composite platen design for reducing eddy-current effects [76, 73]. A robust, adaptive controller was also designed for closed-loop control of a planar motor, but only simulation results have been reported [77, 78, 79, 80, 81, 82].

A capacitive sensing design was developed at Bell Laboratories [83, 84]. At MIT, several recent Master's Theses [85, 86, 87] have studied planar motor and sensor design, modeling, and control. They have reported experiments demonstrating 1-DOF sensor operation and PID control [88, 25]. Many researchers have been attracted to the problems of multiple tethered planar robots operating on a single platen [89, 90, 91, 92, 93]. Parallel mechanisms actuated by multiple planar robots have also been reported [94, 95, 73].

Most of the above work is focused on improving the trajectory tracking and position regulation performance of planar motors, either by incremental improvements to open-loop operation or a more fundamental switch to closed-loop control. These developments lead to a number of improvements. Thermal degradation of positioning accuracy is reduced by building platens with improved thermal dissipation or by reducing currents by using closed-loop control. Repeatability and accuracy at the ends of trajectories are improved by closed-loop control. The long settling times of open-loop control can be reduced by using winding source feedback or accelerometers [33], or by adding damping with closed-loop control. Trajectories can be applied that have velocities and accelerations closer to the actuator limits. These improvements lead to cycle time reductions and accuracy improvements that may be critical for certain applications. However, for trajectory tracking applications that do not require these improvements, the reduced system complexity of open-loop operation makes it preferable.

What has not been emphasized in the literature is that closed-loop operation enables new classes of applications that go beyond simple positioning tasks. Open-loop operation can only be used to track pre-defined trajectories that have been carefully designed to stay within the velocity or acceleration limits of the forcer and not to excite any of the resonance modes. Closed-loop control allows the planar motors

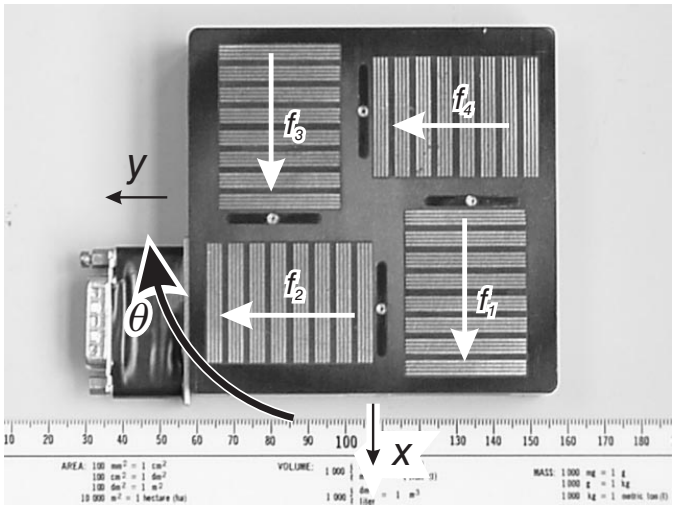


Figure 1.10: Forcer for *normag0* system

to have richer dynamic interactions with the world – guarded moves, disturbance detection and compensation, visual servoing, automatic registration, cooperative manipulation, high-frequency motions, and impedance control – interactions that are impossible under open-loop operation, and that are critical for enabling high-performance manipulation.

The central motivating idea of this dissertation is that high-performance low-DOF robots should be considered for use in challenging manipulation applications. The technical contributions of this dissertation are in taking the raw planar motor sensing and actuation technology, developing the necessary modeling, calibration, and control techniques, and demonstrating that the low-DOF robot formed by the combination of these technologies and techniques is well-suited to high-performance manipulation tasks.

1.3 Outline and summary

At the coarsest level, the body of this dissertation can be divided into three parts. First, Chapters 2, 3, and 4 describe the commutation, modeling, and calibration that are needed to prepare the linear stepping motors for use as servo motors. Second, Chapters 5 and 6 describe the planar robot control problem, present theory and real-time implementation of appropriate controller strategies, and evaluate the advantages of these controllers relative to open-loop operation. Third, applications

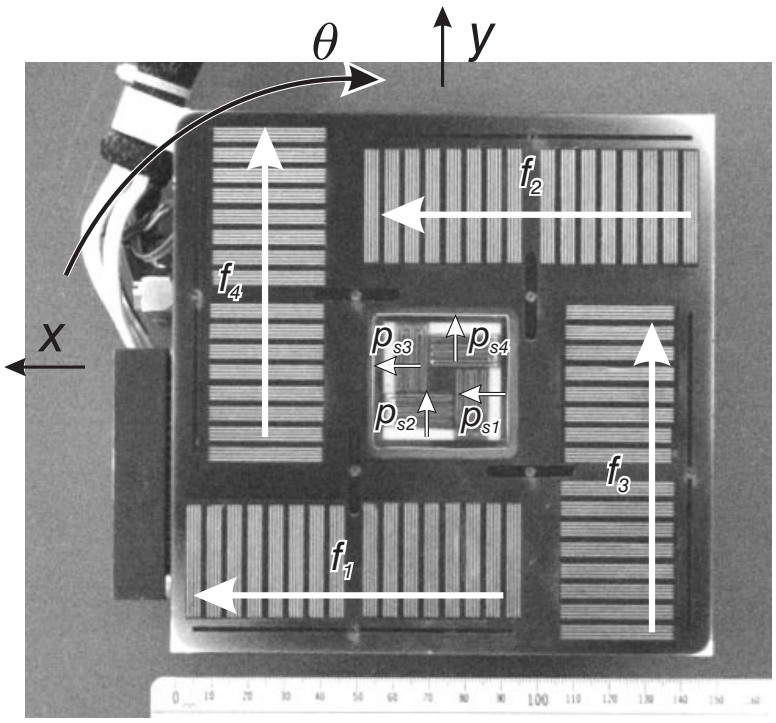


Figure 1.11: Forcer for *normag2* system

are presented in Chapters 7 and 8 that illustrate the use of the planar robot in challenging manipulation tasks, highlight its unique performance capabilities, and show how closed-loop operation enables these new applications.

In more detail, Chapter 2 discusses the development of software-based *commutators* for the planar motor actuators. The commutators compute appropriate currents to send to the actuators based on their position and the desired force output. This commutation strategy is different than that used in open-loop Sawyer motor operation, where an electronic commutator chooses actuator currents to set the equilibrium position of the actuators based on the desired motor position without any information about the actual motor position. The two main differences are the change from a position commanded stepping operation to force commanded servo operation, and the implementation of the commutation function in software rather than in electronics. Two force commutation strategies are presented. A *fixed-amplitude* strategy is similar to open-loop commutation in that large currents are always in the coils. A preferred *fixed phase* strategy only requires large currents when large forces are required. Although the commutator is presented using a first-order model of the actuator, the software implementation allows more complicated and

modifiable models to be used. However, the software implementation cannot run at the speeds of dedicated hardware, so that a special phase compensation technique is required to maintain force output at high velocities. Commutation in general is a well-understood technique, but the special issues involved with software-based commutation of planar motors have not been presented in the literature. A software-based fixed-phase commutator has been reported [96], but this work did not consider phase compensation and did not present experiments with high-velocity motions that would require phase compensation for reasonable performance. A fixed-phase commutator has also been used [77] for a simulated planar motor control, where phase compensation is not an issue. A fixed-amplitude commutator has also been used [40], but only at low speeds. Other reported one-DOF planar motor control experiments [26, 73] did not reveal the commutation method utilized.

There has been much work on modeling of planar motors and other similar motors [97, 96, 74], but they all have major simplifications and very few include any experimental verification of the resulting models. Also, many are intended to model the open-loop operational characteristics and are not concerned with the model fidelity at low currents, because open-loop operation will always have large currents in the motor coils. For example, most models presented in the literature predict zero detent forces, even though it is quite simple to qualitatively verify that this is not the case. The experiments of Chapter 3 are designed to quantify the force output as a function of actuator position, skew angle, commutator current command, and velocity. The quality and scope of these experiments are enabled by integrating a multi-axis laser interferometer system and a precision load cell with a real-time computing system. Data collection with this system was highly automated so that it was possible to explore along the many dimensions of actuator output variations. The same experimental setup was used [22] to determine an error model for the platen sensor. Results show the static force output of the actuators as a function of position, coil currents, and skew angle. A technique is used to measure the actuator force output as a function of velocity in such a way that damping and rolloff effects can be determined.

An interesting characteristic of the planar robot actuators is that the stator and forcers are separable. Multiple forcers can use the same platen, and forcers can be used on any platen that has the appropriate tooth pitch. In particular, the platens and forcers may not even have the same manufacturer. In addition, platens and

forcers from different manufacturers may have variations in tooth geometry or magnetic properties. For these reasons, calibration cannot generally be performed at the manufacturer. The techniques of Chapter 3 require too much equipment and time to be performed in the field on every platen/forcer pairing. However, those results can be used to formulate parametric models for the actuators and sensors, where the values of the parameters represent the important variations between platens and forcers. The redundancy of the sensor output can then be used to formulate a constraint equation that includes the sensor error model parameters. Appropriate data collection and minimization techniques can then be used to identify these parameters without the use of any external sensors. The actuator redundancy can also be exploited for calibration. In this case, if there are no external loads on the forcer, the steady-state actuator currents required to hold a position should be essentially zero. Any non-zero currents reflect disturbance forces, which will primarily come from unmodeled actuator characteristics. Without redundancy, this technique can only model the zero-force model errors. However, the redundant actuation allows the actuators to push against each other in such a way that each actuator can output an appropriate high force without generating a net force or torque on the forcer body. Exploiting the redundancy in this way allows the calibration of the full range of the actuators without requiring any external force references or sensors. These techniques were inspired by work in autonomous calibration of redundant manipulators [98]. A similar technique has been used [99] to calibrate fingertip force sensors. This dissertation reports the first application of these techniques to planar robot actuators and sensors. The autonomous platen sensor calibration improves the translational accuracy modestly from 8.9 μm (1 σ) to 6.5 μm and the angular accuracy dramatically from 1.2 mrad to 0.4 mrad, as verified using a laser interferometer. Further improvements in the sensor calibration were attempted using additional sensing information from the coordination sensor, but poor internal calibration of that sensor resulted in only marginal improvements over the autonomous platen sensor calibration. Autonomous calibration of the actuators resulted in improvements in force ripple ranging from 17 to 51%. Certain actuators seemed to have more ripple than others, and the large-ripple actuators tended to have the most improvement after calibration. It is expected that much of the remaining calibration errors in both the actuator and sensor models are caused by variations in the platen teeth that cannot be captured by the models used in the calibration. Figure 1.12 shows tooth defects

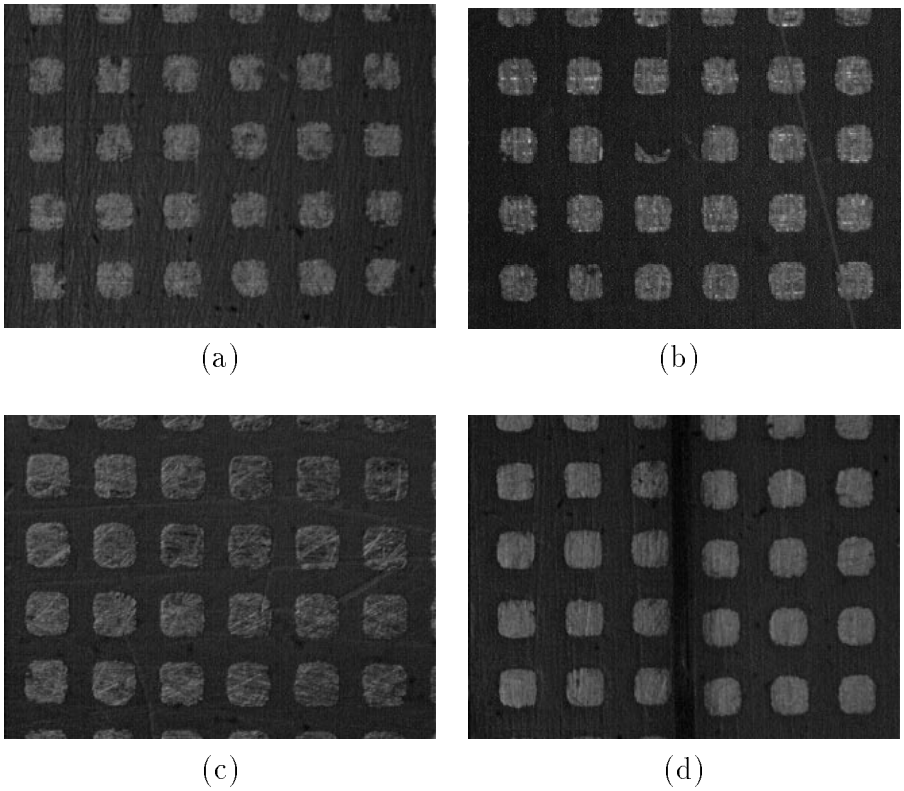
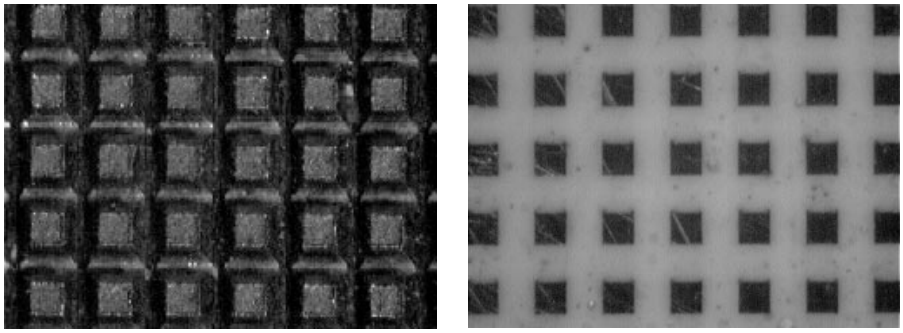


Figure 1.12: The Normag model #225-664-01 platen used for experiments in this work has (a) badly formed teeth, (b) missing teeth, (c) different tooth sizes in different areas, and (d) misaligned platen sheets, all resulting in sensor accuracy errors.

and variations in the commercial platen used for the experiments of this dissertation. Figure 1.13 shows the early results of platens under development at CMU that have markedly better tooth uniformity, which should enable further improvements in sensor accuracy.

Simple PD controllers are presented in Chapter 5 to demonstrate basic closed-loop operation, characterize certain performance improvements, and determine the difficulties of controlling the planar robot. Prior work [40, 26, 73, 96] reported control of only one axis using bulky prototype position sensors. In industry, accelerometer feedback and motor winding source feedback [33] have been used to improve performance and reliability of open-loop operation. The controller reported in Chapter 5, is the first published [62] instance of full 3-DOF closed-loop operation of a planar



(a) (b)

Figure 1.13: Small test platens before (a) and after (b) epoxy backfill indicate that platens under development at CMU should provide a more uniform reference for the platen sensor, improving accuracy.

motor with an integral position sensor. The planar robot control problem is identified, which includes issues that only arise under full 3-DOF control such as tether disturbances, sensor measurement errors, collisions, balancing the regulation of skew angle with the generation of propulsion forces, and coupled actuator saturation. A method for avoiding actuator saturation defines a domain to limit the operation of a controller to specified system states where the controller will provably move to the goal without exciting any actuator constraints. A disturbance estimator is used to provide a measure of disturbance forces acting on the system and to eliminate steady-state errors, if necessary. Improvements in resolution, repeatability, settling time, power usage, and operating temperature resulting from closed-loop operation are experimentally demonstrated.

One problem with the PD controller for gross motions is the need to track trajectories in the workspace. If the trajectory exceeds the capabilities of the robot or if disturbances result in large tracking errors, the PD controller may deviate from the trajectory path, taking a "short-cut" as necessary, or may have major overshoot at the end of the motion. These large tracking errors may also result in a violation of the angle limits of the skew regulation controller. The PD controller has trajectory design requirements similar to the open-loop controller, though the consequences may not be as severe. If the as-designed PD controller performance (tracking error, overshoot, etc.) is to be achieved, the trajectory must be carefully designed. This requirement can lead to significantly conservative approaches to motion planning and

inefficient use of the planar robots. In particular, if multiple planar robots share a platen, collision avoidance becomes necessary, and collision-free paths must be planned in $\mathbb{R}^2 \times t$ with enough spatial-temporal clearance to account for trajectory tracking errors. In Chapter 6, a different approach to the design of collision-free gross motions is presented. Instead of the relatively non-intuitive paths in $\mathbb{R}^2 \times t$, polygonal areas on the platen are defined as *reservation areas* in which only one planar robot may reside. Before moving into a new reservation area, the robot must negotiate exclusive access with its peers. Each polygonal area is composed of a series of convex regions. Within each convex region, it is necessary for a controller to keep the robot from violating the boundaries of the region and the acceleration and velocity constraints, while stably moving to a goal position within the region. By sequencing multiple convex regions, complex paths through the workspace can be formed without any explicit trajectory planning. By avoiding the use of trajectories, momentary or persistent disturbances will not be capable of inducing collisions. The design of the controller for the convex region is presented in Chapter 6 and builds on work previously developed [100, 101]. This controller is designed for gross motions, and is expected to be coupled with a local regulation or impedance controller that is activated when the gross motion completes.

The application that has motivated much of the work of this dissertation is the *minifactory* system. This automated assembly system [64, 65, 66, 67, 68, 69] is under development at CMU to enable rapidly-deployable automation. This system uses planar robot *couriers* to provide both sub-product transportation and fine manipulation capabilities. The couriers are also used for determining the configuration of the factory at startup [70, 71, 102]. Several minifactory tasks related to factory startup are presented, including automatic realignment, discovery of workspace boundaries, identification and calibration of overhead devices, and detection of platen cracks. These tasks are made much easier with closed-loop operation. Stable collision behavior is shown as well as fast sliding along platen boundaries, demonstrating that force and impedance control can be implemented using only the disturbance observer to provide force information. Multi-robot coordination is demonstrated for both a high-precision camera-guided positioning task and a dynamic, high-speed tracking task.

In Chapter 8, the planar motor is used as a miniature mobile parts feeder. Planar vibrations have been shown [103, 104, 72, 14] to produce motion of parts sitting on

a vibrating platform. The closed-loop bandwidth of the planar robot is high enough to track the vibration waveform and produce motion of the parts relative to the robot. A useful parts feeder requires part motion that tends to sort, separate, position, or orient in some known way. Others [13] have used sequences of rotational vibrations to independently position multiple parts sitting on a flat plate, but the motions are not stable and require additional sensing in the form of a vision system to keep parts on their desired paths. In this chapter, individual part motions are not considered, but the planar robot vibrations are combined with a spiral ramp to provide a velocity field that will tend to singulate parts. The spiral structure allows for a recirculation and reorientation effect. The most novel aspect of this application is that the planar robot can use the actuators that are used for vibration to move large distances, bringing the parts to the assembly location rather than requiring the manipulator robots to make large motions to visit fixed parts feeders. This feature is especially useful in the minifactory application, where the overhead manipulators have deliberately limited workspaces. This parts feeding application is an example of a very different type of robotic manipulation. It can be considered distributed manipulation [105] because multiple parts are manipulated simultaneously over the surface of the feeder body. It can be considered minimalist because of the efficient use of actuators and the limitation to planar motions.

Chapter 9 summarizes the results and contributions of this dissertation. The performance improvements gained by closed-loop control are summarized, including an improvement in repeatability from 6 μm to 2 μm, a decrease in settling time from tenths of seconds to below 20 msec. More importantly, with closed-loop operation the robot will not miss teeth or over-rotate if the motions are not carefully planned, enabling a much greater variety of tasks, including wall following, platen joint detection, visual servoing, and force, stiffness, and impedance control. The minifactory and parts feeding applications would be impossible without closed-loop operation.

1.4 Experimental systems

To simplify the presentation in later chapters, this section describes the various systems used to conduct the experiments described in this document. This section is designed to serve as a useful reference for details of the experimental systems, both for readers and for future users of any of these systems. The reader may wish

to skim this section for now if it seems like excessive detail and refer back to it as needed.

1.4.1 *normag0* system

The first system, which will be referred to as *normag0* in this document, was used for modeling and control experiments before a functional prototype of the CMU platen sensor was developed. Much of the equipment was donated from IBM. The forcer is a small Normag model 4XY1302-2-00 (serial number AS302). These motors have four actuators, as shown in Figure 1.10, but the actuators in opposite corners are wired together in series, so that it only has four input pairs (a sine and cosine phase for both the x and y directions). This configuration restricts control to two axes, eliminating the ability to actively change the motor's skew angle. These motors are designed for open-loop control, where one would not normally want to change this angle. However, full closed-loop control must be able to generate torques, so the motor was rewired for this research so that each of the four actuators could be independently controlled, now with eight input pairs (a sine and cosine phase for each of the four actuators).

The modified forcer was driven by a Motion Science MSSX 16006-4 stepping motor amplifier. This model number is a custom version of their standard MSFX 16006-4 model, with the micro-stepping circuitry removed. It is simply an eight channel pulse-width-modulation (PWM) amplifier designed for use with the types of inductive loads found in stepping motors. As supplied, it had a current limit of 6 A, which exceeds the recommended forcer limit of 2 A by a large enough margin to be of concern. It was necessary to remove several power resistors on each amplifier board to reduce the current limits to 3 A. The PWM outputs have substantial filtering to prevent high frequency currents from entering the motor coils, which can cause destructive levels of inductive heating.

A small VME rack with a Motorola MVME-162 68040 real-time computer board and a quad-I/O SBX carrier board with analog-to-digital (ADC), digital-to-analog (DAC) converters, and digital-input-output (DIO) boards was used. The VxWorks [106] real-time operating system was used, with a Sun workstation development platform.

To provide a measurement of the courier position, a Zygo ZMI-1000 Interferometer system was integrated into the system. This system consists of a laser head, interferometer optics, and a control box containing a small VME card cage, the laser head power supply, and three VME boards for controlling and collecting data from the laser system. The system is capable of sampling measurements for two axes at well over 10 kHz at a resolution of 2.8 nm. Initially, the VME cards were installed directly into our VME card cage, allowing for high-bandwidth, low-latency access to the laser measurement data over the VME bus. This system worked well for months, but measurement faults began to occur frequently. The manufacturer noted that the Zygo VME boards were very sensitive to power supply noise and may not work in non-Zygo card cages. The Zygo VME boards were then re-installed in the Zygo box and a second quadIO board was installed in the VME cage with three DIO boards for interfacing with the Zygo system digital interface, again allowing high-bandwidth, low-latency access to the measurement data. These data was used to provide a translation and rotation measurement for a prototype 2-DOF PD controller (using the laser interferometer measurements to simulate the platen sensor).

The *normag0* system was used for the commutator phase advance experiments of Chapter 2, and both the static and dynamic actuator modeling experiments of Chapter 3.

1.4.2 *normag1* system

The second system, *normag1* is the first (in the world) with an integrated platen sensor. The motor housing is a Normag model 4XY1304-2-01 (serial number A10), which has eight actuators, as shown in Figure 1.1. However, the adjacent actuators are wired in parallel, making them function as one double-size (and force) actuator, so that only eight input pairs are required, as with the *normag0* system. This motor is also designed for use in an open-loop stepping mode. However, the platen sensor developed at CMU is compact enough to fit in unused space between the four actuators. A pocket was machined into the aluminum motor housing, and the sensor was installed and potted with epoxy. The excess epoxy was machined off and the bottom of the forcer was lapped smooth. A circuit board for amplification, filtering, and processing of the sensor output signals was installed in the forcer body,

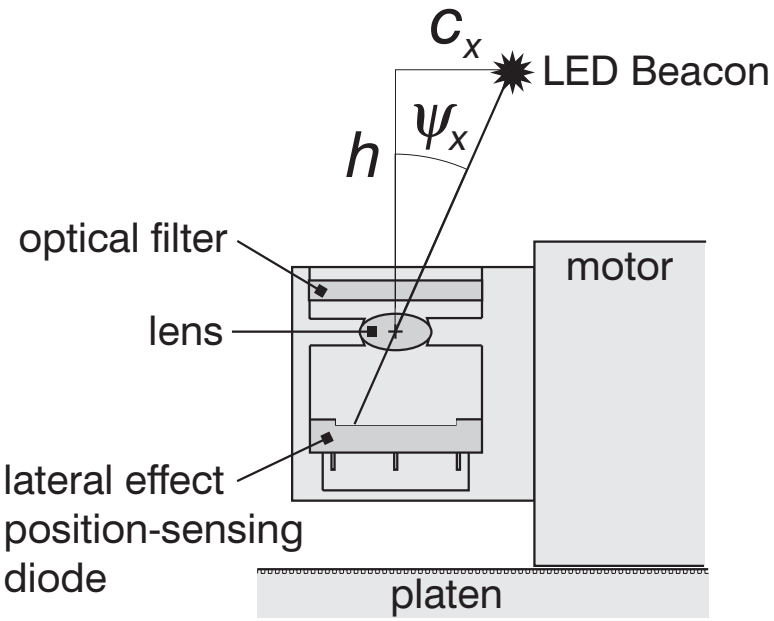


Figure 1.14: Coordination sensor schematic

connecting directly to pins protruding up from the sensor. The board supplied amplified outputs that were sent along the tether to the control computer[4].

The same Motion Science amplifiers were used for this system, although this forcer has a peak current of 4 A because the double actuators are wired in parallel. To further complicate matters, it was necessary to switch between the *normag0* and *normag1* system while the newer system was under development. The output connections of the amplifier were modified to provide jumpering on the forcer side of the connection, allowing the power resistor values to change depending on the requirements of the forcer connected to the amplifier.

After an evaluation of real-time operating systems and hardware as part of the larger minifactory project, it was decided to switch to the POSIX-compliant LynxOS real-time operating system and more modern PowerPC hardware using a standard ATX form-factor motherboard.

A *coordination sensor* [107] was integrated into this system[5]. This sensor attaches to the side of the forcer, as shown in Figure 1.16 and can detect overhead infrared LED beacons. As shown in Figure 1.14, it has a semiconductor lateral photoeffect

[4]The sensor development and integration was performed by Zack Butler, with help from Alfred Rizzi and Ralph Hollis.

[5]The coordination sensor was developed by Wing-Choi Ma and integrated into the system by Zack Butler, with guidance from Alfred Rizzi and Ralph Hollis.

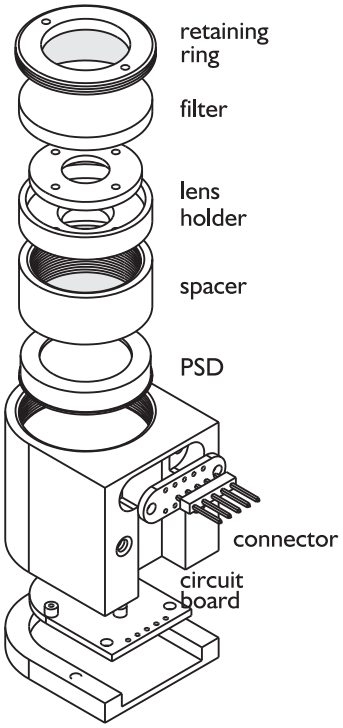


Figure 1.15: Coordination sensor exploded view

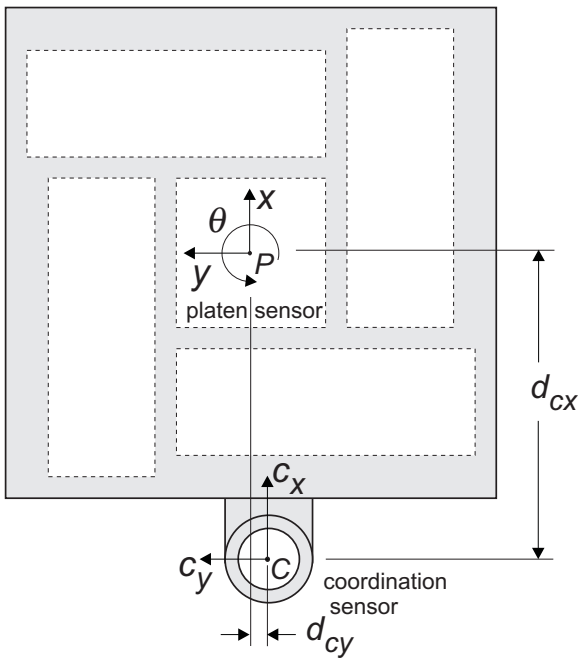


Figure 1.16: Coordinate sensor mounting parameters and sensing axes (top view)

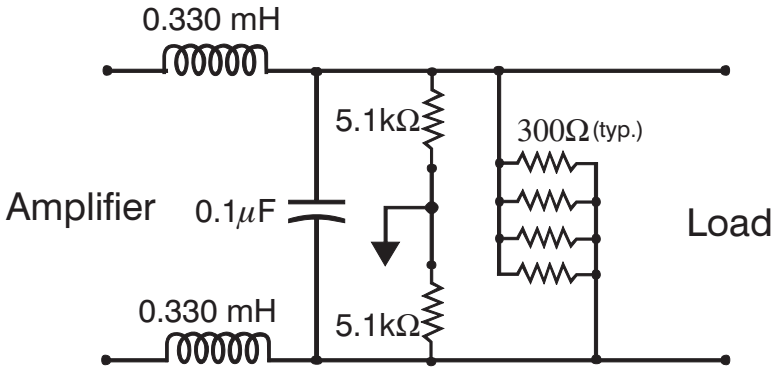


Figure 1.17: Output filter for Copley amplifiers

position sensitive detector (PSD) along with a lens to focus the light onto the surface of the detector. The beacons are modulated to reduce sensitivity to other overhead light sources. This sensor has sub-micron resolution and micron-level accuracy over a range of tens of millimeters and a bandwidth of hundreds of Hz [108]. It is intended for precise localization of low-DOF overhead robots that are outfitted with beacons on their end effectors. The planar robot can then precisely localize the mounting position of these robots during system initialization. This localization is necessary for later cooperative tasks involving both the planar robot and the overhead devices.

The *normag1* system is used in the platen sensor and coordination sensor autonomous calibration experiments of Chapter 4, the PD/PID controller experiments of Chapter 5, and the parts feeding experiments of Chapter 8.

1.4.3 *normag2* system

The third system, *normag2*, is for the most part a duplicate of the *normag1* system. It has an integrated platen sensor and the motor housing is a Normag model 4XY1304-2-01 (serial number A09), as shown in Figure 1.11. The two major differences between it and the *normag1* system is that it lacks a coordination sensor and it uses different amplifiers. Eight Copley model 4212 amplifiers are used to drive the planar motor actuators. It was necessary to install a filtering circuit, shown in Figure 1.17 on each amplifier output to prevent high-frequency current fluctuations that cause significant inductive heating of the platen and significant platen sensor noise. The *normag2* system is used in the dynamic force modeling experiments of Chapter 3, and the actuator autonomous calibration experiments of Chapter 4.

1.4.4 *vole* system

The *vole* system uses the planar robot body from the *normag1* system, with the Normag model 4XY1304-2-01 (serial number A10 forcer), integral platen sensor, and coordination sensor. However, the amplifiers and computing hardware are dramatically different. Eight Copley model 4212 amplifiers, a PowerPC motherboard, plumbing and controls for air and vacuum, and various custom printed circuit boards are integrated into a compact "brain-box," that can be mounted on the side of the platen support frame.

1.4.5 Minifactory components

Custom couriers and platens are under development in the Microdynamic Systems Laboratory for use in the minifactory system. The platens differ from commercial platens in that they have a metric tooth pitch (of 1 mm), the teeth are fabricated by a machining process rather than a chemical etching process, and can be clamped together to form an extended, continuous operating surface for the planar robot couriers. Couriers are also redesigned, with three teeth per pole compared to four teeth in most commercial forcers, which allows for a compact size while also allowing for a slightly larger platen sensor to be integrated in the middle of the forcer body. Although these devices were not used in the work of this dissertation, most of this work will be applied to these systems as they are completed.

1.5 Notation

Commonly used variables are listed in this section. Less common variables will be introduced as needed in the text.

Physical parameters	
m	mass [kg]
I_z	rotational inertia [kg·m^2]
M	diagonal mass matrix [kg] or [kg·m^2]
ρ	platen tooth pitch [m]
ω_ρ	pitch frequency $2\pi/\rho$ [rad/pitch]
d_{gx}, d_{gy}	location of center of mass of forcer [m], [m] (Figure 1.8)
d_a	actuator distance from center [m] (Figure 1.8)
d_f	forcer body width [m] (Figure 1.8)
d_{wa}	actuator width [m] (Figure 1.8)
d_{so}	sensor segment offset [m] (Figure 1.8)
d_s	spacing between centers of sensor segments [m] (Figure 1.8)
d_{cx}, d_{cy}	coordination sensor mounting offsets [m], [m] (Figure 1.16)
o_x, o_y	coordination sensor zero position offsets [m], [m]
Kinematic, state, and related intermediate variables	
x, y, θ	labels for the three planar axes of the robot
s_i	raw sensor output [counts]
z_i	raw sensor offset [counts]
ϕ_s	spatial phase of sensor segments [rad]
ϕ_a	spatial phase of actuators [rad]
ψ_a	spatial phase input for (open-loop) commutator [rad]
p_s	positions of sensor segments [m]
p_a	positions of actuator segments [m]
q	planar robot pose 3-vector [m m rad]
$\dot{q}$	planar robot velocity 3-vector [m/s m/s rad/s]
$\ddot{q}$	planar robot acceleration 3-vector [m/s^2 m/s^2 rad/s^2]
c_x, c_y	coordination sensor beacon distances [m], [m]
ψ_x, ψ_y	coordination sensor beacon angles [rad], [rad]
h	coordination sensor beacon height [m]
v	scalar velocity [m/s]
Force, control, and related intermediate variables	
I, I_c, I_A, etc.	current [A]
f	4-vector of actuator forces [N]
u	planar acceleration command 3-vector [m/s^2 m/s^2 rad/s2]
u_w	planar wrench command 3-vector [N N Nm]
w	planar wrench 3-vector [N N Nm]
f_x, f_y, τ	elements of wrench vector [N], [N], [Nm]
J	force Jacobian
(continued)	

Force, control, and related intermediate variables *(continued)*	
w_s	state-dependent controller terms
w_t	time-dependent and disturbance compensation controller terms
b_i	linear damping coefficient [N/m/s]
$b_v(\cdot)$	actuator force damping model [m/s $\rightarrow$ N]
$b_w(\cdot)$	actuator force rolloff model [m/s $\rightarrow$ N]
K	controller gain matrix [N/m] or [Nm/rad]
T_d	controller derivative gain matrix [s]
T_i	controller integral gain matrix [1/s]
T	controller or commutator sample time [s]
Actuator saturation	
$\mathcal{F}$	Set of achievable wrench outputs
$\Phi(\cdot)$	norm-like function for specifying $\mathcal{F}$
$p_{\mathcal{V}}$	parameters for boundaries of $\mathcal{F}$
$\mathcal{E}$	the set that bounds $\mathcal{F}$
$\mathcal{F}'$	modified set of achievable wrench outputs
$\mathcal{D}$	set containing all possible feedforward/integral wrenches
$\mathcal{S}'$	set of unsaturated states
$\mathcal{U}$	set of unsaturated wrench outputs
$\Upsilon(\cdot)$	norm-like function for acceleration limits
Superscript, subscript, and overstrike conventions	
$\bar{\cdot}$	estimated quantity
$\tilde{\cdot}$	error quantity
$\breve{\cdot}$	uncalibrated quantity
$\hat{\cdot}$	calibrated quantity
$\cdot^+$	maximum allowable value
$\cdot^-$	minimum allowable value
$\cdot^*$	desired value
$\cdot'$	used for an alternate definition of a variable (does *not* signify a derivative operator)
$\cdot_i$	ith element of vector
$\cdot_x$	element of vector corresponding to x axis
$\cdot_y$	element of vector corresponding to y axis
$\cdot_\theta$	element of vector corresponding to θ axis
Operators	
$\|\cdot\|_V$	$(\cdot^T V \cdot)^{1/2}$
$D\cdot$	partial derivative operator
Miscellaneous	
i, j, k	indices
t	time [s]

Chapter 2

Commutation

Each of the four linear motors in the forcer require two current inputs. In total, eight currents flow into the actuators, producing a net x, y force and a torque. The force generated by each motor is a function of its current values and its position and skew angle. A commonly-used first-order model of the force generated by a single linear motor actuator is

$$f = kI_A \sin(\phi_a) + kI_B \cos(\phi_a), \tag{2.1}$$

where f is the force generated by the actuator, k is a motor constant with units of force over current, $I_{\{A,B\}}$ is the current in coils $\{A, B\}$, and ϕ_a is the *spatial phase* of the actuator. This phase is the position of the actuator expressed as an angle (in units of radians),

$$\phi_a = 2\pi p_a/\rho, \tag{2.2}$$

where ρ is the tooth pitch in meters and p_a is the position of the actuator in meters, such that a spatial phase of 2π is equivalent to one tooth pitch. This force expression neglects second-order effects such as manufacturing errors, magnetic saturation, non-uniform teeth, and forcer rotations. Their importance will be investigated in the following chapter. It also does not consider dynamic effects such as amplifier bandwidth, commutator update rate, and the electrical and magnetic dynamics, which will also be investigated in the following chapter.

In principle, a control strategy can be designed that directly operates on the force model (2.1). The controller will need to select I_A and I_B based on the motor phase ϕ_a and desired force f_{des}. Note that there will not be a unique solution for the currents unless ϕ_a is a multiple of $\pi/2$ (corresponding to positions where one of the

motor quadrature pairs is at an equilibrium and cannot generate any force). More interestingly, the force output of the actuator is constrained by the limits on the individual coil currents, so that if both currents are fixed to their limits, the peak force of the actuator will vary with the phase. In particular, at $\phi_a = \pi/4$ rad, the maximum force will be 141% of that available at $\phi_a = 0$ rad. However, it is not clear that this additional force can be exploited usefully by a controller. If this additional force was used for motions at top acceleration, the 41% variation in force may cause significant audible noise and oscillations in the motion. If it were used to generate a large static force, it would result in a detenting behavior, where the forcer could only generate peak forces at certain positions, i.e. where ϕ_a is near a multiple of $\pi/4$ rad.

For simplicity in the model and controller, and to make it easy to describe the capabilities of the planar robot, the currents will be chosen such that the maximum force will not vary with position and will be given by the maximum force available at $\phi_a = 0$. The current magnitudes should not exceed the current limit, I_m, and for a given desired force the currents should vary smoothly with motor position. If the currents are computed as

$$I_A \;=\; I\cos(\psi_a) \quad \text{and} \quad I_B \;=\; -I\sin(\psi_a), \tag{2.3}$$

where I and ψ_a are the new parameters with units of amperes and radians, then the force will be given by

$$f \;=\; kI\sin(\phi_a - \psi_a). \tag{2.4}$$

To generate the peak force, set $I = I_m$ and choose ψ_a as a function of the motor position, $\psi_a = \phi_a \pm \pi/2$.

This parameterization of the coil currents has been used for open-loop operation of planar motors. However, the motor positions are not known, so forces cannot be explicitly commanded. Instead, I is set to I_m and ψ_a is used to set the equilibrium phase $\phi_{a,eq}$ of the motor by choosing $\psi_a = \phi_{eq}$ so that $f_{gen} = 0$ at $\phi_a = \phi_{a,eq}$. In practice, the linear motors are driven by stepper drivers that compute (2.3) using embedded lookup tables, and simply advance ψ_a a small fixed amount for every pulse input they receive.

For closed-loop operation, it is more common to command the actuator force than the equilibrium position. Parameters I and ψ_a must be chosen as a function of

motor position ϕ_a to generate the desired force. Put another way, we want to invert a model of actuator force output $f(\psi_a, \phi_a, I)$ with respect to (I, ψ_a). As the actuator model is a many-to-one mapping (i.e. many (I, ψ_a) combinations will produce the same force output), the commutator must use some heuristic or secondary criterion to select the output currents. Two examples are shown below, and one is selected based on its simplicity and efficient use of current.

A *fixed amplitude* approach can be used that, like open-loop operation, keeps I at its maximum value, but computes ψ_a to generate the desired force, $\psi_a = \phi_a - \arcsin(f_{des}/kI)$. This commutation scheme requires large currents to be in the motor coils at all times, resulting in unwanted thermal effects that lead to accuracy degradation through platen deformations. In addition,this approach will choose setpoints at locations on the force curve that have steep slopes. If there is noise in the position measurement or significant motion between commutator updates, the actuator output force will vary significantly, introducing high-frequency disturbances into the system.

Instead, a *fixed phase* approach will be used. It fixes ψ_a relative to ϕ_a as $\psi_a = \phi_a - \pi/2$. The force is controlled by varying the amplitude I according to $I = f_{des}/k$. The current I is then proportional to f_{des} and no currents are required if $f_{des} = 0$, significantly reducing thermal effects. In addition, it drives the forcer at the peak of the force curve, so that, to first order, position changes have no effect on the force output. Note that with this commutator, the controller supplies input I as a linear function of the desired force, just as it would for a conventional servo motor.

For simplicity and to ease the integration of enhanced models, the equations

$$\begin{aligned} I_A &= I\cos(\phi_a - \pi/2) \\ I_B &= I\sin(\phi_a - \pi/2) \end{aligned} \tag{2.5}$$

are implemented in software, forming a software-based *commutator*. In the context of planar linear motors, a commutator is simply a mechanism that selects motor currents to achieve a desired force. The usefulness of a commutator is in providing higher level controllers with an ideal force source (within actuator limits), as opposed to a system where a current must be specified for each coil. Commutators separated from higher level controllers are especially important for planar linear motors because currents must be commanded at kHz rates. For example, to operate the *normag0* robot at 1 m/s, a sinusoid of nearly 1 kHz must be applied to the motor coils, calling for updates of at least 2 kHz.

Commutation has traditionally been implemented in hardware, either by mechanical electrical connections (brushed DC motors), or analog or custom digital electronics (brushless DC and stepper motors). Here, in contrast, the software commutator uses the same general-purpose real-time computing hardware that executes controllers and higher level code for the robot. The decision to use software commutation has implications on the commutator implementation, one of which is discussed in the next section. It should be noted that it would be straightforward to implement the commutator using a dedicated DSP or FPGA if it were necessary to reduce usage of the real-time computing hardware or for commutation rates even faster than those required for our applications. Note that the platen sensors would also have to be read and the positions of each motor computed at this faster rate to provide a position measurement output.

2.1 Phase advance

Real-time issues became apparent in initial tests of the *fixed phase* commutator described in the previous section. While open-loop controllers were able to drive motors to velocities over 1 m/s, when a large force command was sent to the commutator, the motor would not reach half that speed.

The commutator did not perform as well as the open-loop controller because the commutation loop contains delays, making the motor position sensed at the beginning of the cycle invalid by the time the output is produced. This effect is illustrated in Figure 2.1. The force curve represents the force generated by the actuator as a function of the actuator position. The commutator shifts the phase of the force function once per cycle based on the latest position estimate, and adjusts the amplitude to generate the proper force. The commutator wants to set the phase such that the motor is operating at the peak of the force function. However, while the sensor signal is being processed and new outputs take effect, the motor will have moved, causing the output to miss the peak of the force curve by an amount $v \cdot t_l$, the velocity of the motor times the latency. In addition, since the updates occur at discrete times, the motor force over the next output cycle will span a range of the force curve of length vT, where T is the sample time in seconds. Instead of the commanded force, there will be a reduced force that changes during the output interval (the bold arrow). With a phase advance term of $vt_l + \frac{vT}{2}$, the bold arrow

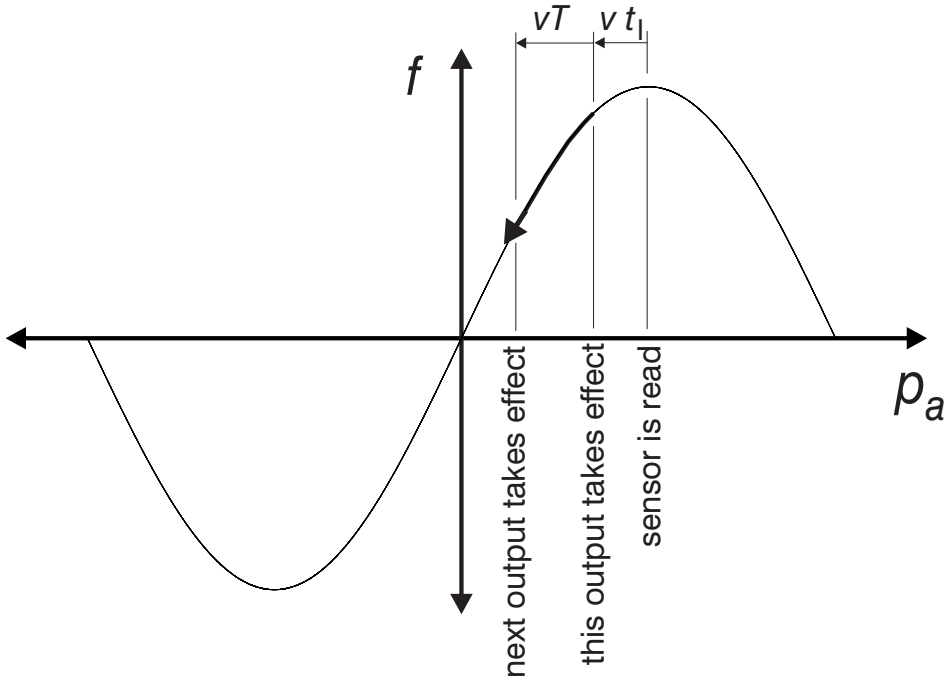


Figure 2.1: System latencies and control rates cause the motor's output force to drop at high velocities.

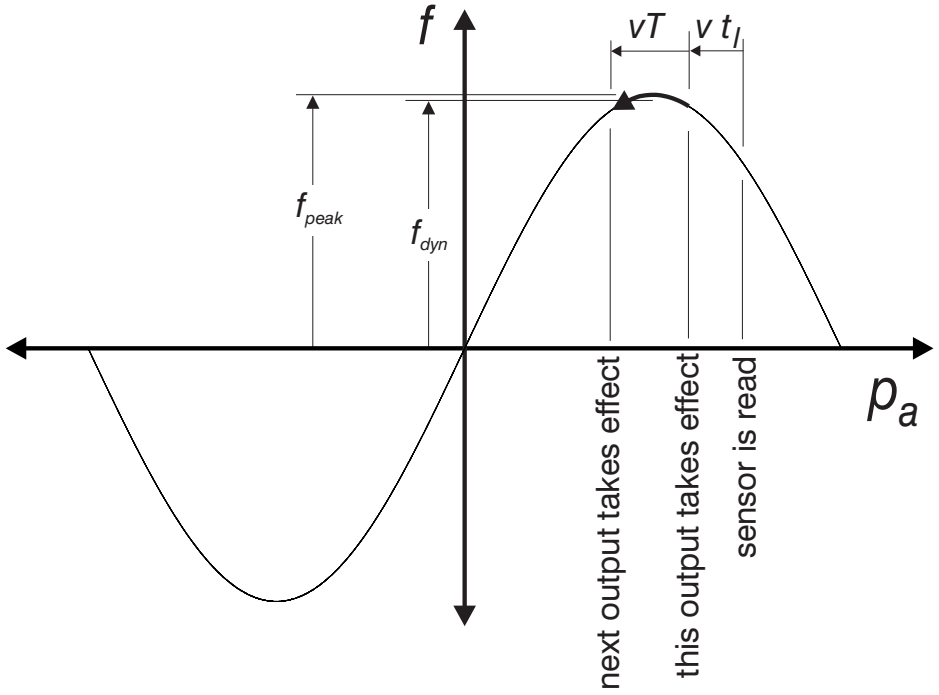


Figure 2.2: Even after compensation, the dynamic force output, f_{dyn} will be lower than the peak static force, f_{peak}.

can be shifted so that it is centered about the peak of the force curve, as shown in Figure 2.2. The approach is to determine the latency time and commutation rate of the commutator and compute a delay time (in units of seconds). This value is multiplied by the latest velocity to get a phase advance position offset, which is added to the latest position. The resulting sum is the best estimate of the average motor position (assuming constant velocity) over the time period when the next output will be in effect. After compensation, the resulting dynamic force will still be reduced relative to the peak static force, as shown in Figure 2.2. The resulting dynamic force f_{dyn} can be estimated based on the peak static force f_{peak}, velocity v, and sampling time T,

$$\begin{aligned} f_{dyn} &= \int_{-a}^{a} \frac{f_{peak}\sin(\phi_a)}{2a} d\phi_a \\ &= f_{peak}\left(-1 + \frac{\sin(a) - \sin(-a)}{2a}\right), \end{aligned} \tag{2.6}$$

where $2a = 2\pi vT/\rho$ is the distance traveled between samples, expressed in units of spatial radians.

Commutator latencies include input and output delays and computation times. The experiments in this section were implemented on the *normag0* system. This work predates completion of the platen position sensor, so a high-speed Zygo laser interferometer system was used as a stand-in for two axes of the platen sensor. As the Zygo interface boards and real-time computer shared a VME backplane, the input delay is negligible. The commutation code required 200 μs (on a Motorola MVME-162 68040 running the VxWorks real-time operating system) to compute output currents based on the input position.

The amplifiers introduce an output delay which is slightly more complicated to measure. To do so, a spectrum analyzer was used to measure the transfer function of the amplifier driving one motor of a stationary forcer floating on the platen. Note that the amplifier dynamics can be reasonably modeled by a constant delay time as long as the lowest frequency poles and zeros are above the driving frequencies of interest. If this is not the case, the delay time will have to vary with the commanded velocity. For our Motion Science amplifier, the lowest pole appears to be at 1400 Hz, yielding a constant delay time of 114 μs.

The last delay is from the commutation sample time, 1/3500 s for these tests, giving a total delay time of

$$\frac{1}{2} \cdot \frac{1}{3500} + 114 + 200 = 457\mu\text{s}.$$

This value is the delay time that is multiplied by the motor velocity to compute the amount of phase advance.

2.2 Experimental verification

It is perhaps naive to believe the schematic diagram in Figure 2.1 is an accurate representation of the actuator force output at high velocities. At even 0.5 m/s, a 3500 Hz commutator update rate only provides seven updates per cycle. The amplifier dynamics become important, with the coil currents (I_A, I_B) not changing instantaneously, but undergoing independent step responses, which will not generally produce a nice step response in the overall output force. Nevertheless, simple models are preferable, and this section investigates how well the phase advance model matches reality.

To validate the estimated phase advance delay time, full scale force commands were sent to a *fixed phase* commutator controlling the x motors of the forcer, and the position data was recorded. This test was repeated for a range of values for the phase advance delay time. Force commands are generated such that the motor will accelerate forward and backwards at maximum acceleration between velocities of -1 m/s and 1 m/s several times[1]. The complete sequence typically takes less than 0.5 s, with an overall motion of less than 50 mm.

One complication with this experiment was the use of the interferometer, which could only measure the x translation and skew angle. Because of the large x accelerations, it was necessary to have a servo loop to keep the skew angle small, but the x actuators could not be used without affecting the experiment. The y actuators had to regulate both the skew angle and the unsensed y position. The solution was to use a PID angle regulation controller with a *fixed amplitude* commutator using the y motors only. Recall that the fixed amplitude commutator maintains large currents

[1]While the forcer can move faster, the interferometer (which has to increment a counter for every 2.5 nm of motion) has a limit of about 1 m/s.

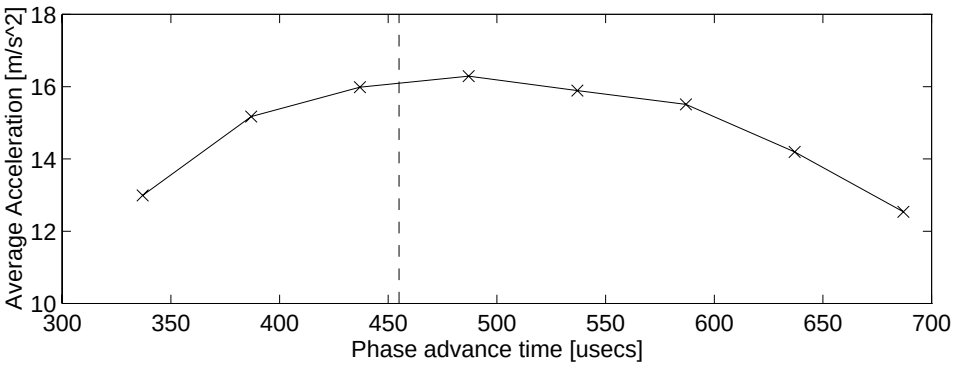


Figure 2.3: Average acceleration as a function of the phase advance time. The computed value is indicated by the vertical dashed line.

in the actuators, allowing the y translation to be locked in an open-loop sense even as the y actuators regulate the skew angle. Further details of this controller [61] are omitted here as the 3-DOF platen sensor would allow full 3-DOF control if this experiment were repeated today.

The interferometer x position data for the two long constant acceleration segments are differenced to get velocity. Over repeated runs with varying phase advance delay times, the fastest response indicates the best value for the delay time. To quantify the fastest response, a velocity of 1.6 m/s is divided by the time it takes for the forcer to change from +0.8 m/s to -0.8 m/s, to compute an average acceleration.

Figure 2.3 shows this average acceleration as a function of the phase advance time. The phase advance time computed in the previous section is indicated by the dashed vertical line. Note that it matches the phase advance time corresponding to the peak acceleration to within 50 μs.

To demonstrate the improvement from the phase advance term, Figure 2.4 shows the position and velocity trajectories for a peak acceleration command starting from a rest position. Note that although adding phase advance yields a substantial improvement, it is still significantly below the theoretical maximum acceleration. This remaining error is from damping and rolloff effects discussed in Chapter 3.

It should be noted that phase advance is a common technique used in the control of stepper motors. It has even been previously applied to linear and planar motors [109, 88]. However, these works tend to use phase advance as a heuristic solution to compensate for a slew of dynamic effects and require experimental determination of the phase advance magnitudes.

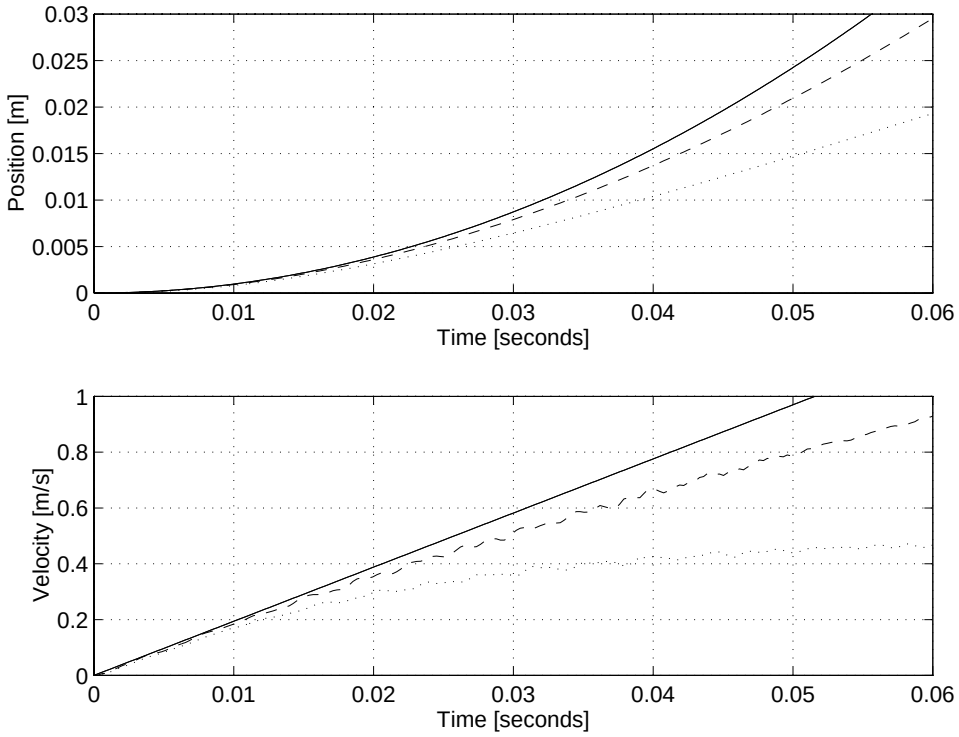


Figure 2.4: Trajectory (solid line) tracking with (dashed line) and without (dotted line) phase advance.

The importance of commutation is in simplifying the models that the controller must consider. With simpler models, simpler controllers can be used or additional higher-level goals can be integrated into the controller. Pushing higher-level goals down to lower levels permits the robot to react faster, improving its ability to dynamically respond to environmental interactions, an important ability for unstructured dynamic manipulation tasks.

Chapter 3

Modeling

```
Depend on the predictability and
steadiness of life to support you!
Lucky Numbers 13, 14, 15, 24, 28, 34
```

With the commutator in place, actuator currents can be automatically selected by sending current commands to the commutator. However, recall that the force model used in Chapter 2 neglected second-order effects such as manufacturing errors, magnetic saturation, non-uniform teeth, imperfect amplifiers, finite update rates, and forcer rotations. The commutator also uses a static force model, neglecting effects from the amplifier bandwidth, the commutator update rate, and the electrical and magnetic dynamics. Before proceeding to attach controllers to the commutator, it is necessary to examine these effects.

A skeptical reader may wonder why the first-order actuator model is not sufficient. After all, open-loop controllers have operated successfully with such a model.

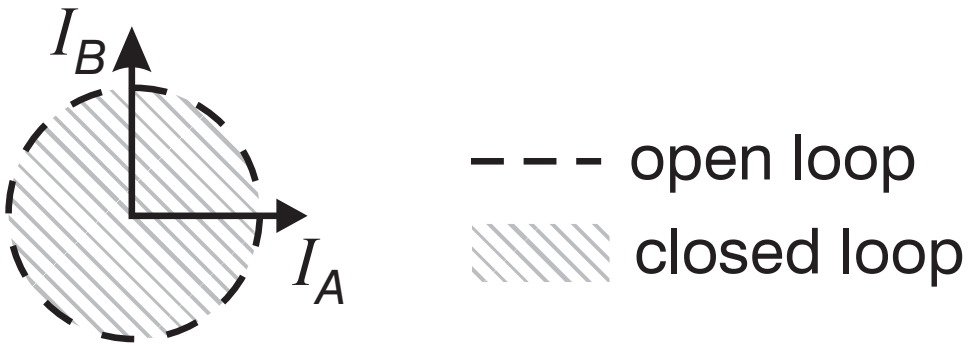


Figure 3.1: Closed-loop operation demands a larger domain of coil currents than open-loop operation.

The first reason is that the commutator uses a much larger part of the domain of (2.1) than an open-loop controller. As shown in Figure 3.1, the coil currents under open-loop control are constrained to a circle, while the commutator coil currents can be anywhere within the disk bounded by that circle. If a first-order model is accurate only on the circle, then it will be invalid for closed-loop control. Furthermore, there have been few published results that attempt to validate the first-order model over any domain.

However, the stepping motor community has not ignored planar linear motors. The main focus of much work is to develop a model of the force generated by a toothed motor as a function of the currents in the motor coils and the motor position. Given the magnetic and geometric properties of the motor materials, and some reasonable assumptions, it is straightforward to compute an estimate of the total magnetic flux passing through the motor teeth for a given set of coil currents. To compute the force produced by this flux, it is necessary to know the flux pattern of the magnetic field in and around the teeth, which is difficult given the sharp edged geometry. In [36, 86, 97], various assumptions are made about the flux field pattern, but all eventually fit their results to a sinusoidal stiffness function proportional to the coil current. If a sinusoidal function were assumed *a priori*, these derivations would only be useful in giving an expression for the proportionality constant. These works provide no experimental validation of their models, and little sense of how well they match reality.

However, certain deviations from the sinusoidal model have been considered. Magnetic saturation in the motor materials causes the amplitude of the sinusoid to be a nonlinear function of coil current. This effect was shown to introduce a spatially periodic open-loop positioning error at a fourth harmonic of the tooth pitch distance in [36]. However, only qualitative effects on open-loop control were considered. Another important effect is the decrease in force capability with increasing skew angle. In [74], a model of this effect is presented. Under some complicated assumptions, an analytical model involving several nested transcendental functions is derived. However, experimental results did not show this model to be any better than a much simpler cosine dropoff function.

Another property of the actuator model is eddy current damping. Eddy currents are induced in the platen surface whenever the magnetic flux traveling through the surface is changed. This change occurs when the forcer moves, exerting flux

through a different area of the platen, or when the currents in the coils change, exerting different flux in one area of the platen. The currents act to resist the flux changes, and are dissipated by resistive losses in the platen. The net effect of eddy currents is that of a damping force. This effect is especially large in planar linear motors because the normal method of constructing the stator out of laminations to reduce eddy currents cannot be easily applied to the 2-dimensional platen. In [36, 86, 31], engineering equations are used to compute an equation for the eddy-current damping force as a function of velocity. With the assumption that the flux changes are proportional to the motor velocity, these equations are linear with respect to velocity.

Electrical dynamics may also be important. The motor coils typically have relatively large inductances (which vary with position and current) of several mH. For planar linear motors, pulse-width-modulation (PWM) current amplifiers are used to drive the coil currents. These amplifiers operate by varying the duty cycle of a high-frequency square wave voltage output. The PWM frequency should be chosen much higher than the frequency of controller outputs sent to the amplifier. The rail voltage should also be chosen high enough to ensure that the effective current slew rate is sufficient for the controller bandwidth. If these conditions are not met, delays and nonlinear distortions of the commanded waveforms will introduce additional dynamics and nonlinearities into high-frequency actuator outputs.

There are significant thermal effects generated by the motor coils. For open-loop operation, each pair of coils may generate 4-6 W of heat from resistive losses, significantly more from eddy current losses, and some commercial motors are rated to run at up to 110°C [18]. For precision assembly operations, the thermal deformations of the platen may become important, since the platen sensor measures the forcer position relative to local platen teeth. In [26], a finite element model is used along with the time history of forcer positions to compute platen displacements. Platen deformations of nearly 60 μm were reported; the model was able to compensate to under 12 μm. However, with closed-loop control, large currents are only used during motions, for disturbance rejection, and possibly for generating large static forces. In this case, platen expansion from heating should be much less a problem.

For the actuators and sensor, the sharp corners of the teeth eliminate the possibility of analytic magnetic models. The 3-D geometry makes assuming an *a priori* flux pattern or a finite element model questionable. Furthermore, even if the initial

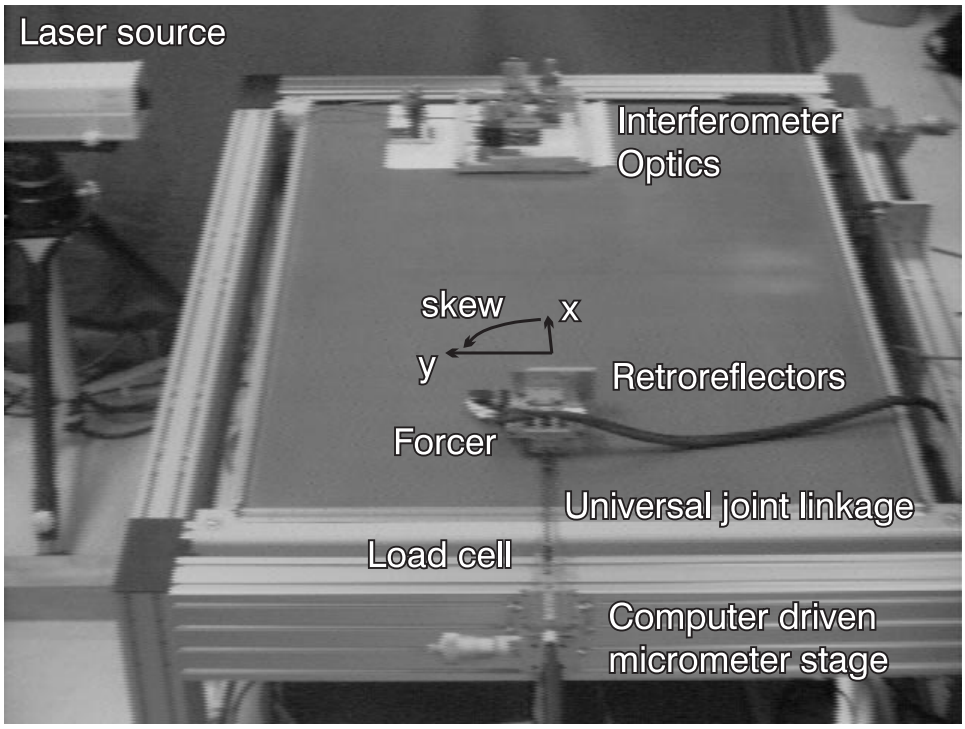


Figure 3.2: Test system for static force measurements.

material properties are known, they may change during fabrication processes. Unmodelable manufacturing imperfections such as bent teeth and pitch variations will also affect the model.

Given the above difficulties, an empirical approach to modeling is used here. As with the commutation experiments, the particular forcer used for the experiments of this chapter is the *normag0* motor, shown in Figure 1.10. As before, it does not have an integrated platen sensor, as this work was performed before sensor development was complete. Instead, a 2-DOF (translation and skew angle) high-speed laser interferometer is used to supply partial position measurements for the commutator.

This chapter is divided into two parts. In the following section, the static force characteristics of the linear motors are examined, while Section 3.2 explores their dynamic force characteristics.

3.1 Static Force Characterization

In this section, the fidelity of the first-order force model (2.1),

$$f = k_A I_A \sin(\phi_a) + k_B I_B \cos(\phi_a), \tag{3.1}$$

with the coil currents reparameterized using (2.3)

$$I_A = I\cos(\psi_a) \quad \text{and} \quad I_B = -I\sin(\psi_a), \tag{3.2}$$

is investigated for cases where the motor velocity is near zero. Manufacturing imperfections in the platen or motor may cause a mismatch between the k values or a phase shift between the two transcendental terms, both causing *force ripple*, an unmodeled spatial variation in the force output. Given the complexity of the magnetic circuit, it is also not clear that the force will be a linear function of the coil currents. Nonlinearities not only will cause force ripple, but will interfere with the linearization of the force output, which is one of the main purposes of the commutator.

Note that the actuator force model is influenced by both the forcer and the platen. In particular, if the platen tooth geometry varies significantly across a platen or between platens, force output will be affected and a single actuator model will no longer be sufficient. Here, platen variations are not explicitly considered. If platen variations are significant enough to affect performance, then multiple models (or models that vary with the location on the platen) will be required.

Static force measurements are performed using a *locked rotor* test commonly used in motor modeling. In this test, a constant set of currents is applied to the motor coils. A load is applied to the motor through a load cell, allowing the force to be recorded as the motor deflects from its equilibrium position. The test system is shown in Figure 3.2.

The locked rotor test is slightly more complicated for a planar motor because the air bearing allows for completely unconstrained motion in the plane. A mechanical constraint or rotation servo is required to keep the forcer from rotating. As a mechanical constraint would add friction forces and be difficult to align, the y motors of the forcer are used to regulate the skew angle to zero. A PID angle controller with a *fixed-amplitude* commutator is used for skew angle regulation and passive stiffness in the (unsensed) y direction, as was the case for the phase advance experiments in Chapter 2.

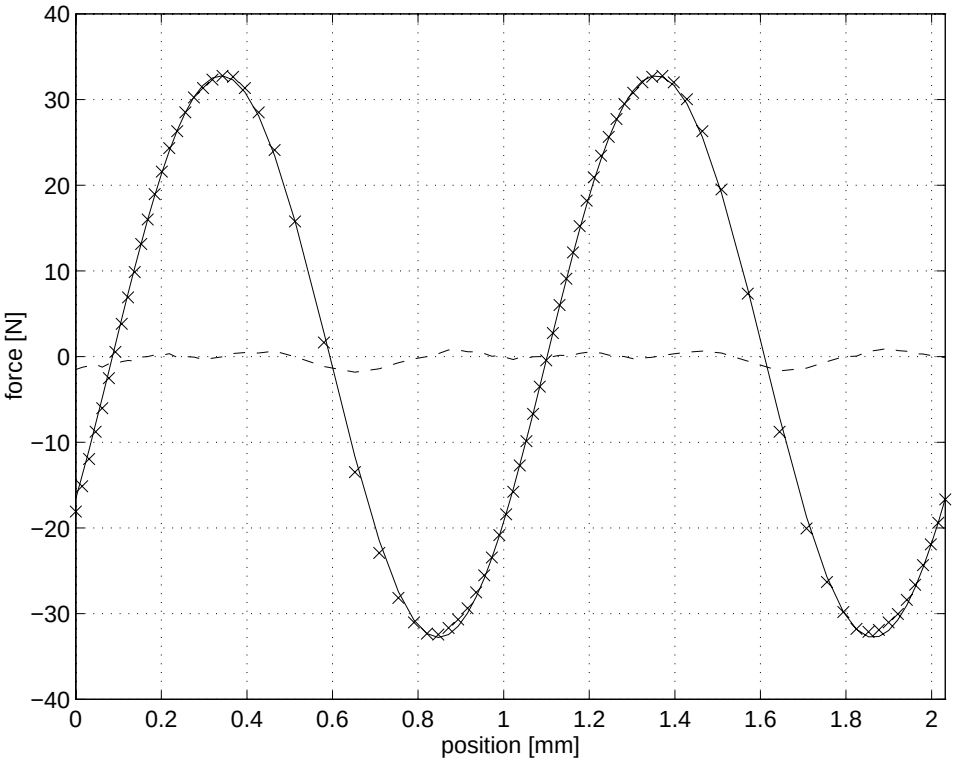


Figure 3.3: Force versus position (ϕ_a), measured data ($\times$) differ from a sine function (solid line) by a small error (dashed line). Here, $\psi_a = 0°$ and $I = 2\text{A}$.

3.1.1 Experimental setup

The experimental setup, shown in Figure 3.2, consists of a Normag forcer on a Normag platen connected through a double universal joint linkage to an Entran ELFTC500-20 load cell. The load cell is mounted on a 3-axis micrometer stage, with the x translation controlled by a Newport model 850 automated micrometer. The coupling between the forcer and load cell is intended to transmit only axial forces, and to attach to the forcer as close to the platen surface as possible to minimize out-of-plane torques that may tend to deflect the air bearing. As there is significant compliance in this coupling, the Zygo two-axis high-speed interferometer is used to monitor the true position and skew angle of the forcer.

For these tests, the current parameter inputs (I, ψ_a) for the x motors of the forcer are set to fixed values. After setting these currents, there is a delay of 10 minutes to minimize thermal expansions during data collection. Next, the micrometer is commanded to move in short increments over a range of two pitch units. After the micrometer settles to each commanded position, the load cell and interferometer

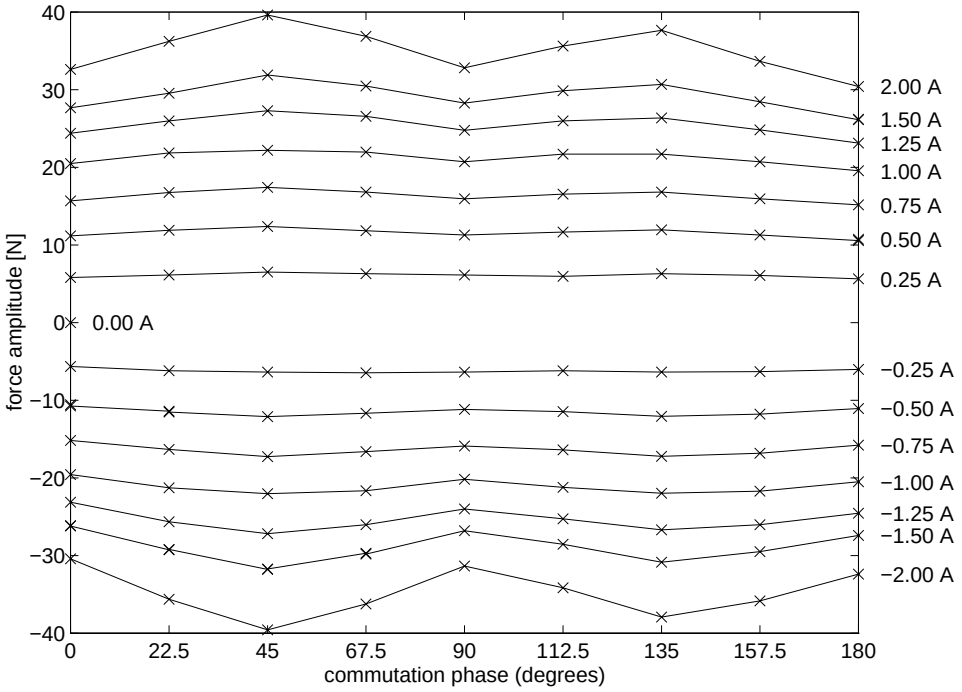


Figure 3.4: Force ripple, showing the peak force capability of the forcer as a function of input parameters ψ_a and I.

are read and the values stored. The force generated by the forcer as a function of position is shown in the force curve of Figure 3.3. This test is repeated with different values of parameters ψ_a and I. The data collection procedure is fully automated, allowing more data to be collected than would otherwise be feasible.

3.1.2 Experimental results

A total of 127 tests were completed to generate force curves for ψ_a from 0° to 180°, and for I from -2.0 A to 2.0 A. For each run, the micrometer was positioned every 25 μm along a 2032 μm (2 pitch) distance. The total data collection time was about 32 hours, taken over several days.

Figure 3.3 shows a typical force curve. Note that the sine function matches the data points[1] to within 1.5 N. Also note that, neglecting higher harmonics, it is only the amplitude of the sine function which is useful, because the phase and period of the sine function are determined by the tooth pitch and the phase ϕ_a. Thus, only the peaks of the force curves are plotted for multiple runs on one diagram, as in

[1]The points are not evenly spaced due to compliance in the force linkage.

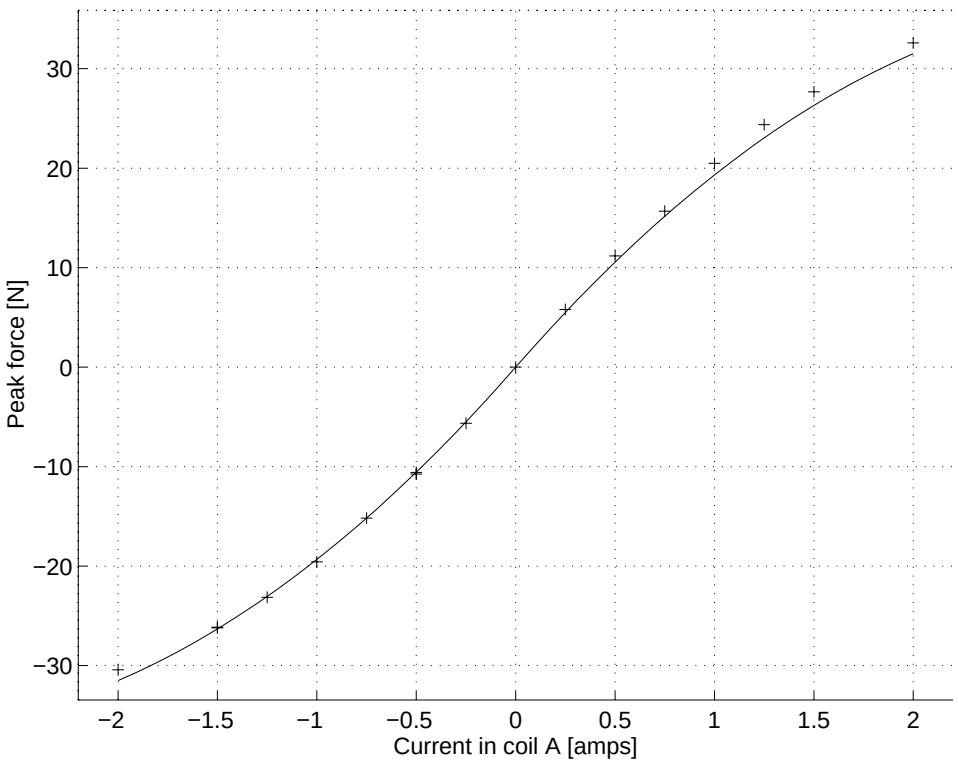


Figure 3.5: Force linearity, showing peak force as a function of I, with ψ_a=0. Measured values (+), and (3.3) fit to these data (solid line) with $a_0 = 22.91$ N/A, $a_1 = -3.57$ N/A^3, and $m = 3$.

Figure 3.4. This figure shows the variation of the peak force of the forcer as the parameters I and ψ_a vary.

This plot shows significant *force ripple* when the motor is commutated with (2.3), exceeding 20% in places. If smoother force generation is desired, these data can be used to correct the motor forces. One approach is to generate a lookup table that gives an appropriate set of currents as a function of the phase angle and desired force, as in [110]. However, this technique requires a complete set of force data to generate the lookup table for each forcer. Based on visual inspection of the manufacturing uniformity of the actuator teeth, it seems unlikely that a force model for an actuator of one forcer would apply to an actuator in another forcer, so that this calibration is required for every forcer. While three days of data collection to get these data once is reasonable, it seems excessive to do this for each forcer that would be needed in a typical planar motor-based assembly system.

Fortunately, we may be able to use a simpler technique. Plotting a vertical cross section of the data for ψ_a=0 produces Figure 3.5. This curve should be close to

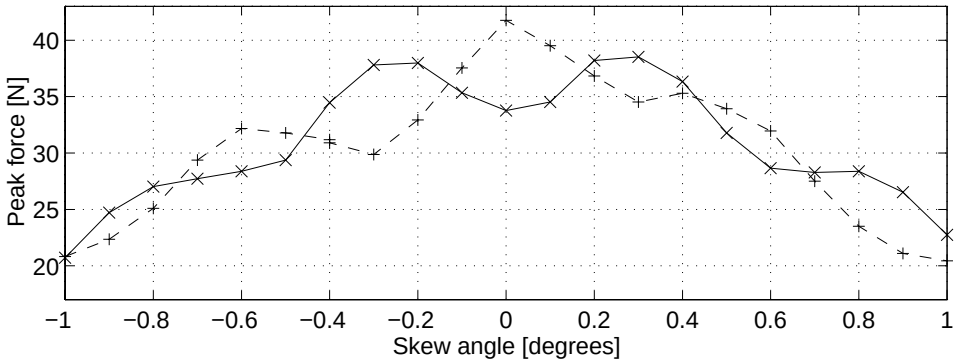


Figure 3.6: Peak force as a function of forcer skew angle, with I=2 A and $\psi_{a1,2} = 0°$ (solid line) or $\psi_{a1,2} = 45°$ (dashed line).

linear if (2.1) is valid. However, there is significant deviation from linearity due to magnetic saturation. This nonlinear function is well modeled by a simple function of the form[2]:

$$f_c = a_0 I_c - a_1 I_c^m, \tag{3.3}$$

where I_c is the current applied to the motor coil, f_c is the peak force generated by the coil, m is a positive odd integer exponent, and a_0 and a_1 are positive parameters that must be determined for each motor coil. The curve connecting the data points in Figure 3.5 is generated by computing the a_0 and a_1 parameters based only on the slope of the data at zero current and the magnitudes at maximum currents. In contrast to a 3D lookup table, the data needed to fit this curve can be taken in a manner of minutes using a hand-held force gauge for a *pull-out* test to determine the peak forces for only four values of I. Early results indicate that the magnetic saturation effect accounts for slightly more than half of the force ripple. The remaining ripple is probably due to higher order terms in the force curves. However, this technique was not investigated further as a more powerful autonomous calibration technique was developed, as discussed in the following chapter.

Force generated by the forcers is also dependent on the skew angle. As the forcer rotates, the motor teeth begin to partially overlap multiple rows of platen teeth, causing the peak force to decrease. To compensate for the motor skew angle, this effect must be included in the force model. The misalignment of teeth will cause a

[2]Chai and Leenhouts[111] have used similar empirical functions to model the saturation in rotary stepper motors.

decrease in k with increasing angle magnitude, and the phases ψ_a of the two motors mounted on opposite sides of the forcer will shift in opposite directions as the angle changes. Thus, (2.4) is replaced by

$$f_{gen} = h(q_\theta)kI\sin(\phi_a - \psi_a). \tag{3.4}$$

To examine the function $h(q_\theta)$, the locked rotor test is repeated, but now using the y motors to servo the forcer skew angle to non-zero values. The data in Figure 3.6 show a force dropoff with increasing angle magnitude, as expected. Each plot also shows an artifact from the uncompensated force ripple. Without the ripple, the angle dependence is well modeled by a cosine function

$$h_\theta = cos(k_\theta q_\theta). \tag{3.5}$$

The forcer used for these experiments had an angle range of about $\pm 1.8° = 0.031$ rad, suggesting that the force drops to zero at those limits and $k_\theta = \frac{\pi/2}{0.031}$. The angle range limit also coincides with the angle where the forcer teeth will overlap two rows of platen teeth, suggesting that it may be possible to compute k_θ directly from the tooth geometry as

$$k_\theta = \frac{\pi/2}{\sin(\rho/d_{wa})}, \tag{3.6}$$

where d_{wa} is the width of a forcer tooth (shown in Figure 1.8), and ρ is the tooth pitch of the system. For the Normag forcer tooth width of 31 mm, $k_\theta = \frac{\pi/2}{0.033}$, which is within a few percent of the value computed above based on the angle at zero force $k_\theta = \frac{\pi/2}{0.031}$.

In work contemporaneous with that presented here, Ish-Shalom [74] derived a model for the force output as a function of the skew angle by starting with the geometry of the overlap between forcer and platen teeth,

$$h'_\theta = \frac{\sin(\pi\sin(q_\theta)d_{wa}/\rho)}{\pi\sin(q_\theta)d_{wa}/\rho}. \tag{3.7}$$

For $d_{wa} >> \rho$, this model will closely match (3.5) over the operating range of the forcer, but it is more complicated and is not defined for $q_\theta = 0$. However, this work does provide additional evidence that the validity of (3.6) was not just a coincidence, suggesting that the angle dependence of other models of forcers may be well modeled given only d_{wa}/ρ, with no force measurements required.

3.2 Dynamic measurements

This section is concerned with the effect of motor velocity on force production. There is an *eddy current damping* force caused by the large time-varying magnetic fields generated as the motor passes over the unlaminated steel platen surface. As these forces act in opposition to the forcer velocity, they are often treated as a mechanical damping effect [36]. There are other damping forces such as tether drag and air resistance. These effects can all be treated as non-linear mechanical damping forces as they will oppose the direction of motion, effectively decreasing acceleration and increasing deceleration capabilities. Another class of effects, which will be called *rolloff* effects, always oppose the commanded acceleration, reducing both acceleration and deceleration capabilities. The main rolloff effect is the force reduction resulting from a limited commutation rate discussed in Chapter 2. In certain cases, amplifier dynamics may also cause an additional rolloff effect. This section presents an experimental technique to measure both the damping and rolloff effects as a function of velocity.

Using the proper value for the phase advance parameter, the bang-bang acceleration tests of Chapter 2 are repeated, again using the *normag0* system. This time, the interferometer position data are differenced twice to get an acceleration signal, and filtered to reduce the noise from force ripple and other sources. The motor acceleration as a function of velocity for the two constant acceleration segments of the motor trajectory can then be plotted. The upper traces in Figure 3.7 show ten repeated runs as well as an average for the positive acceleration segment of the trajectory. The lower traces show the same for the negative acceleration segment of the trajectory.

To compute the damping force, note that by definition, f_d will satisfy the relations

$$f_d(v, f_c) = f_d(v, -f_c), \text{ and} \tag{3.8}$$

$$f_d(0, f_c) = 0, \tag{3.9}$$

where f_c is the commanded force and v is the forcer velocity (in the actuator's direction). The rolloff force f_r will satisfy the relations

$$f_r(v, f_c) = -f_r(v, -f_c), \text{ and} \tag{3.10}$$

$$f_r(0, f_c) = 0. \tag{3.11}$$

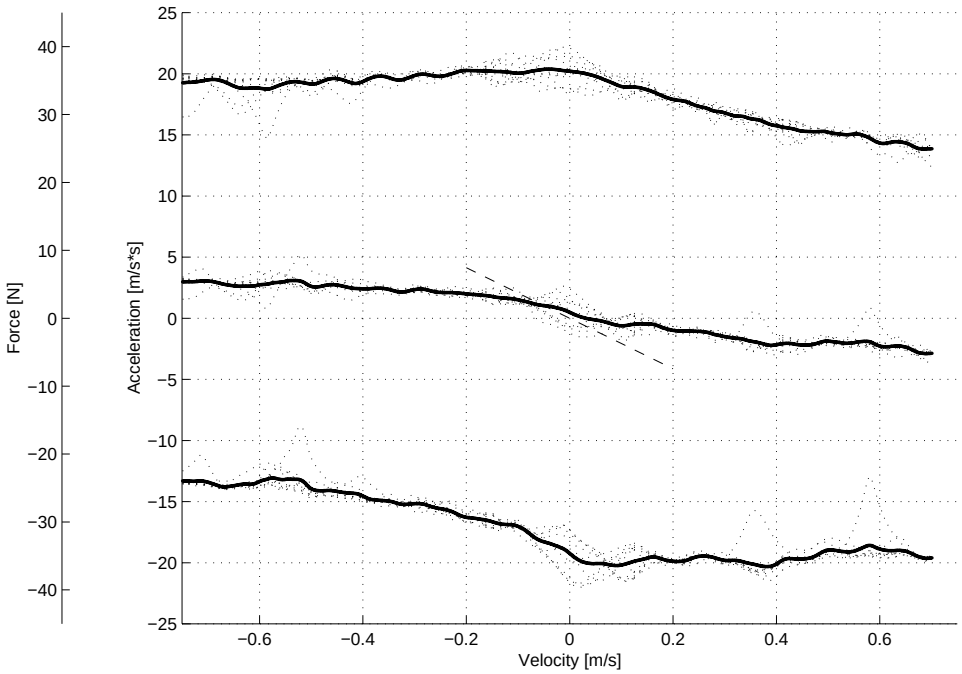


Figure 3.7: Acceleration and damping forces versus velocity. The upper and lower traces show acceleration vs. velocity. The middle trace shows the damping force. The dashed straight line is the linear damping force determined experimentally from an impulse response test. The vertical axes show both the acceleration and the equivalent force required to produce the acceleration.

Neglecting static force effects which will tend to be averaged out at high velocities, the measured force f_m will be given by

$$f_m(v, f_c) = f_c + f_d(v, f_c) + f_r(v, f_c). \tag{3.12}$$

If the force is measured twice at the same velocity but with opposite commanded forces, the damping force can be computed by averaging the two measured forces

$$\begin{aligned} \frac{1}{2}(f_m(v, f_c) + f_m(v, -f_c)) &= \frac{1}{2}(f_c + f_d(v, f_c) + f_r(v, f_c) \\ &\quad - f_c + f_d(v, f_c) - f_r(v, f_c)) \\ &= f_d(v, f_c). \end{aligned} \tag{3.13}$$

The rolloff force can then be computed by subtracting the damping force at velocity v from the measured force at that velocity and then further subtracting the measured force at zero velocity

$$\begin{aligned} f_m(v, f_c) - f_d(v, f_c) - f_m(0, -f_c) &= f_c + f_d(v, f_c) + f_r(v, f_c) \\ &\quad - f_d(v, f_c) - f_c \\ &= f_r(v, f_c). \end{aligned} \tag{3.14}$$

Referring again to Figure 3.7, the damping force as a function of velocity is computed by simply averaging the upper and lower curves. This computation gives the acceleration change, which can be multiplied by the motor mass, 1.8 kg, to get the damping force. This result is shown in the middle trace. In Figure 3.8 the rolloff forces are shown for both the top and bottom traces in Figure 3.7. Note the curves suggest a great degree of symmetry with respect to the signs of the forcer velocity and acceleration direction.

Although these data are somewhat noisy, they suggest that the damping forces are not linear, but level off after a velocity of about 0.5 m/s. Nordquist and Smith [31] use an alternate method for computing the damping forces. They assume the motor can be modeled over small displacements from an equilibrium position as a linear spring-mass-damper system. An impulse response test (tapping the forcer with a rubber mallet in the direction of interest) is used to determine the damping ratio of the system based on the motor mass and local stiffness. Applying this method to the Normag forcer gives a damping coefficient, k_v, of 37.2 N/m/s. The resulting linear damping force curve from this damping coefficient is shown plotted

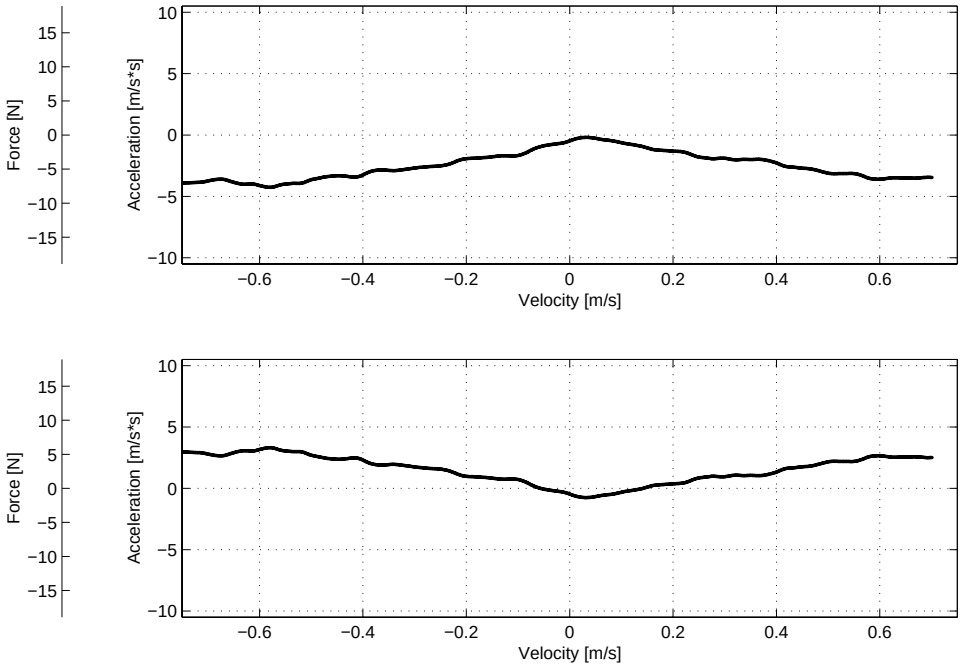


Figure 3.8: Acceleration rolloff versus velocity: the upper and lower plots show acceleration rolloff for the upper and lower traces of Figure 3.7. response test. The vertical axes show both the acceleration and the equivalent force required to produce the acceleration.

as a dashed line in Figure 3.7. It does not match the slope of our force computation except possibly near the zero velocity zone. However, this estimate supports the conclusion of a non-linear damping force, because if the dashed line is extended, it implies drastic drops in the available acceleration, which are not observed in the operation of the system.

Gjeltema [86], using a Normag motor, found the damping forces to be linear with velocity. However, only the acceleration characteristics of the motor are examined. Without deceleration data, damping forces (which *decrease* the maximum acceleration and *increase* the maximum deceleration) can not be distinguished from amplifier dynamics and commutation update rate effects (which decrease the maximum acceleration *and* deceleration).

3.2.1 Additional experiments

More recently, the experiments of this section were repeated using the *normag2* system. While previous experiments using the interferometer were limited to slew rates of 1 m/s, the platen sensor of this system allows for testing over a larger velocity range. The amplifiers have output filters and internal dynamic compensation for the load inductance that make the overall amplifier dynamics more complicated for this system. Bode plots of the amplifier dynamics are shown in Figure 3.9. They were measured with an HP 3562A dynamic signal analyzer, using the amplifier current reference as the input signal and the motor coil current, measured with a Tektronix A6303 current probe, as the output signal. During the test, the forcer was holding a nominal position in open-loop mode, but the position command was perturbed with 50 μm of white noise at 5 kHz. Note that the inductance of the motor coil does vary with position for this type of actuator, but tests at different positions produced only modest changes in the resulting Bode plots. The *normag2* system was intended for a usable velocity range of 1 m/s, so only the frequencies below 1 kHz are of interest. The phase plot is not well modeled by a line over this frequency range and the amplifier dynamics are not therefore well modeled by a delay, as with the Motion Science amplifiers of the *normag0* system described in the previous section, which have a set of dominant poles at 1400 Hz. Instead, the more complicated dynamics of the Copley amplifiers necessitate a function t_{amp} of the form

$$t_{amp} = k_1|v| + k_2, \tag{3.15}$$

to model the output delay in seconds caused by the amplifier assuming a forcer velocity of v m/s. Note that the velocity v is related to the phase plot frequency, f_{amp} expressed in Hz, by $v = f_{amp}\rho$, and the delay t_{amp} is related to the phase amplitude ϕ_{amp} by $t_{amp} = \phi_{amp}/(360 f_{amp})$. The parameters k_1 and k_2 can then be optimized so that the delay time t_{amp} is compatible with the data from the phase plot of Figure 3.9. For the *normag2* system, a least-squares fit produced parameter values $k_1 = 2.82 \times 10^{-5}$ s^2/m and $k_2 = 3.21 \times 10^{-6}$ s. The amplifier output delay time t_{amp} is then computed based on the estimated velocity in each direction and added to the other delays in the real-time system. For the *normag2* system, the commutator is updated at 5 kHz, but the code is structured so that there is a one cycle delay between sensor read and actuator command updates. In addition, there are additional software computation delays, as depicted in Figure 3.10. The total delay time t_d is given by

$$t_d = \frac{3}{2}T + t_{clock} + t_{comp} + t_{amp} - \frac{t_{int}}{2}, \tag{3.16}$$

where T is the sample time of the commutator (typically 200 μ s), t_{clock} is measured at every cycle by reading the microsecond resolution hardware clock (approximately 100 μs), t_{comp} is determined based on logged timing information and set to a fixed value (22 μs), and t_{int} is known based on the sensor filtering circuit design [22] (70 μs) The total delay time t_d is multiplied by the motor velocity to compute the amount of phase advance.

The bang-bang acceleration tests of this section were repeated for the *normag2* system. The acceleration traces and recovered damping forces are shown in Figure 3.11. These data indicate that the damping force is reasonably modeled as linear. The nonlinear damping exhibited in Figure 3.7 contradicts these results. What was observed in unrelated experiments was that the eddy-currents induced in the platen will depend on the exact current waveforms in the actuator coils during motions. Commutation at a different frequency or even having different PD gains may affect the damping forces recovered by this method. The damping, which had been assumed to be a physical property of the actuators, is actually a system property, dependent on the amplifiers, commutation scheme, and controller. In this case, adaptive control techniques may be most appropriate for determining the dynamic force characteristics of a particular system. Note that while the experimental results

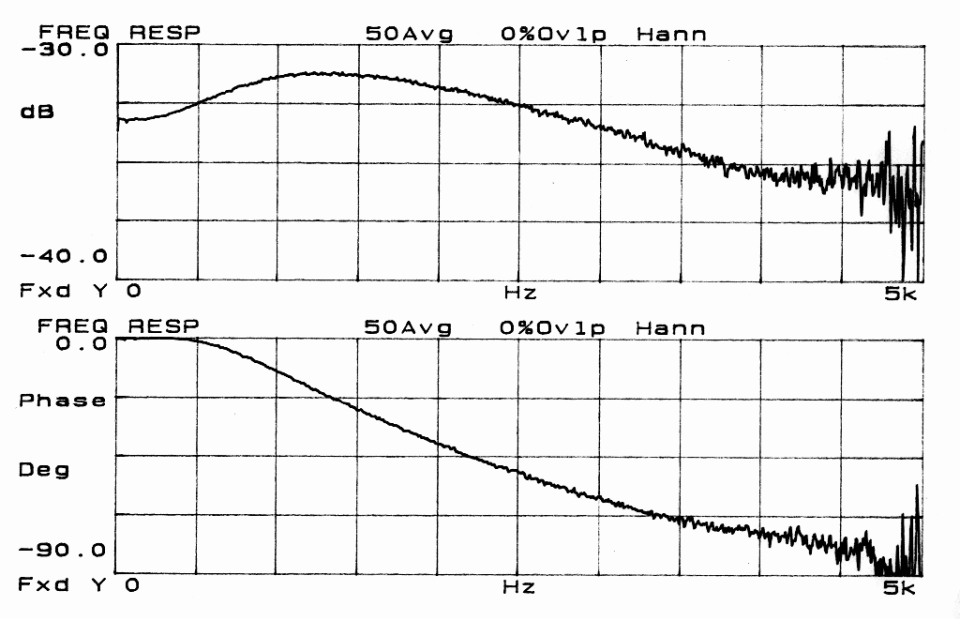


Figure 3.9: Amplifier dynamics for *normag2*

of this and the previous section relating to damping forces may not be directly applicable to other systems, the techniques presented for characterizing the dynamic force output of the actuators are applicable and represent an improvement over techniques described in the literature.

The recovered rolloff forces are shown in Figure 3.12, and show a significant force reduction. The measured rolloff, which should be symmetric with velocity is also shown mirrored about the zero velocity axis to give an indication of the accuracy of the measurement. Note that the relative vertical positions of the curves are chosen arbitrarily such that the peak value for each curve corresponds to zero rolloff. Part of this rolloff can be attributed to the finite commutation rates, discussed in Section 2.1. However, the size of this rolloff effect is larger than that predicted by the phase advance rolloff effect of (2.6). The additional rolloff may be caused by tether dynamics (recall that this test involves fast back and forth motions which tend to whip the tether around) or unmodeled amplifier dynamics (which will vary somewhat based on the motor velocity and current commands).

The experimental results in this chapter are an important step in improving the planar robot's understanding of its own operation. This understanding can allow the robot to more precisely react to events and to better know what the result of that reaction should be given a state of the environment. In this way the robot can

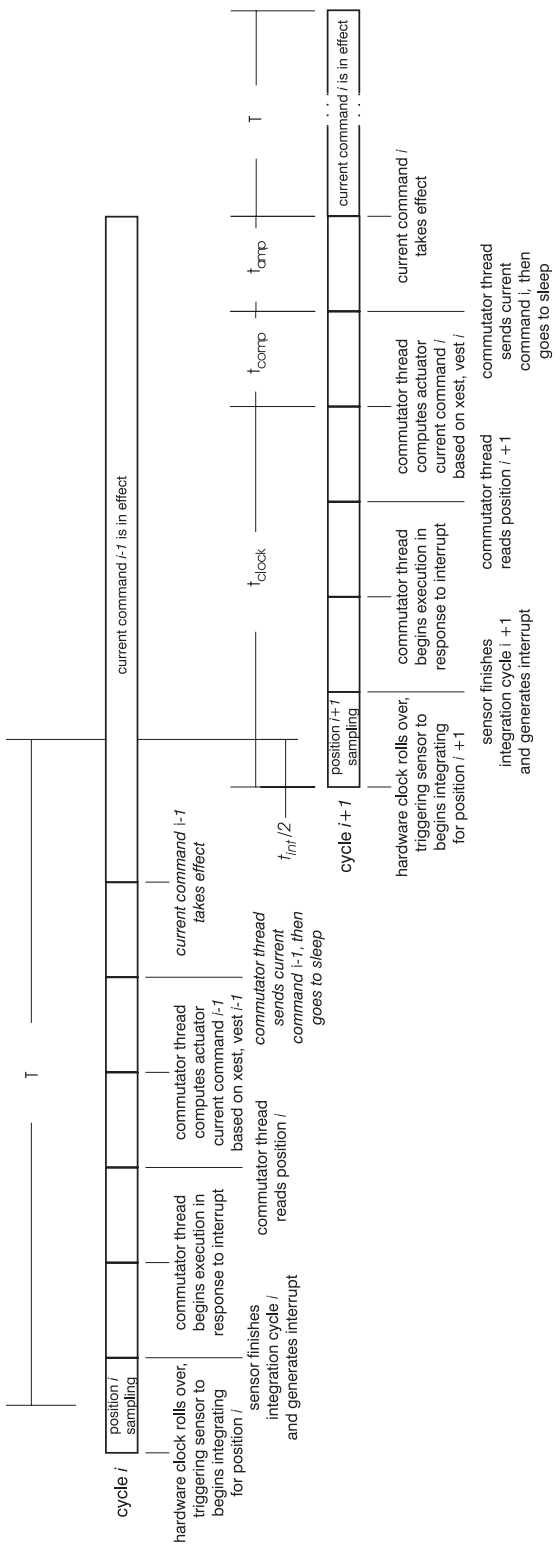


Figure 3.10: Input/output timing diagram for phase advance computation

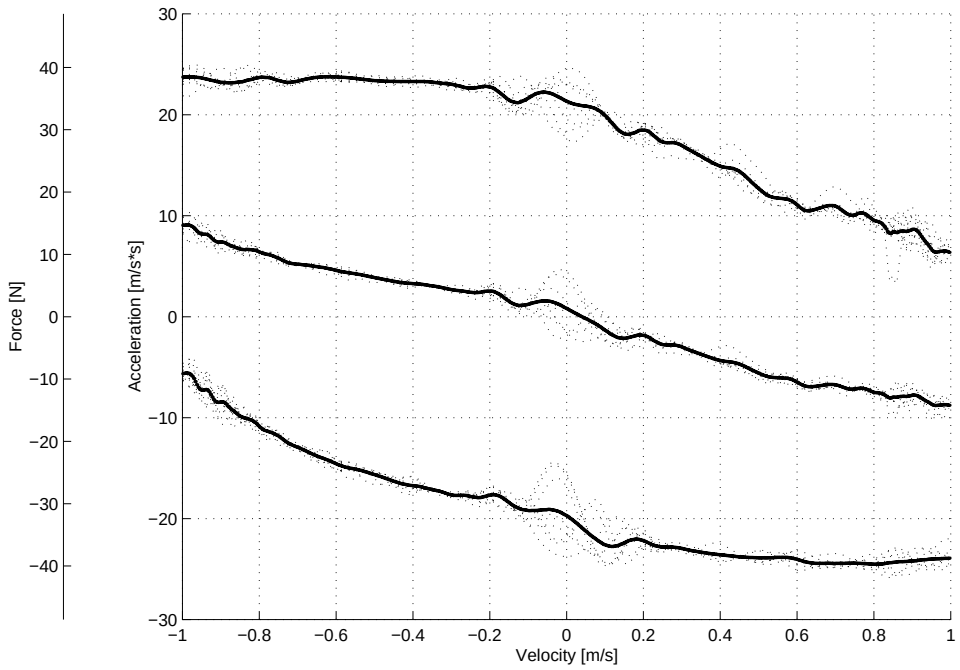


Figure 3.11: Acceleration and damping forces versus velocity for *normag2* system. The vertical axes show both the acceleration and the equivalent force required to produce the acceleration.

gain information about its environment not just through its sensors but also through its actuators. In manipulation tasks, sensing can often be expensive or difficult to integrate, whereas models are essentially free, once developed. The low-DOF nature of the planar robot allows a detailed examination of the actuator models because models of the link, drivetrain, kinematics, and dynamics are trivial or unnecessary.

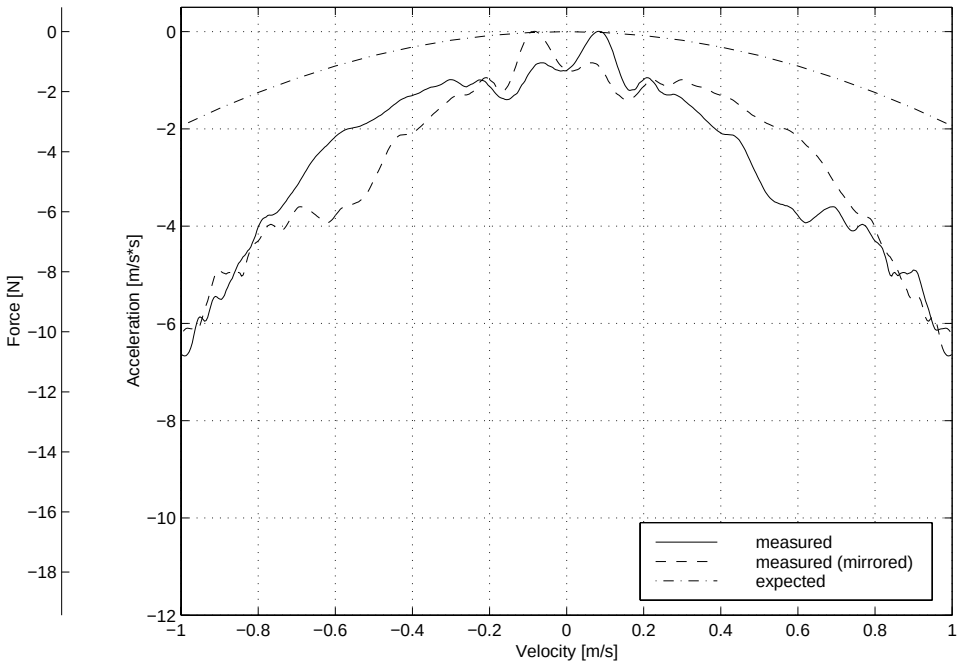


Figure 3.12: Acceleration rolloff versus velocity for *normag2* system. Expected rolloff is computed using (2.6). The vertical axes show both the acceleration and the equivalent force required to produce the acceleration.

Chapter 4

Autonomous calibration

The measurements of the previous chapter can be used to construct detailed models for the force output of the linear motor actuators. These models would then be used in the commutator or controller to allow for more accurate force output. However, the models will vary based on the particular platen and forcer used in a system. If different vendors supplied the platens and forcers or if they were purchased at different times, the end user would have to perform these measurements. While this measurement procedure was reasonable for research into the structure of the actuator force output, the need for an actuated load cell and the length of testing required makes it impractical for use outside a research lab.

Although not detailed in this document, an error model for the platen sensor output was similarly characterized, using a two-axis high-speed laser interferometer as a reference measurement [22]. This test required the careful alignment of a series of optics and it is unreasonable to expect every user to have such an expensive piece of metrology equipment. Once again, while it is possible to get an idea of the sensor error model in the laboratory, the exact measurements will only be valid for the one particular sensor and platen used for the experiments.

However, these measurements can be used to inform the development of parametric models for the planar motor actuators and sensors. The hope is that the *parameters* of the model would be able to encode manufacturing variations between forcers or platens, while the *structure* of the model would represent the common characteristics of their nominal design. The modeling problem is then reduced to a parameter identification problem.

Fortunately, it has been shown that external measuring devices are not always necessary to accomplish such parameter identification tasks. In one example, identification of the kinematic parameters of redundant manipulator systems was performed by recording the joint angles during self-motions with the end-effector fixed in the workspace [98]. This technique was also used for the calibration of parallel mechanisms with redundant sensing [112]. Similarly, two 6-DOF force sensors pressing against each other in different configurations have been shown to provide sufficient information to enable their calibration [99]. The lesson to be taken from these works is that redundant systems can permit autonomous calibration. In this chapter, these techniques are applied to the calibration of the sensors and actuators of the planar robot and their suitability as a technique for on-site calibration of the planar robot is demonstrated.

The planar robot has three DOFs (translation and rotation in the plane of the platen). The robot's platen sensor has four position sensing elements and the forcer has four linear actuators, yielding one redundancy in both sensing and actuation. As will be made more clear in the sections following, the sensor redundancy provides a constraint equation that can be used to form an objective function for calibration, while the actuator redundancy allows internal forces to be generated such that the full output range of the actuators can be calibrated without requiring an external force sensor.

The remainder of this chapter presents the application of autonomous calibration strategies to both the sensors and actuators of the planar robot. An important question this chapter seeks to answer is whether autonomous calibration methods are appropriate for the particular model structures of the planar robot sensors and actuators. Well-known mathematical techniques are used to investigate whether the model parameters can be identified by the particular calibration procedure. While these methods can detect a number of deficiencies, they are insensitive to certain model defects and violations of assumptions. These defects are best detected through the use of an independent external sensor, and thus comparisons against an independent sensor are included.

The following section presents autonomous calibration of the platen sensor using a linear constraint equation. In Section 4.2, the platen sensor error model is extended to include additional effects and nonlinear constraint equations are derived

to simultaneously calibrate the platen sensor parameters and the mounting parameters of the auxiliary *coordination sensor*. In Section 4.3, autonomous calibration of the actuator force outputs is presented. Finally, Section 4.4 presents details on the real-time implementation of the calibrated models.

4.1 Platen sensor autonomous calibration

4.1.1 Formulation

The platen sensor, mounted on the underside of the forcer, measures the location of forcer relative to the stationary platen teeth. Ideally, its quadrature decoding produces a perfect measure of the sensor's spatial phase. However, similar to the torque ripple of salient tooth actuators, the sensor will have spatial inaccuracies periodic in the tooth pitch. These inaccuracies will act as position-dependent disturbances that can lead to steady state errors and induce limit cycling in controllers with integral action. It is important to reduce these errors as much as possible. The sensor also relies on the locations of the platen teeth to determine motions beyond a single tooth pitch. There may be both local and global errors in the platen teeth locations. Individual teeth may be malformed or bent, and the tooth pitch may vary gradually over areas of the platen. Calibration of the hundreds of thousands of teeth on a typically sized platen is not practical, so individual defective teeth are not considered here. Global tooth deviations are also not considered, as the platen teeth will usually serve as the global position reference for the system.

This section describes a method for the autonomous calibration of the ripple errors of the platen sensor. Prior work [22] used a highly-precise laser interferometer to calibrate the platen sensor error. A 16th order polynomial was used as an error model of the sensor, and captured roughly half of the error of the sensor. The remaining error was from variations in the platen tooth geometry, which had been assumed to be uniform. A spectral analysis of the uncalibrated sensor error indicated that most of the sensor inaccuracies were at the first, second, fourth, and sixth harmonics of the tooth pitch. This observation suggested the use of a four frequency Fourier model,

$$e_i := k_{i,0} + \sum_{j=1\ldots4} k_{i,2j-1}\sin(\zeta_j\omega_\rho\check{p}_{s,i}) + k_{i,2j}\cos(\zeta_j\omega_\rho\check{p}_{s,i}), \tag{4.1}$$

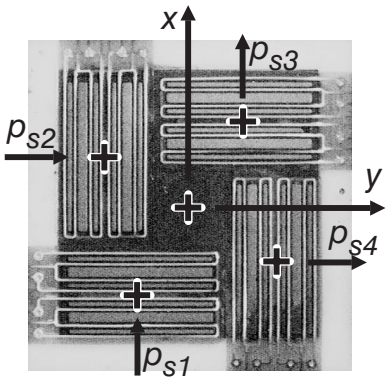


Figure 4.1: Platen sensor detail from *normag2* system

where $\check{p}_{s,i}$ is the uncalibrated output of the i^{th} sensor, $\omega_\rho := 2\pi/\rho$, ρ is the pitch of the sensor teeth (1.016 mm for all devices in this document), and the harmonics used are $\zeta = [1\ 2\ 4\ 6]^T$. This model assumes the error is small enough that $\check{p}_{s,i}$ is a reasonable substitute for the actual positions $p_{s,i}$ (i.e. $\cos(\zeta_j \omega_\rho \check{p}_s) \approx \cos(\zeta_j \omega_\rho p_s)$ and $\sin(\zeta_j \omega_\rho \check{p}_s) \approx \sin(\zeta_j \omega_\rho p_s)$ for $\zeta_j \in \{1, 2, 4, 6\}$). Effects of the forcer angle, q_θ, on the output (beyond changing the positions of the segments) are similarly neglected. As a result of these assumptions, the error, e_i, is linearly related to the parameter vector $k_i := [k_{i,0}, \ldots, k_{i,8}]^T$ and can be written as

$$e_i = \begin{bmatrix} 1 \\ \sin(1\omega_\rho \check{p}_{s,i}) \\ \cos(1\omega_\rho \check{p}_{s,i}) \\ \sin(2\omega_\rho \check{p}_{s,i}) \\ \cos(2\omega_\rho \check{p}_{s,i}) \\ \sin(4\omega_\rho \check{p}_{s,i}) \\ \cos(4\omega_\rho \check{p}_{s,i}) \\ \sin(6\omega_\rho \check{p}_{s,i}) \\ \cos(6\omega_\rho \check{p}_{s,i}) \end{bmatrix}^T \begin{bmatrix} k_{i,0} \\ k_{i,1} \\ \vdots \\ k_{i,8} \end{bmatrix} \tag{4.2}$$

$$=: \ ,_i^T k_i, \tag{4.3}$$

where $,_i$ is used to collect the sine and cosine terms. Note that all elements in k_i have units of distance and will have similar magnitudes.

Neglecting noise and unmodeled effects, the calibrated sensor output can be written as $\hat{p}_{s,i} := \check{p}_{s,i} + e_i$. The *sensor segment pair* in either cardinal direction can then be used to independently compute the angle of the forcer:

$$\hat{q}_{\theta,x} := \frac{\hat{p}_{s,1} - \hat{p}_{s,3}}{d_s} \quad \text{and} \quad \hat{q}_{\theta,y} := \frac{\hat{p}_{s,2} - \hat{p}_{s,4}}{d_s}, \tag{4.4}$$

where d_s is the distance between sensors, as shown in Figure 1.8. In normal operation, the angles in (4.4) are averaged to reduce measurement noise, but here *the redundancy is used to form a constraint equation*,

$$\begin{aligned} g_p &= (\hat{q}_{\theta,x} - \hat{q}_{\theta,y}) d_s \\ &= (\hat{p}_{s,1} - \hat{p}_{s,3}) - (\hat{p}_{s,2} - \hat{p}_{s,4}). \end{aligned} \tag{4.5}$$

Note that the $\hat{p}_{s,i}$ terms are functions of the model parameters k_i. By adjusting the values of the parameters, it is possible to reduce the size of g_p, which would be zero for a perfectly calibrated sensor with no measurement noise.

Given a set of M observations, $\check{p}^j_{s,i}$, indexed by $j \in \{1, \ldots, M\}$, we construct the overall cost $\sum_{j=1}^{M} (g_p^j)^2$ and find the set of parameters, k_i, which minimize it. Here, the constraint is linear in the parameters and a closed-form solution exists. Defining the constraint error for a single observation as

$$\check{g}_{p_j} := (\check{p}^j_{s,1} - \check{p}^j_{s,3}) - (\check{p}^j_{s,2} - \check{p}^j_{s,4}) \tag{4.6}$$

we can rewrite (4.5) as $g_p^j = \check{g}_p^j + (e_1^j - e_3^j) - (e_2^j - e_4^j)$. Making explicit the linear dependence of (4.1) on k_i yields

$$g_p^j = \check{g}_p^j + \sum_i (\, , \, {}^j_i)^T k_i = \check{g}_p^j + \, , \, {}^j k, \tag{4.7}$$

where $, \, {}^j$ collect the $, \, {}^j_i$ into a single matrix and k collect the k_i into a single vector, "stacking" them from each i sensor segment model. This form directly admits a linear least squares solution. Collecting the data over the index j leads to $G_p = \check{G}_p + \, , \, k$, with $G_p := [g_{p_1}, \ldots, g_{p_M}]^T$, $\check{G}_p := [\check{g}_{p_1}, \ldots, \check{g}_{p_M}]^T$, and $, \, := [, \, {}_1, \ldots, , \, {}_M]^T$. The optimal parameter estimate is then given by

$$\hat{k} = \, , \, {}^{\dagger} \check{G}_p, \tag{4.8}$$

where $, \, {}^{\dagger}$ is the Moore-Penrose pseudo-inverse of $, \, .$

The singular values of $, \,$ give an indication of the identifiability of the parameter vector k. Any zero singular values indicate that the parameter vector can be varied

along certain directions without having any effect on the cost function. Obviously, in this case there will not be a unique solution for (4.8) and the parameters cannot be uniquely determined. In practice, the singular values will be small but not zero, and the singular values must be compared against each other to see if any are near enough to zero to indicate an identifiability problem. The condition number (ratio of largest to smallest singular value) can be used to characterize the identifiability of the parameter vector. A condition number of less than 100 for the column-scaled matrix , is considered a requirement for reliable identification [113]. Column scaling of , prior to computing its condition number is required when the parameters have vastly different magnitudes. This scaling is not important for the sensor calibration of this section, as the model parameters in (4.1) all have units of distance and can be expected to have similar magnitudes.

The size of the residual vector $r :=$, $k - \check{G}_p$ indicates the combined effect of input noise, measurement noise, and unmodeled parameters. Computing the residual for nominal and calibrated values of k indicates how much improvement is gained by performing the calibration. Residuals can be computed using the calibration data set or a separate test data set to check that overfitting is not occurring.

Another verification technique [113, 114, 115] is to examine the covariance matrix, given by $(, ^T,)^{-1}$. The square-roots of the diagonal elements of this matrix give uncertainties, σ_i, of the parameter estimates resulting from (4.8). Scaling can be used to examine how the uncertainties in the parameter estimates affect the accuracy of the recovered sensor models. Scaled sensitivity values σ_i' for each parameter estimate $\hat{k}_i$ can be computed as $\sigma_i' = \sigma_i / \bar{\sigma}_{k_i}$ using the *mean scaling values* [113]

$$\bar{\sigma}_{k_i} = \frac{\epsilon}{\text{mean}(|, _i|)}, \tag{4.9}$$

where ϵ is a rough guess for the residual error level after the calibration (chosen here as 1 μm roughly corresponding to the noise level of the sensors). The sensitivity values σ_i' indicate how the uncertainty in the calibrated estimate of parameter k_i affects the residual error level after calibration. In cases where the condition number of , is large, parameters with large sensitivity values are candidates for elimination from the model [113, 114].

The most definitive, though often the most difficult, technique for calibration verification is to use an independent external sensor to measure the sensor errors

before and after calibration. The whole point of autonomous calibration is to eliminate such sensors, but during development of an autonomous calibration strategy, an external sensor is necessary to confirm the mathematical results of the calibration. In the following section, experiments using a laser interferometer are included to validate and characterize the efficacy of the autonomous calibration technique.

4.1.2 Experiments

Using an open-loop controller, the *normag1* forcer was moved to a series of uniformly distributed random q positions over a range of $0 < \{q_x, q_y\} < 10\rho$ and $-0.5° < q_\theta < 0.5°$. Data collection was complete within 6 minutes, as the high bandwidth and simultaneous sampling of the sensor allowed for fast motions with no delays for settling. The output of the data collection routine consisted of a vector of platen sensor outputs for each of 1000 forcer positions.

Using this dataset, , and $\check{G}_p$ were constructed from $\check{p}_s$ using the method described in Sec. 4.1.1. To confirm the identifiability of k, the condition number[1] of , was computed and found to be 7.7, suggesting that the parameters were sufficiently excited by the data, and that there are no structural redundancies between the parameters. This result may be somewhat surprising as there doesn't appear to be a position reference anywhere in the calibration equations, and yet there is not an arbitrary scale factor for the parameters. It is important to note that this calibration only seeks to compensate for the ripple nonlinearities of the sensor and not the overall scale factor. The parameters k of the model are in units of distance, with the assumption that the tooth pitch is 1.016 mm, so that in fact the tooth pitch is being used as the position reference for the sensor ripple errors.

The sensitivity values σ_i' were computed as described above to examine how the parameter estimates affect the accuracy of the fit. The sensitivities fell in the range 0.03 to 0.1, with the largest values corresponding to the parameters associated with the first harmonic terms ($k_{i,1}\sin(1\omega_\rho \check{p}_{s,i})$, and $k_{i,2}\cos(1\omega_\rho \check{p}_{s,i})$). These terms probably have the worst sensitivity because rotations of the forcer during collection are needed to cause the x or y sensor pairs to have different relative positions. The rotation range for the forcer of $\pm 0.5°$ during the calibration data collection produces

[1] As the arbitrary offset parameters $k_{i,0}$ cannot be individually identified, the condition number computation considered only their sum.

a relative motion between the sensors of only 0.17 mm, so it is not surprising that the first harmonic terms, which require relative motions closer to the pitch size of 1.016 mm to be fully excited, will be less well identified. However, it is also important to keep the skew angles small because the error model (4.1) was designed for small rotations and will be less accurate at higher rotation angles. Although further experiments could explore this tradeoff in more detail, the condition number and sensitivity results suggest that all the parameters are reasonably well-identified even with the relatively small rotation range chosen here.

The impact of the remaining unmodeled effects was computed by evaluating the residual error, $r := , k - \check{G}_p$. Using the identified parameters, the residual[2] was 4.1 μm, significantly smaller than the 29.3 μm residual using a zero vector for k, but still higher than the sub-micron resolution of the sensor.

To examine the generality of k, the robot was commanded to move a distance of 40 mm passing through the calibration area. For this *fly-by* test, the residual decreased from 12.1 μm to 5.5 μm with calibration, indicating that k was also valid over this dataset. To provide an independent verification and better characterize the performance of the autonomous calibration, retroreflectors were mounted on the forcer, allowing a laser interferometer to precisely measure two axes. The 40 mm fly-by test was repeated, and the translation along the direction of motion and rotation in the plane were sampled by both the interferometer and platen sensor. Using the interferometer measurements as ground-truth, the sensor translation errors are shown in Figure 4.2 and the errors for each of the sensor angle measurements are shown in Figure 4.3. The position error decreased from 8.9 μm to 6.5 μm with calibration. The error of $q_{\theta,x}$ stayed about the same at 0.4 mrad, while the error of $q_{\theta,y}$ decreased from 1.2 mrad to 0.4 mrad with calibration.

Although the autonomous calibration does yield a nominal improvement in position error and an improvement in the $q_{\theta,y}$ angle error by a factor of 3, there is a significant systematic error in $q_{\theta,x}$ that does not decrease. For this test, the robot moved along the y direction, nominally leaving fixed the outputs of the sensors that measure motion in the x direction and are used to compute $q_{\theta,x}$. Errors in $q_{\theta,x}$ are therefore best explained as an unmodeled dependence of the output on lateral position. The lateral direction here is defined as the direction perpendicular to the

[2]Standard deviations are used to characterize the size of residual and error vectors in this chapter, unless otherwise noted.

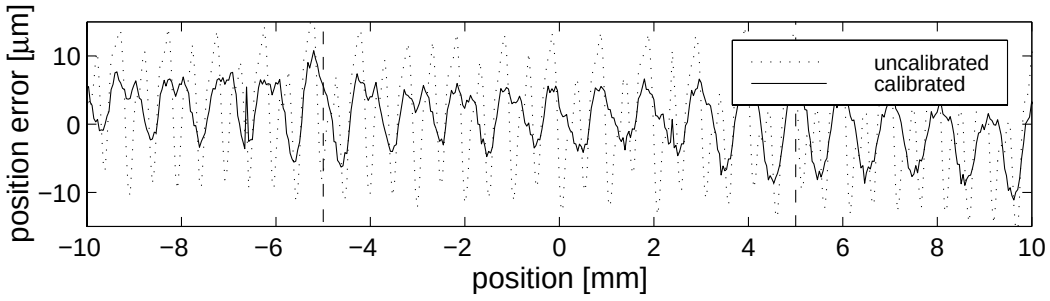


Figure 4.2: Position error in direction of motion during motion through platen sensor calibration region (indicated by dashed lines).

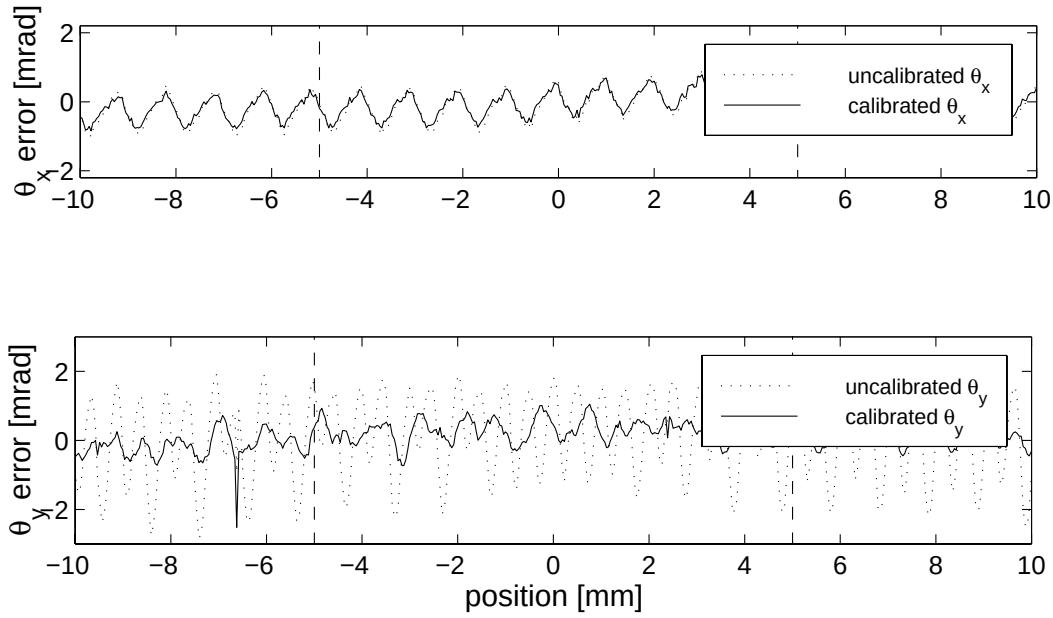


Figure 4.3: Angle error during motion through platen sensor calibration region.

direction that a sensor segment is designed to sense. As these *lateral dependencies* are roughly the same size as the residual errors and are unmodeled by (4.1), they are likely to be limiting further error reduction. However, the structure of the constraint equation (4.5) makes it difficult to distinguish lateral dependencies of one sensor segment pair from translational effects of the perpendicular sensor segment pair.

4.2 Dual sensor autonomous calibration

4.2.1 Formulation

In hopes of identifying the lateral effects that appear to cause trouble with the autonomous platen sensor calibration, the auxiliary coordination sensor can be used to provide additional constraints. This sensor, as introduced in Section 1.4.2, is an upward-facing lensed lateral effect photo-diode. Its primary purpose is for micron-precision localization of LED beacons attached to overhead devices. If one of the beacons has a fixed position, it can also be used to provide additional sensor calibration information. However, the precise mounting position and other parameters of the coordination sensor are not known. In this section, the autonomous calibration method is applied to the calibration of both the coordination sensor and an extended model of the platen sensor.

To model the lateral motion effects, the platen sensor models (4.1) are augmented with additional parameters designed to model the first harmonic of the lateral motion of the sensor

$$\begin{aligned} e'_1 &= e_1 + k_{1,9}\sin(\omega_\rho(\check{q}_y - d_{so}\check{q}_\theta)) + k_{1,10}\cos(\omega_\rho(\check{q}_y - d_{so}\check{q}_\theta)) \\ e'_2 &= e_2 + k_{2,9}\sin(\omega_\rho(\check{q}_x + d_{so}\check{q}_\theta)) + k_{2,10}\cos(\omega_\rho(\check{q}_x + d_{so}\check{q}_\theta)) \\ e'_3 &= e_3 + k_{3,9}\sin(\omega_\rho(\check{q}_y + d_{so}\check{q}_\theta)) + k_{3,10}\cos(\omega_\rho(\check{q}_y + d_{so}\check{q}_\theta)) \\ e'_4 &= e_4 + k_{4,9}\sin(\omega_\rho(\check{q}_x - d_{so}\check{q}_\theta)) + k_{4,10}\cos(\omega_\rho(\check{q}_x - d_{so}\check{q}_\theta)), \end{aligned} \tag{4.10}$$

where $\check{q}_x = (\check{p}_{a,1} + \check{p}_{a,3})/2$ and $\check{q}_y = (\check{p}_{a,2} + \check{p}_{a,4})/2$ are the average of the uncalibrated sensor readings along each axis, and $\check{q}_\theta$ is the average of the uncalibrated angle measurements computed as in (4.4). The error of each sensor segment is now dependent on the measurements of the other segments.

Although the coordination sensor requires internal calibration to compensate for electronic gains, mounting inaccuracies, optical distortions, and PSD nonlinearities, we assume for now that these effects have been calibrated[3] so that it outputs a perfect measurement of the angles to the LED beacon (ψ_x and ψ_y). Assuming a stationary beacon, the motion of the sensor is given by

$$c_x = h\tan(\psi_x) \quad \text{and} \quad c_y = h\tan(\psi_y). \tag{4.11}$$

[3]In practice these parameters would be identified prior to integration with the forcer.

As the coordination sensor supplies two additional measurements, two more constraint equations can be derived. A physically meaningful constraint is to equate the displacements of coordination sensor (c_x, c_y) with that of the platen sensor (q_x, q_y)

$$\begin{bmatrix} g_x \\ g_y \end{bmatrix} = -\begin{bmatrix} \hat{q}_x \\ \hat{q}_y \end{bmatrix} + \begin{bmatrix} \cos(\hat{q}_\theta) & -\sin(\hat{q}_\theta) \\ \sin(\hat{q}_\theta) & \cos(\hat{q}_\theta) \end{bmatrix} \begin{bmatrix} d_{cx} + c_x \\ d_{cy} + c_y \end{bmatrix} + \begin{bmatrix} o_x \\ o_y \end{bmatrix}, \tag{4.12}$$

where c_x, c_y are relative to coordinate frame C shown in Figure 1.16, which has a stationary origin under the LED beacon but rotates about the c_z axis to match any rotations q_θ of the forcer. The platen sensor positions $\hat{q}_{x,y}$ are relative to stationary coordinate frame P, and are simply the average of the two platen sensor measurements in each direction. Parameters d_{cx} and d_{cy} are the physical offsets between the centers of the two sensors as measured on the body of the forcer, while offsets o_x and o_y account for the arbitrary zero position of the platen sensor relative to the beacon. The angle of the forcer, $\hat{q}_\theta$, is the average of $\hat{q}_{\theta,x}$ and $\hat{q}_{\theta,y}$ in (4.4).

These constraint equations contain five new parameters to be identified (h, d_{cx}, d_{cy}, o_x, o_y), in addition to the platen sensor parameter vectors (k_1, k_2, k_3, k_4). The three constraint equations g_x and g_y (above) and g_p, (4.5), serve as error vectors that must be combined for minimization. Note that they all have units of distance, but the noise levels of these constraints are significantly different. To see this difference, first assume that the platen sensor position measurements and coordination sensor position measurements have normal distributions with standard deviation σ_s. Then the angle measurements, computed using $q_\theta = (p_{s1} - p_{s3} + p_{s2} - p_{s4})/(2d_s)$ will have standard deviation of $\sigma_\theta = \sqrt{4(\sqrt{2}\sigma_s/2)^2}/d_s = \sqrt{2}\sigma_s/d_s$. The sensor noise levels can be similarly mapped thru the constraint equations to estimate their standard deviations

$$\begin{array}{rcccl} \sigma_{g_x} & \approx & \sqrt{2\sigma_s^2 + (\frac{\sqrt{2}}{d_s} d_{cy}\sigma_s)^2} & \approx & 1.4\sigma_s \\ \sigma_{g_y} & \approx & \sqrt{2\sigma_s^2 + (\frac{\sqrt{2}}{d_s} d_{cx}\sigma_s)^2} & \approx & 14.2\sigma_s \\ \sigma_{g_p} & \approx & \sqrt{2\sigma_s^2} & \approx & 1.4\sigma_s, \end{array} \tag{4.13}$$

where $\sigma_{g_{x,y,p}}$ are the standard deviations of the respective constraint equations, d_{cx} is assumed to be near zero, and d_s/d_{cy} is assumed to be about 10. Therefore, constraint equation g_y will be inherently about 10 times noisier than the other constraint equations, suggesting that the overall error vector g should include scaling of the constraints, chosen here as $g := [g_x^T \;\; g_y^T/10 \;\; g_p^T]^T$.

Note that g is a non-linear function of the parameters and an iterative procedure must be used to find the best parameter estimate. The Levenberg-Marquardt algorithm [115], widely used for non-linear least-squares minimization problems, is used here. It combines gradient-descent and inverse-Hessian[4] approaches. At each step, the Jacobian, $J := \partial g/\partial k$, is evaluated and used to compute the next update to the parameter vector. Column-scaling of J is important here, as the model parameters have varying magnitudes. Scaling is performed based on the *extreme scaling values* [113] given by

$$\sigma_{k_i} = \frac{\epsilon}{\max_j \left(\sqrt{\left(\frac{\partial g_x^j}{\partial k_i}\right)^2 + \left(\frac{\partial g_y^j}{10 \partial k_i}\right)^2 + \left(\frac{\partial g_p^j}{\partial k_i}\right)^2} \right)}, \tag{4.14}$$

where ϵ is a guess which roughly indicates the expected error level of the calibration, chosen here at 1 μm. The scaled Jacobian, J', is then given by $J' = JH$, with $H = \mathrm{diag}(\sigma_{k_1}, \ldots, \sigma_{k_M})$. The covariance matrix in this case is computed using the unscaled Jacobian matrix as $(J^T J)^{-1}$. The roots of its diagonal elements σ_i are scaled based on the mean scaling values [113]

$$\bar{\sigma}_{k_i} = \frac{\epsilon}{\mathrm{mean}_j \left(\sqrt{\left(\frac{\partial g_x^j}{\partial k_i}\right)^2 + \left(\frac{\partial g_y^j}{10 \partial k_i}\right)^2 + \left(\frac{\partial g_p^j}{\partial k_i}\right)^2} \right)}, \tag{4.15}$$

to get the scaled sensitivity values $\sigma_i' = \sigma_i / \bar{\sigma}_{k_i}$.

4.2.2 Experiments

The *normag1* robot was positioned so that the coordination sensor was directly under an LED beacon and was moved to a series of uniformly distributed random q positions over a range of $0 < \{q_x, q_y\} < 2\rho$ and $-1.5° < q_\theta < 1.5°$. In this case, the forcer was designed to rotate about the approximate center of the coordination sensor in order to minimize the effects of coordination sensor nonlinearities. The output of this data collection process consisted of a vector of platen sensor outputs, $\check{q}$, and coordination sensor outputs, (ψ_x, ψ_y), for each of 1000 forcer positions.

To provide a good initial parameter vector, the platen sensor autonomous calibration results were used to initialize the basic set of platen sensor parameters, while

[4] The Hessian is the matrix of second derivatives of the cost function with respect to the model parameters.

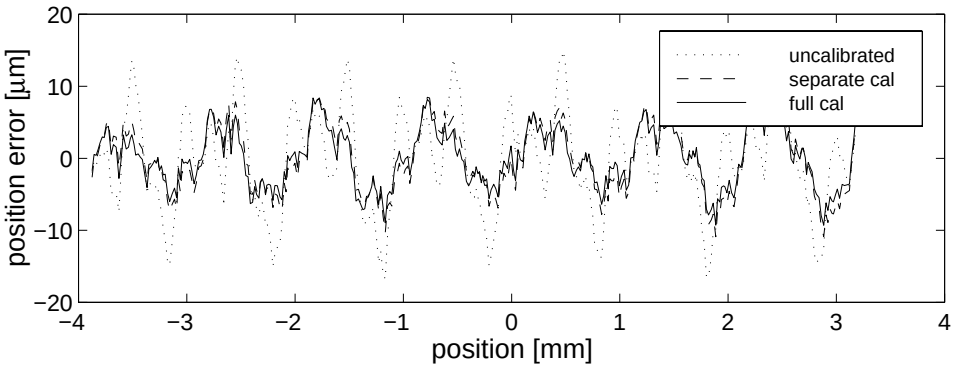


Figure 4.4: Position error in direction of motion during motion through dual sensor calibration region.

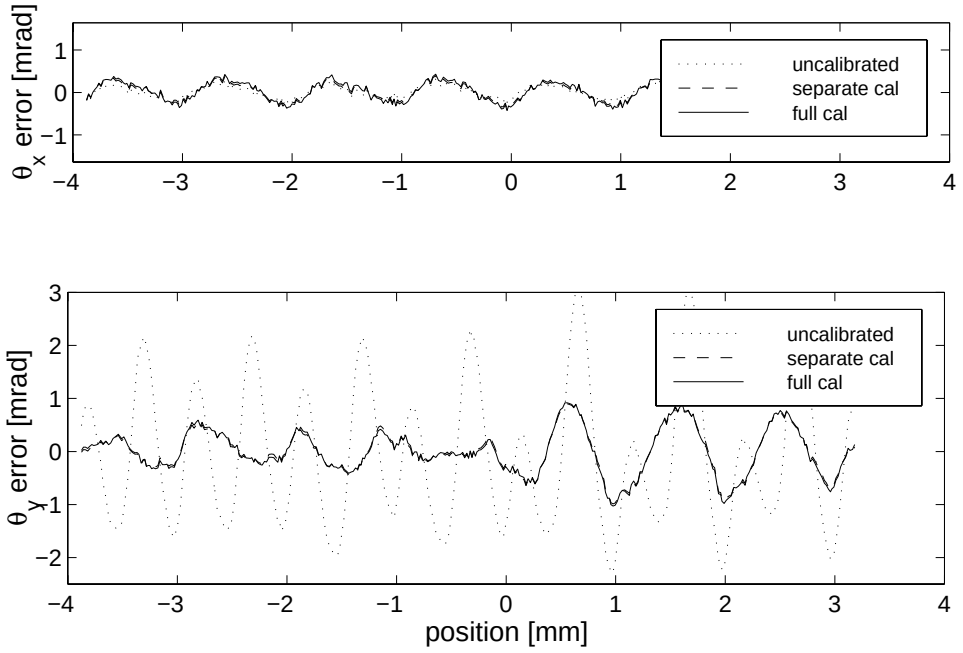


Figure 4.5: Angle error during motion through dual sensor calibration region.

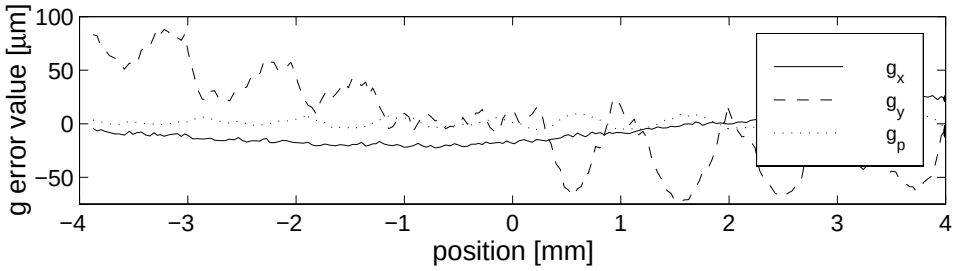


Figure 4.6: Constraint equation errors during motion through dual sensor calibration region.

the coordination sensor parameters (o_x, o_y, l_x, l_y, h) were initialized by minimizing a partial error vector $g_{xy} = [g_x^T \; g_y^T/10]^T$ with the platen parameters fixed. With these initial parameter estimates, the complete error vector g was minimized over the full set of parameters, including the lateral effect terms ($k_{i,9}, k_{i,10}$). The Levenberg-Marquardt routine converged within 10 iterations. The condition number of J' was 3.6 at the final parameter estimate. For comparison, the unscaled Jacobian had a condition number of over 2000, demonstrating the benefits of column scaling.

The residuals of the three constraints (g_x, g_y, g_p) were computed at the initial (separate calibration of the coordination sensor and platen sensor parameters) and final (combined calibration of all parameters) parameter vectors, with values (4.8, 34, 4.2) μm and (4.6, 29, 4.2) μm, respectively. The small difference suggests that the addition of the lateral effect parameters for the platen sensor did not have a large effect. To test the calibration over a different dataset, residuals for the three constraints were measured using the calibrated sensor outputs for an 8 mm flyby through the calibration area, shown in Figure 4.6. The errors in g_y and, to a lesser extent, g_p vary systematically with tooth pitch, indicating unmodeled platen sensor errors. However, the g_x and g_y errors vary gradually over multiple pitches, suggesting unmodeled coordination sensor effects.

The scaled sensitivity values σ_i' were computed and all had values between 0.025 and 0.05. Although the larger values corresponded to the first harmonic terms ($k_{i,1}\sin(1\omega_\rho \check{p}_{s,i})$, $k_{i,2}\cos(1\omega_\rho \check{p}_{s,i})$, $k_{i,9}\sin(\omega_\rho(\check{q}_y - d_{so}\check{q}_\theta))$, and $k_{1,10}\cos(\omega_\rho(\check{q}_y - d_{so}\check{q}_\theta))$) from (4.1) and (4.10), there was not a pronounced difference between the sensitivities of any of the parameters.

The laser interferometer was again sampled to provide an independent verification for a fly-by of 8 mm, while simultaneously sampling the platen sensor and coordination sensor outputs. The platen sensor position error (using the interferometer as ground truth) decreased from 7.3 μm uncalibrated to 4.4 μm using the initial parameters from the separate calibrations to 3.6 μm for the final calibration. The error of $q_{\theta,y}$ (using the moving sensor segment pair) decreased from 1.4 mrad uncalibrated to 0.40 mrad for the separate calibration, and stayed at 0.40 mrad for the combined calibration. The error of $q_{\theta,x}$ (using the nominally stationary sensor segment pair), which was expected to decrease when adding the lateral platen sensor parameters, remained small but increased from 0.13 mrad uncalibrated to 0.16 mrad for the separate calibration to 0.19 mrad for the combined calibration. In addition, the identified values of the $k_{i,9}, k_{i,10}$ parameters were all under 2 μm, which were smaller than the lateral effects seen in previous data, strongly suggesting that these less significant parameters are not being properly identified.

The calibration appears to be limited by unmodeled nonlinearities in the coordination sensor. Although the experiments were designed to keep the coordination sensor angles small, it was necessary to move several pitches for identifiability reasons, and even this small motion seems to have produced errors of several microns, as shown in the g_x error in Figure 4.6. Although promising for cases where the coordination sensor is available, this combined calibration technique was not used to avoid the need for calibration of the internal parameters (lens distortion, optical axis, etc.) of the coordination sensor.

Instead, the platen sensor is calibrated using the technique of Section 4.1. These experiments demonstrated an improvement in accuracy to a level of 6 μm in translation and 0.4 mrad in rotation, a level of accuracy comparable to results of an interferometer-based calibration [22]. However, these values are significantly higher than the 0.2 μm sensor resolutions. Two effects are limiting further accuracy improvements. First, the actuator and platen teeth are assumed to be uniform, but the prototype systems actually have significant manufacturing deviations, as shown in Figure 1.12. To determine a lower bound on the accuracies that can be expected from the calibration, the forcer was moved slowly over a distance of seven teeth while its position was recorded by both the platen sensor and interferometer. The errors between the two position signals were used to create a lookup table that mapped the raw sensor phase angle to an error correction value, averaged over the seven

teeth. If the teeth were identical, applying this error correction to the original sensor measurements should have produced the interferometer measurements. However, there were variations of 1.9 μm, suggesting that even over a small area of the platen, tooth variation is significant enough to produce accuracy errors an order of magnitude larger than the sensor noise level. Further accuracy improvements require a global model of the platen or more uniform platens, such as the prototype shown in Figure 1.13.

However, the platen sensor autonomous calibration of Section 4.1.2 appears to give sufficient accuracy. Since the time that the experiments of this chapter were completed, the *normag2* planar motor was outfitted with an integrated platen sensor. Before calibration, the sensor accuracy was poor enough that closed-loop control was too fragile to be usable. After calibration, it performed qualitatively as well as the *normag1* system. Although the calibration of the second system was not verified with the interferometer, the improved behavior does provide at least anecdotal evidence that the autonomous calibration technique is sufficient for calibration of the platen sensor and that the model for the sensor is appropriate in that the parameter values encapsulate most of the variation between two instantiations of the CMU sensor.

4.3 Actuator autonomous calibration

The salient-tooth stepping motor design of the actuators makes it impossible to derive analytic models for the resulting force output. Using simple first-order models results in significant force ripple (variation with position) and in certain cases, rolloff at higher force commands due to magnetic saturation. The autonomous calibration presented below seeks to model these effects. The main reasons for going to this trouble are to decrease the size of disturbances that must be rejected by the controller and to improve open-loop (i.e. without extrinsic force sensing) force generation for force control applications. It should be noted the calibration technique assumes a calibrated 3-DOF platen sensor. A simple external wrench is assumed to be acting on the forcer (the zero wrench or a constant wrench to account for a constant tether disturbance) and Newton's First Law provides the constraint equations. Note that the actuator redundancy does not provide a constraint equation, but rather allows the actuators to push against each other so that they can have large force outputs without having any net wrench on the forcer body. This feature of redundancy

allows the calibration of the full output range of the actuators, rather than just the zero-current detent forces.

4.3.1 Formulation

In the measurements presented in Section 3.1, a load-cell was used to measure the force outputs of the *normag0* system. Force ripple, saturation, detent forces, and angle dependencies were apparent from the data. A reasonable model for a single actuator that includes these effects is given by

$$\begin{aligned} f_i \;:=\; \cos(k_\theta q_\theta)\{\; & (k_{i,1} I_i + k_{i,2} I_i^3)(\, 1 + k_{i,3}\sin(\omega_\rho p_{a_i}) + k_{i,4}\cos(\omega_\rho p_{a_i}) \\ & \qquad + k_{i,5}\sin(2\omega_\rho p_{a_i}) + k_{i,6}\cos(2\omega_\rho p_{a_i}) \\ & \qquad + k_{i,7}\sin(4\omega_\rho p_{a_i}) + k_{i,8}\cos(4\omega_\rho p_{a_i})\,) \\ & + k_{i,9}\sin(\omega_\rho p_{a_i}) + k_{i,10}\cos(\omega_\rho p_{a_i}) \\ & + k_{i,11}\sin(2\omega_\rho p_{a_i}) + k_{i,12}\cos(2\omega_\rho p_{a_i}) \\ & + k_{i,13}\sin(4\omega_\rho p_{a_i}) + k_{i,14}\cos(4\omega_\rho p_{a_i})\;\} \end{aligned} \tag{4.16}$$

where f_i is the output of the i^{th} actuator, $\omega_\rho := 2\pi/\rho$, ρ is the pitch of the actuator teeth (1.016 mm for all devices in this document), and k_θ is computed *a priori* using (3.6) with $d_{wa} = 31.0$ mm. The first term, $\cos(k_\theta q_\theta)$, represents the force dependence on skew angle from (3.5). The next few terms, $k_{i,1} I_i + k_{i,2} I_i^3$, model the force as a function of current, as was shown in Figure 3.5, and match the form of (3.3). The next set of terms, $(1 + k_{i,3}\sin(\omega_\rho p_{a_i}) + k_{i,4}\cos(\omega_\rho p_{a_i}) k_{i,5}\sin(2\omega_\rho p_{a_i}) + k_{i,6}\cos(2\omega_\rho p_{a_i}) k_{i,7}\sin(4\omega_\rho p_{a_i}) + k_{i,8}\cos(4\omega_\rho p_{a_i}))$, model the force ripple using Fourier-type terms with the harmonics observed in Figure 3.4. The remaining six terms are used to model the zero-current force ripple, using the same Fourier-type harmonic terms. The model assumes that the positions of the actuators, p_{a_i}, are known, which requires a calibrated platen sensor so that the locations of the actuators can be computed. Note that this model is bilinear in the parameters, $k_{i,1} \ldots k_{i,14}$.

However, if the current range is limited such that the linear term $k_{i,1}I_i$ dominates[5] the non-linear term $k_{i,2}I_i^3$, then the following linear model can instead be used

$$\begin{aligned} f_i \;:=\; & \cos(k_\theta q_\theta)\{I_i(k_{i,1} + k_{i,2}\sin(\omega_\rho p_{a_i}) + k_{i,3}\cos(\omega_\rho p_{a_i}) \\ & \qquad + k_{i,4}\sin(2\omega_\rho p_{a_i}) + k_{i,5}\cos(2\omega_\rho p_{a_i}) \\ & \qquad + k_{i,6}\sin(4\omega_\rho p_{a_i}) + k_{i,7}\cos(4\omega_\rho p_{a_i})) \\ & + k_{i,8}\sin(\omega_\rho p_{a_i}) + k_{i,9}\cos(\omega_\rho p_{a_i}) \\ & + k_{i,10}\sin(2\omega_\rho p_{a_i}) + k_{i,11}\cos(2\omega_\rho p_{a_i}) \\ & + k_{i,12}\sin(4\omega_\rho p_{a_i}) + k_{i,13}\cos(4\omega_\rho p_{a_i})\}. \end{aligned} \tag{4.17}$$

It can be rewritten in linear form

$$\begin{aligned} f_i \;=\; & \begin{bmatrix} I_i\cos(k_\theta q_\theta) \\ \sin(\omega_\rho p_{a_i})I_i\cos(k_\theta q_\theta) \\ \cos(\omega_\rho p_{a_i})I_i\cos(k_\theta q_\theta) \\ \sin(2\omega_\rho p_{a_i})I_i\cos(k_\theta q_\theta) \\ \cos(2\omega_\rho p_{a_i})I_i\cos(k_\theta q_\theta) \\ \sin(4\omega_\rho p_{a_i})I_i\cos(k_\theta q_\theta) \\ \cos(4\omega_\rho p_{a_i})I_i\cos(k_\theta q_\theta) \\ \sin(\omega_\rho p_{a_i})\cos(k_\theta q_\theta) \\ \cos(\omega_\rho p_{a_i})\cos(k_\theta q_\theta) \\ \sin(2\omega_\rho p_{a_i})\cos(k_\theta q_\theta) \\ \cos(2\omega_\rho p_{a_i})\cos(k_\theta q_\theta) \\ \sin(4\omega_\rho p_{a_i})\cos(k_\theta q_\theta) \\ \cos(4\omega_\rho p_{a_i})\cos(k_\theta q_\theta) \end{bmatrix}^T \begin{bmatrix} k_{i,1} \\ k_{i,2} \\ \vdots \\ k_{i,12} \\ k_{i,13} \end{bmatrix} \\ =:\; & \gamma_i k_i \end{aligned} \tag{4.18}$$

Assuming there is a constant, but unknown, external wrench $w_e = [f_{e,x}\ f_{e,y}\ \tau_e]^T$ acting on the robot[6], there will be a constraint equation for each axis

$$\begin{aligned} g_{fx} &:= 0 = f_1 + f_2 + f_{e,x} \\ g_{fy} &:= 0 = f_3 + f_4 + f_{e,y} \\ g_\tau &:= 0 = f_1 - f_2 + f_3 - f_4 + \tfrac{2\tau_e}{d_a}, \end{aligned} \tag{4.19}$$

[5] Note that $k_{i,1}$ and $k_{i,2}$ will be in units of N/A and N/A^3, respectively and will have dramatically different values.

[6] This wrench is to account for a bias from the tether disturbance, which is assumed to not change much over the small robot motions during data collection.

where d_a is the distance between actuators, as shown in Figure 1.8, and the output force directions are assigned as shown in Figure 4.7. For each observation j, these constraint equations can be combined to give

$$, {}^j k \;:=\; \begin{bmatrix} , {}^j_{fx} \\ , {}^j_{fy} \\ , {}^j_{\tau} \end{bmatrix} k \;:=\; \begin{bmatrix} \gamma_1^j & \gamma_2^j & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \gamma_3^j & \gamma_4^j & 0 & 1 & 0 \\ \gamma_1^j & -\gamma_2^j & \gamma_3^j & -\gamma_4^j & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ k_3 \\ k_4 \\ f_{e,x} \\ f_{e,y} \\ \frac{2\tau_e}{d_a} \end{bmatrix} \;=\; 0. \tag{4.20}$$

In this case, the noise of each constraint in (4.19) from the actuator force outputs will not have large enough differences in magnitudes to necessitate scaling of the constraints. In addition, the amount of variation in the external wrench, $[f_{e,x},\ f_{e,y},\ \tau_e]^T$, is not known and likely dominates the other noise sources.

Given a set of M observations, $(p_{a_i}^j, I_i^j)$, indexed by $j \in \{1, \ldots, M\}$, we construct the overall cost $\sum_{j=1}^{M} (, {}^j k)^2$ and find the parameter estimate, k, which best minimizes it. Here, the constraint is linear in the parameters and a closed-form solution exists. The parameter vector k can be found by collecting the data over index j and solving

$$0 \;=\; \begin{bmatrix} , {}_{fx} \\ , {}_{fy} \\ , {}_{\tau} \end{bmatrix} k \;=:\; , k \ , \tag{4.21}$$

where $, {}_{fx}$, $, {}_{fy}$, and $, {}_{\tau}$ are $M \times 55$ (13 parameters for four actuators plus three external wrench offset parameters gives 55 total parameters), and $,$ is $3M \times 55$. Note however that $k = 0$ is a solution for (4.21), indicating that the overall scaling of the parameter vector cannot be recovered. In fact, as there is no force reference to be found in the experimental setup, it is not surprising that it is impossible to recover this scale information. A simple solution is to arbitrarily fix the first parameter to a reasonable value. Estimating this scale factor to within 5–10% is sufficient for closed-loop control purposes, where its main effect is in altering the effective gain of the controller output. In this case, simply choosing k_1 such that the resulting force models match the nominal force model at the peak rated current would be sufficient. For unusual applications where a more precise force output capability is necessary,

some precision force reference such as a load cell, calibrated spring, or mass/pulley arrangement will be required to provide the precise value for k_1.

Rewriting (4.21) in the form

$$[, _1 \ , _2 \ \cdots \ , _{55}] \begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_{55} \end{bmatrix} = 0 \ , \tag{4.22}$$

makes it straightforward to show that

$$[, _2 \ \cdots \ , _{55}] \begin{bmatrix} k_2 \\ \vdots \\ k_{55} \end{bmatrix} = -, _1 k_1 \ . \tag{4.23}$$

Define $, ' := [, _2 \ \cdots \ , _{55}]$, $k' = [k_2 \ \cdots \ k_{55}]$, and $G_a := -, _1 k_1$, and it is clear that this problem can be solved, similar to the platen sensor autonomous calibration (4.8), using

$$\hat{k'} = , '^{\dagger} G_a. \tag{4.24}$$

The condition number and sensitivity analyses can be performed just as described in Section 4.1.1.

4.3.2 Experiments

Using a PD controller (to be described in Chapter 5) commanding a *fixed-phase* commutator on the *normag2* system, the robot was commanded to move to a series of uniformly distributed random q positions over a range of $-3\rho < q_{\{x,y\}} < 3\rho$ and $-0.2° < q_\theta < 0.2°$.

Because of the redundancy in the system, there will be more than one combination of actuator forces $[f_1 \ f_2 \ f_3 \ f_4]$ that result in a null output wrench $w = [f_x \ f_y \ \tau]$. Without redundancy, the actuator outputs would all have to be nominally zero at an equilibrium position. In this case, there is one extra degree-of-freedom, which allows testing of the full output ranges of the actuators. The actuators are used to push against each other such that the x forces, y forces, and torques simultaneously cancel. For example, if all four actuators of Figure 4.7 have the same force output magnitude f_{int}, f_1 and f_3 push in the directions indicated, and f_2 and f_4

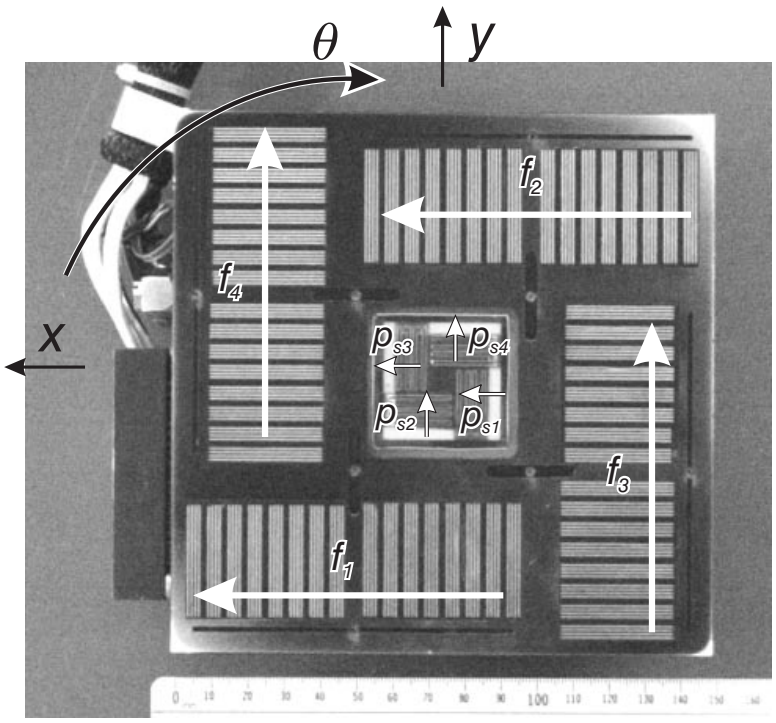


Figure 4.7: Forcer for *normag2* system

push opposite to their indicated directions, there will be no net force or torque on the forcer body. The family of such solutions with zero net output wrench can be parameterized by the nominal internal force f_{int}. The controller computes actuator forces based on f_{int} and the closed-loop wrench command $u_w := [f_{c,x}\ f_{c,y}\ \tau_c]$ as

$$\begin{aligned} f_1 &= f_{int} + f_{c,x}/2 + \sigma_\tau \tfrac{\tau_c}{2d_a} \\ f_2 &= -f_{int} + f_{c,x}/2 + \sigma_\tau \tfrac{\tau_c}{2d_a} \\ f_3 &= -f_{int} + f_{c,y}/2 + (1-\sigma_\tau)\tfrac{\tau_c}{2d_a} \\ f_4 &= f_{int} + f_{c,y}/2 + (1-\sigma_\tau)\tfrac{\tau_c}{2d_a} \end{aligned} , \tag{4.25}$$

where $0 \leq \sigma_\tau \leq 1$ is chosen dynamically to distribute the closed-loop torque responsibility between the x and y actuator pairs based on their relative remaining force capacities. At each test position, f_{int} is chosen over the full force range $-26\text{N} < f_{int} < 26$ N using a uniformly random distribution.

Some care was taken to ensure the tether was not applying large forces or torques to the robot body, but no special tether management devices were used. It was necessary to let the controller settle at each position to ensure that it was at equilibrium, but data collection still took less than 20 min. A difficulty was that at high f_{int} values, the PD controller had very little room before actuator saturation would occur,

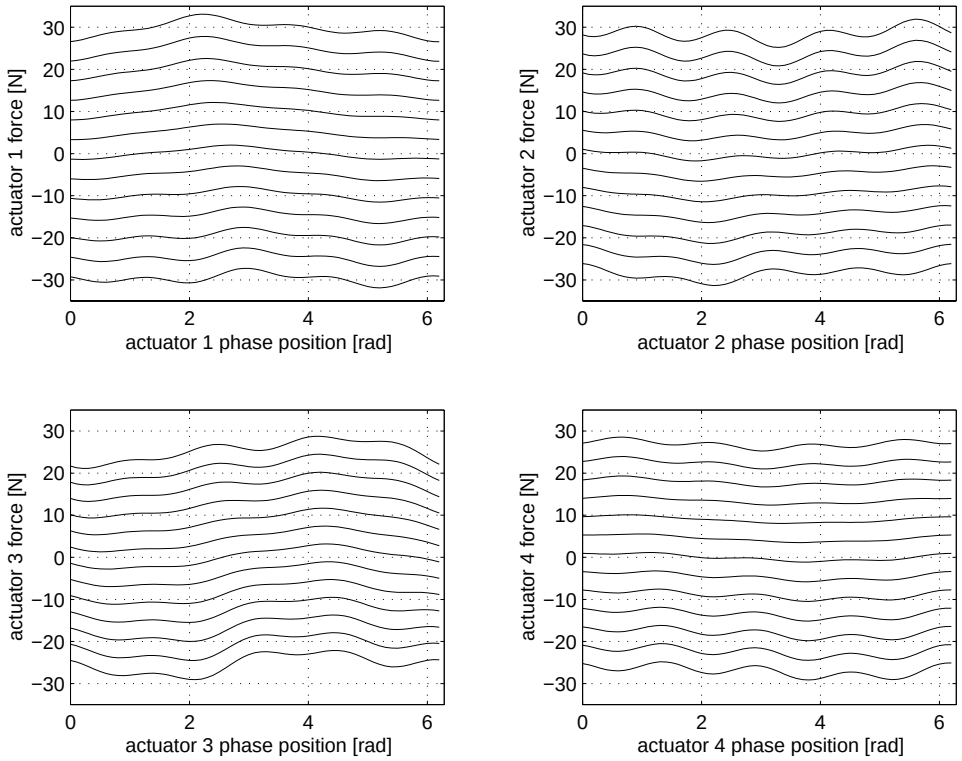


Figure 4.8: The calibrated output force is shown for each of the four actuators. Each trace shows the force output as a function of the actuator phase position ($\phi_a = 2\pi/\rho p_a$) for a constant current input, with traces corresponding to currents from -3 A to 3 A in 0.5 A increments.

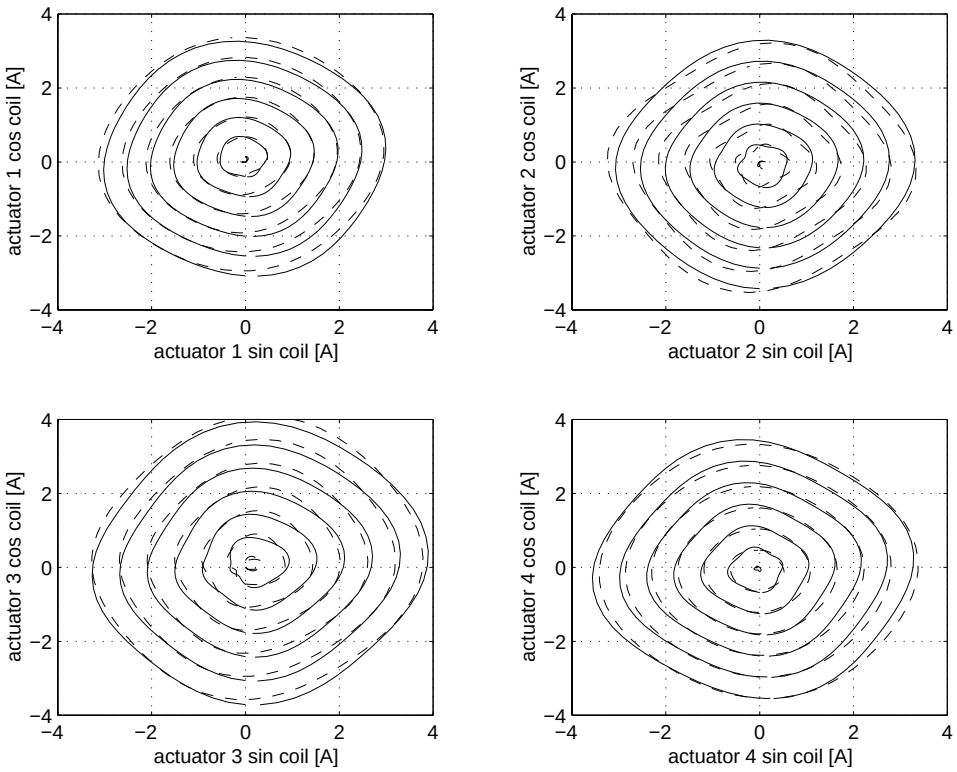


Figure 4.9: The calibrated output force is shown for each of the four actuators. Each loop shows the current amplitudes required to generate force output for a constant force as the spatial phase angle varies. Traces correspond to forces from -30 N to 30 N in 5 N increments, with dashed lines indicating positive output force, and solid lines indicating negative output force.

so that certain positions resulted in the robot not setting at the desired position. The output of the data collection is 2000 vectors of calibrated platen sensor outputs p_{a_i} and the currents I_i applied to the actuators. These values were the averages of 200 samples taken at 1 kHz. The range of values for the 200 samples was examined to ensure that the robot was settling at a position. Of the 2000 data vectors, 85 of the vectors were discarded because of excessive motions at those test positions, most likely resulting from the effects of errors in the uncalibrated actuator outputs on the controller gains.

Using this dataset, the matrix $,'$ was constructed from p_{a_i} and I_i using the equations in the previous section. The condition number of $,'$ was 7.7, indicating good conditioning of the problem. The calibrated parameter vector $\hat{k}$, listed in Table 4.2, was found using (4.24). To examine the resulting force profiles of the actuators, the model for each actuator was used to compute the calibrated actuator force over a series of positions and input currents, shown in Figure 4.8. An alternate view, Figure 4.9, shows the currents the commutator would send to each of the coils as a function of position for a series of force command values. Note that in polar coordinates, the angle corresponds to the actuator position and the radius corresponds to the magnitude of the current output. There is significant variation between the four actuators. Actuator 3 in particular requires significantly larger currents to generate the same output force. Manufacturing variations such as alignment of the two linear motor segments, bent actuator teeth, and the number of windings in the coils may all be partially responsible.

The sensitivity values σ_i' were computed as they were for the platen sensor autonomous calibration to examine how the parameter estimates affect the accuracy of the fit. The sensitivities ranged from 0.018 to 0.041, with the largest values corresponding to current gain parameters k_8 and k_9 and the torque offset τ_e, but all the sensitivities are small enough to not be of concern.

To check the performance of the identification, several residuals can be examined: $r = ,k$, $r_{fx} = ,_{fx}k$, $r_{fy} = ,_{fy}k$, and $r_\tau = ,_\tau k$. Each residual is computed for both the nominal and calibrated model parameters, and their mean and standard deviations are shown in Table 4.1. Note that the mean error values are reduced to near zero, while all standard deviations are reduced by more than 50%, except for r_{fx}, which was relatively small to begin with. These results suggest the calibration is doing fairly well, but that there are significant unmodeled effects. These effects may

	Nominal [N]	Calibrated [N]	Improvement [%]
mean(r)	-0.707	0.000	n/a
mean(r_{fx})	-0.196	0.000	n/a
mean(r_{fy})	-0.071	-0.000	n/a
mean(r_τ)	-1.876	-0.000	n/a
std(r)	2.756	1.090	60
std(r_{fx})	1.688	1.061	37
std(r_{fy})	2.545	1.089	57
std(r_τ)	3.827	1.360	64

Table 4.1: Residuals for actuator calibration using nominal and calibrated parameters

be from a number of sources, such as tether force disturbances, sensor calibration errors, unmodeled amplifier effects, and unmodeled actuator characteristics.

To gain insight into the remaining residual errors, each is plotted against the input parameters $q_x, q_y, q_\theta, f_{int}$ in Figures 4.10, 4.11, 4.12, and 4.13. Although significant structure can be seen for the nominal parameters, especially for r_{fy} vs. q_y (Figure 4.11) and r_τ and r vs. q_x and q_y (Figures 4.12 and 4.13 there is very little structure visible for the calibrated residuals, suggesting that (4.18) is doing a reasonable job of modeling the actuator force output. Note that the PD controller used for data collection has significant steady-state errors and some data was discarded due to excessive noise, so that the test positions were not truly random and there is grouping visible in these figures.

To provide an independent verification of the calibration, tests were performed using a reference spring and several reference masses. For the spring test, shown in Figure 4.14, an extension spring with a spring constant of approximately 400 N/m was connected between the forcer and a point fixed in the workspace. Two tests were performed, with the spring aligned with either the x or y axis, and attached to the forcer at the locations depicted in Figure 4.14. Using a PID controller, the forcer was commanded to move such that it extended the spring until it reached a preset force limit. From that position, the controller setpoint was moved in steps such that the spring was retracted with the controller settling at each position step, where the position and current commands were recorded.

For the x axis test, 365 positions were tested with step sizes of $\rho/10$. The recorded data were used to compute the output force as determined by the linear actuator model with both the nominal and calibrated parameters. These forces

Parameter no.	k_0	$\hat{k}$	Param.	Nominal	Calibrated
$k_{1,1}$	9.895	9.895	$k_{3,1}$	9.895	8.459
$k_{1,2}$	0.000	0.135	$k_{3,2}$	0.000	0.067
$k_{1,3}$	0.000	-0.181	$k_{3,3}$	0.000	-0.148
$k_{1,4}$	0.000	-0.148	$k_{3,4}$	0.000	-0.245
$k_{1,5}$	0.000	-0.239	$k_{3,5}$	0.000	-0.308
$k_{1,6}$	0.000	0.192	$k_{3,6}$	0.000	0.021
$k_{1,7}$	0.000	-0.163	$k_{3,7}$	0.000	-0.307
$k_{1,8}$	0.000	0.645	$k_{3,8}$	0.000	-2.357
$k_{1,9}$	0.000	-1.410	$k_{3,9}$	0.000	-1.415
$k_{1,10}$	0.000	-0.184	$k_{3,10}$	0.000	0.041
$k_{1,11}$	0.000	0.150	$k_{3,11}$	0.000	-0.078
$k_{1,12}$	0.000	-0.213	$k_{3,12}$	0.000	-0.379
$k_{1,13}$	0.000	-0.063	$k_{3,13}$	0.000	0.073
$k_{2,1}$	9.895	9.506	$k_{4,1}$	9.895	8.992
$k_{2,2}$	0.000	0.120	$k_{4,2}$	0.000	-0.153
$k_{2,3}$	0.000	0.120	$k_{4,3}$	0.000	0.060
$k_{2,4}$	0.000	-0.156	$k_{4,4}$	0.000	0.012
$k_{2,5}$	0.000	-0.164	$k_{4,5}$	0.000	-0.075
$k_{2,6}$	0.000	-0.105	$k_{4,6}$	0.000	0.203
$k_{2,7}$	0.000	-0.402	$k_{4,7}$	0.000	-0.246
$k_{2,8}$	0.000	-0.752	$k_{4,8}$	0.000	0.666
$k_{2,9}$	0.000	1.005	$k_{4,9}$	0.000	0.717
$k_{2,10}$	0.000	-0.057	$k_{4,10}$	0.000	0.064
$k_{2,11}$	0.000	0.250	$k_{4,11}$	0.000	0.096
$k_{2,12}$	0.000	-0.505	$k_{4,12}$	0.000	-0.262
$k_{2,13}$	0.000	-0.215	$k_{4,13}$	0.000	0.127
			$k_{e,1}$	0.000	0.333
			$k_{e,2}$	0.000	0.120
			$k_{e,3}$	0.000	1.520

Table 4.2: Nominal and autonomously calibrated actuator parameters

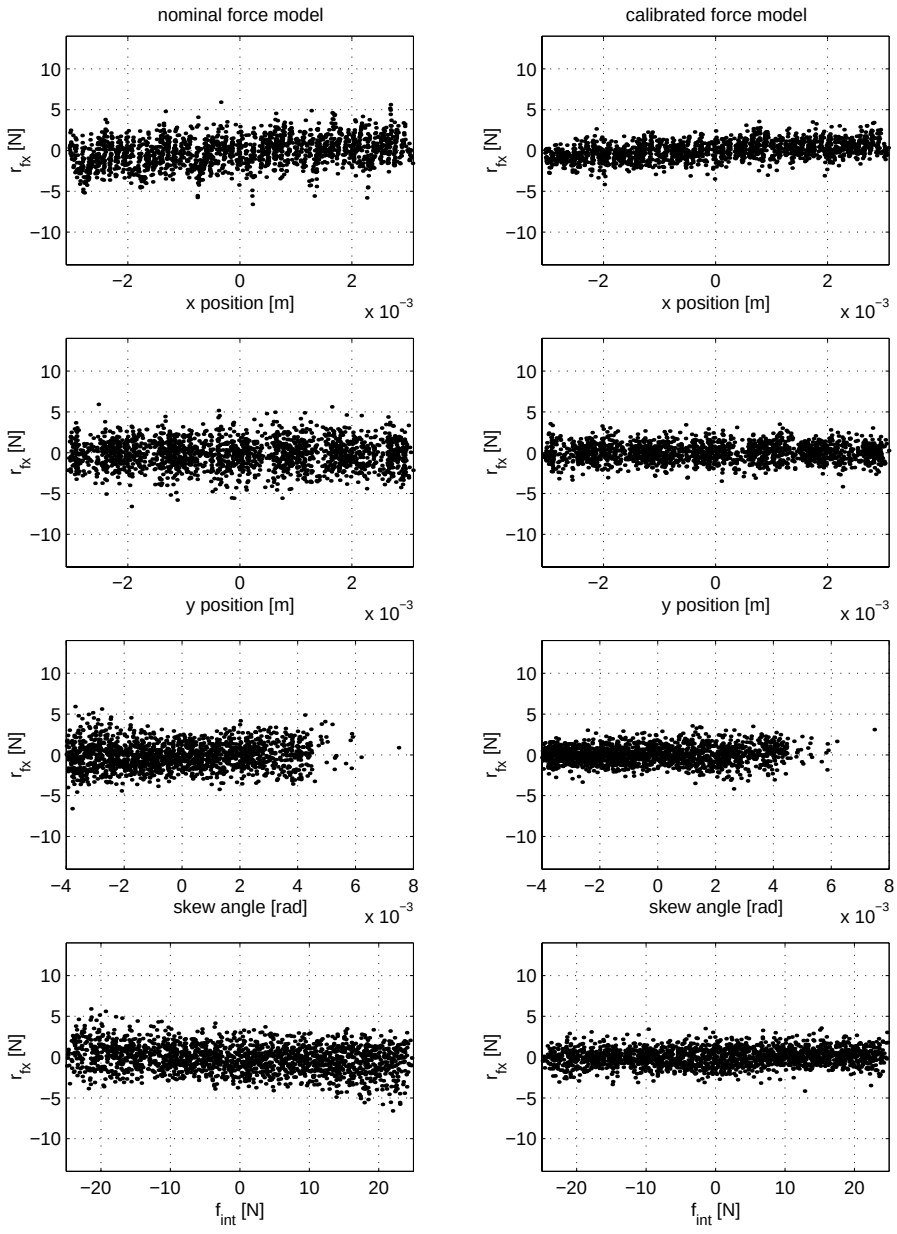


Figure 4.10: Dependence of r_{fx} residual on input position and force

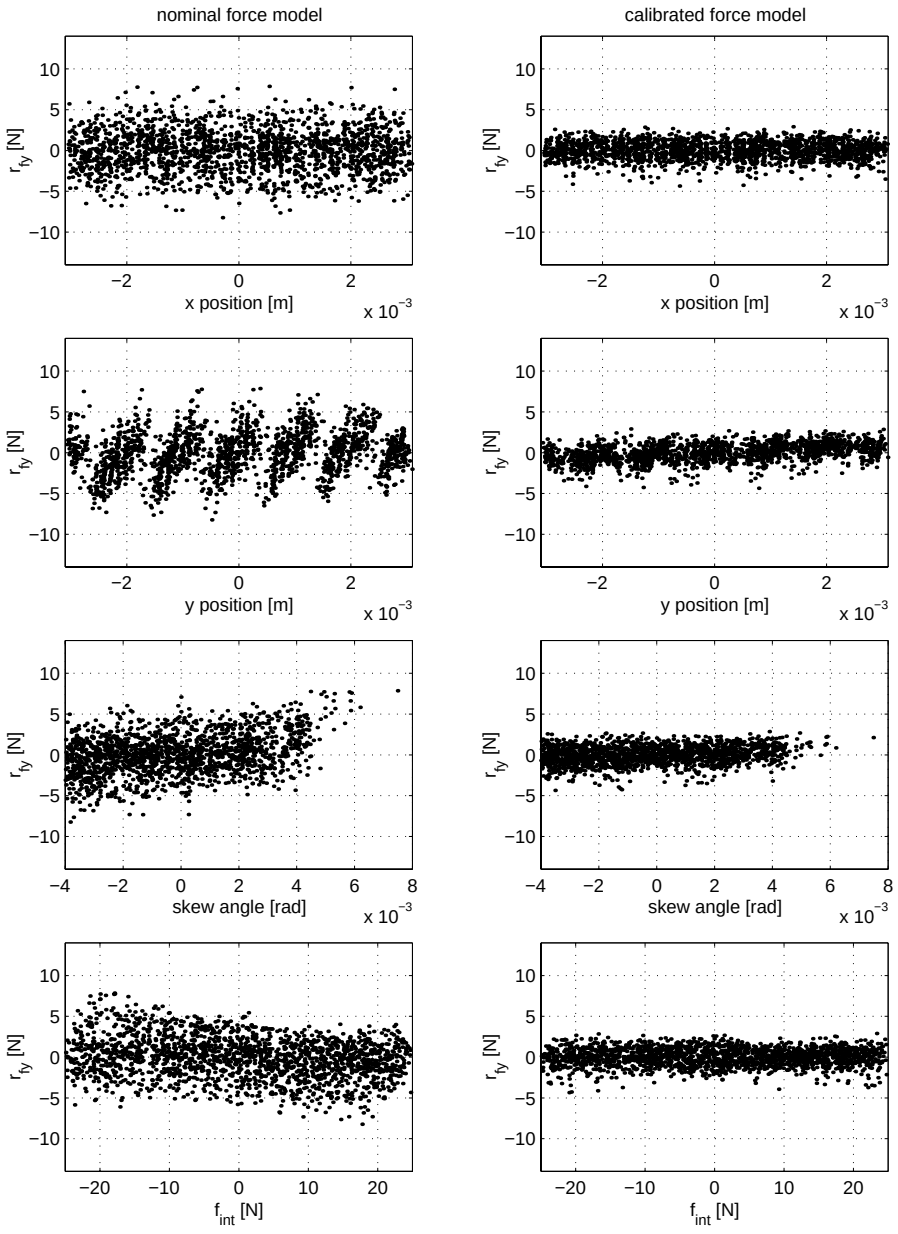


Figure 4.11: Dependence of r_{fy} residual on input position and force

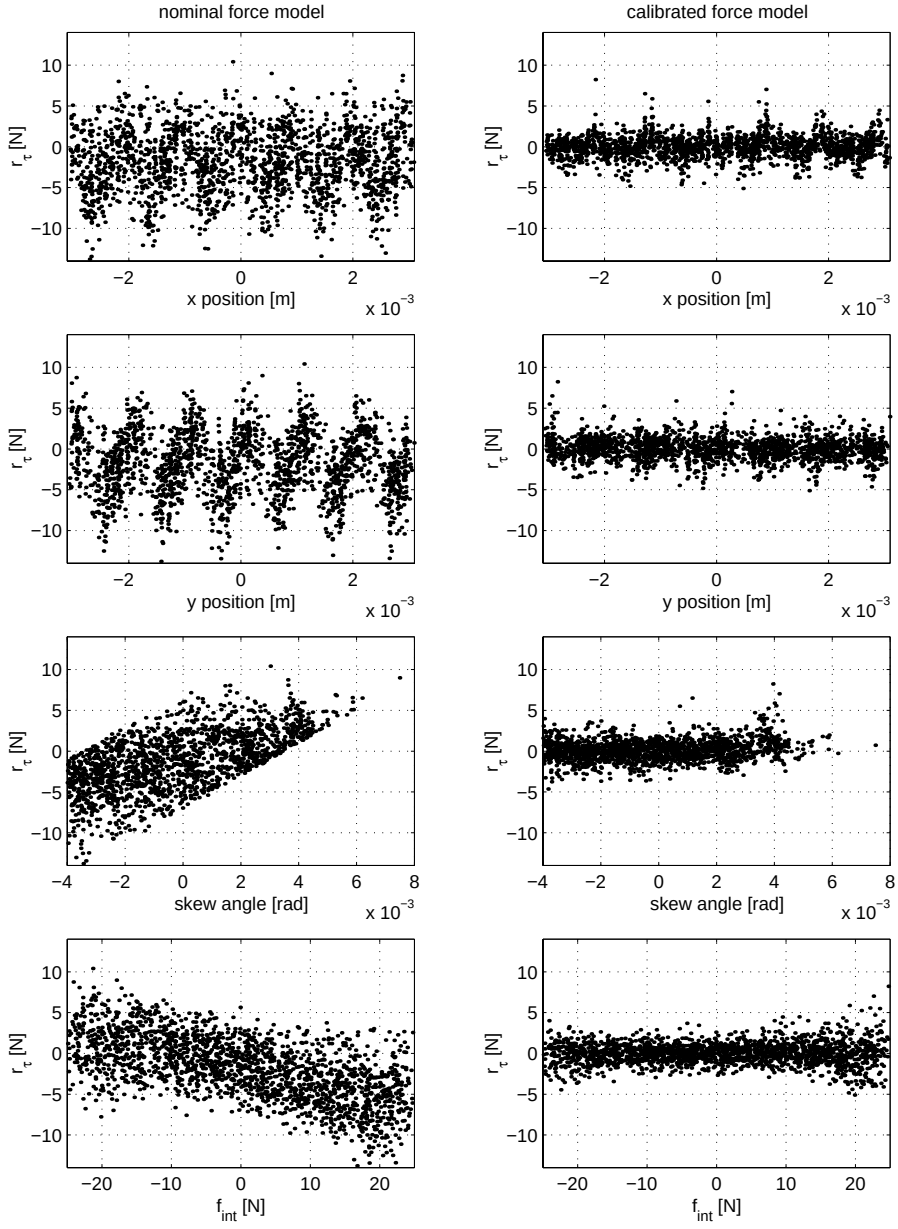


Figure 4.12: Dependence of r_τ residual on input position and force

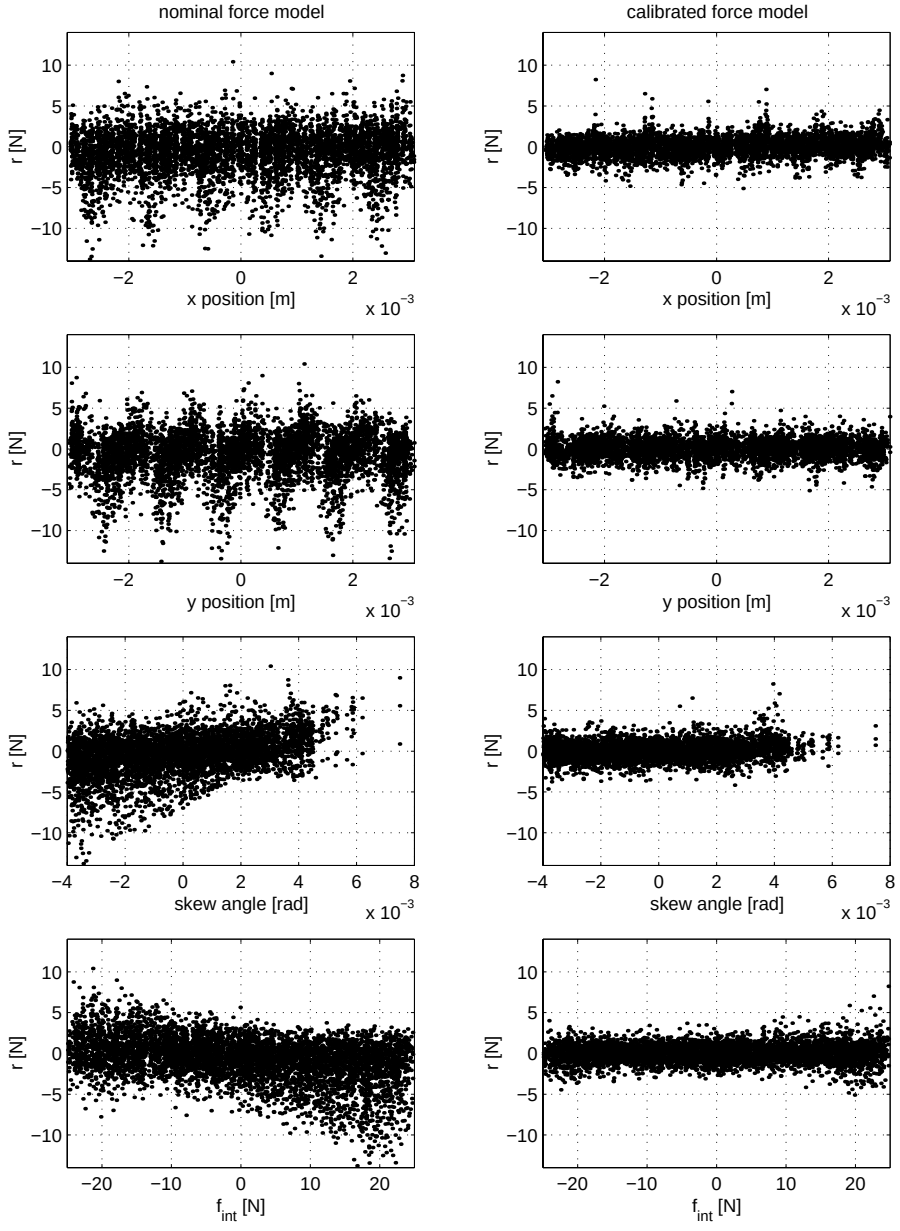


Figure 4.13: Dependence of r residual on input position and force

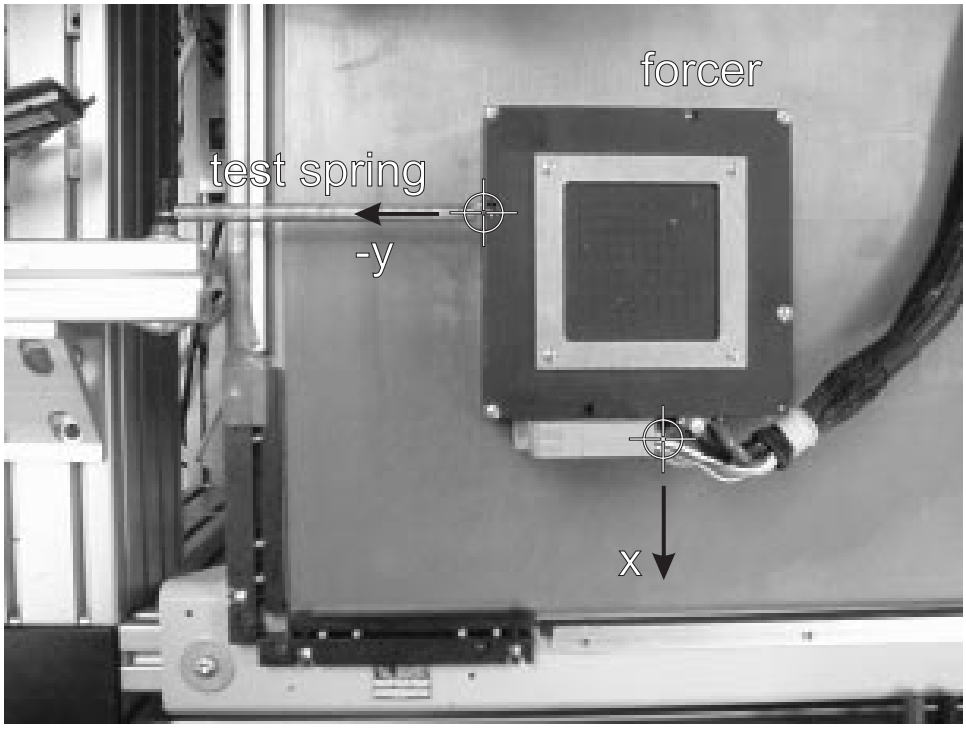


Figure 4.14: A spring is attached between the forcer and a fixed location to provide a force reference for verifying the actuator calibration results. The y axis test is shown here; the x axis test used the indicated attachment location.

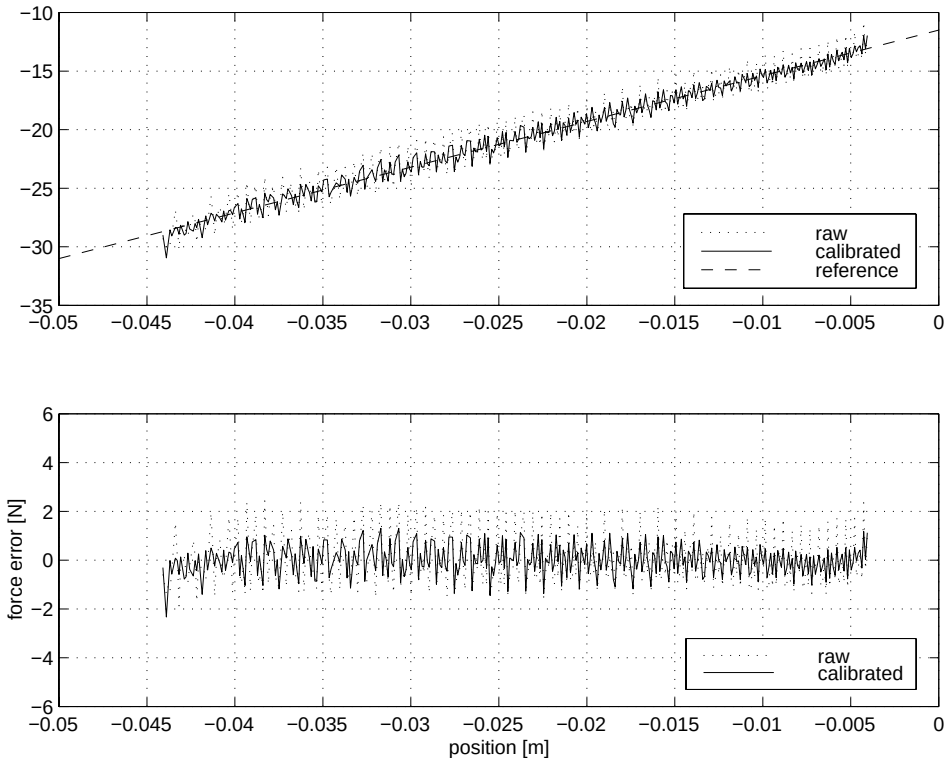


Figure 4.15: For the x axis spring pull test, the reconstructed force for nominal and calibrated (upper) models are shown. The deviations from a straight line fit (lower) indicate errors in both force models.

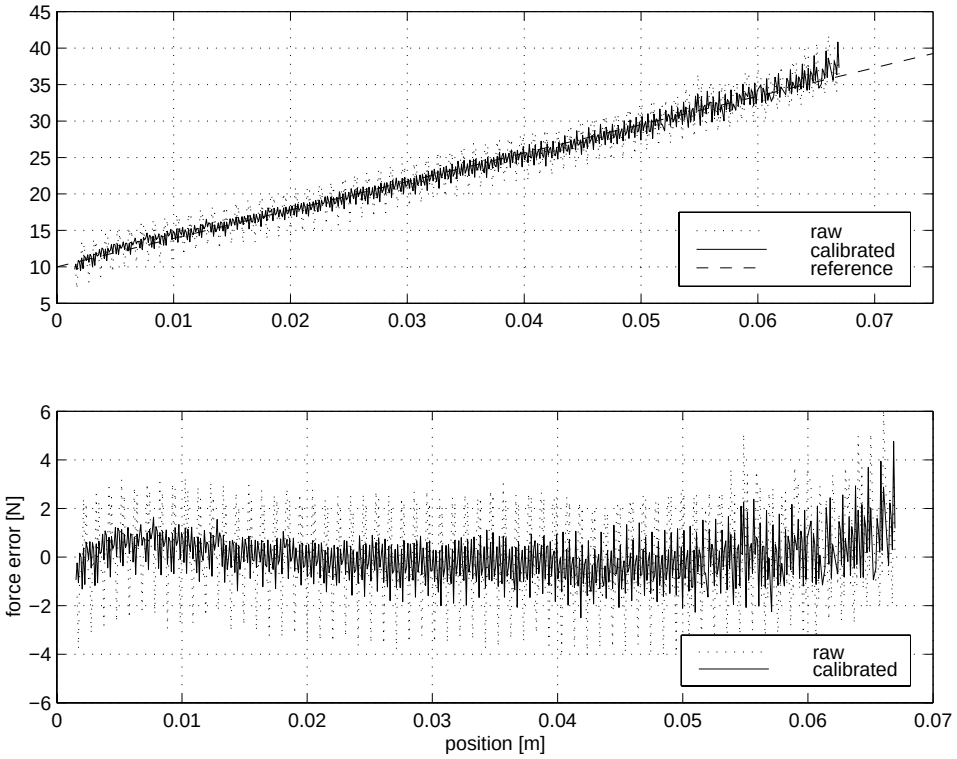


Figure 4.16: For the y axis spring pull test, the reconstructed nominal and calibrated (upper) forces are shown. The deviations from a straight line fit (lower) indicate errors in the force model.

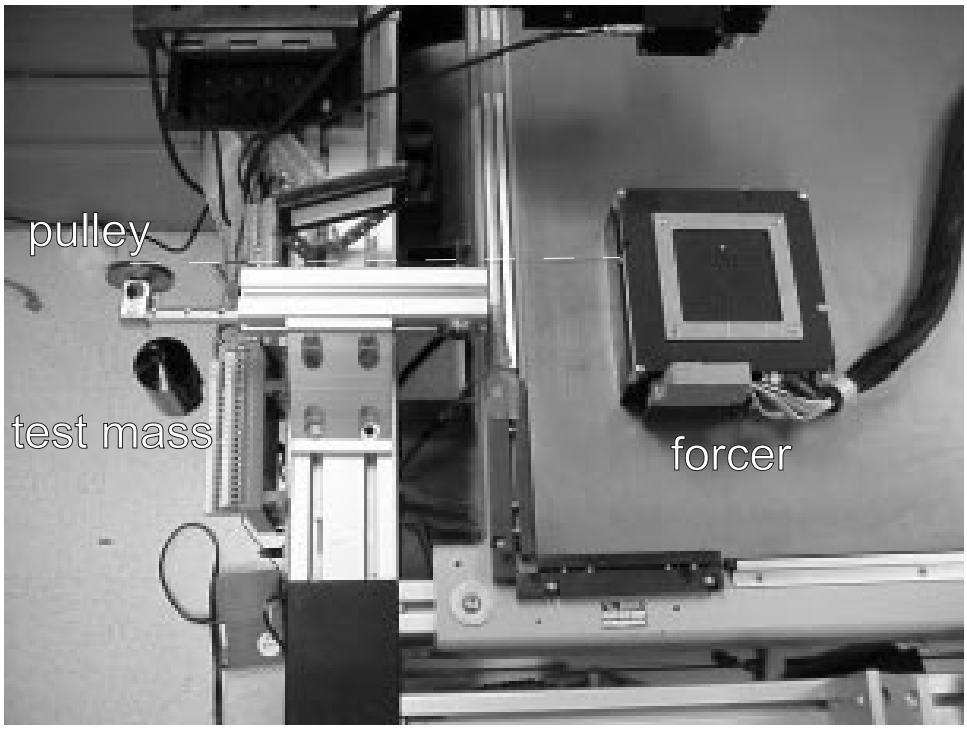


Figure 4.17: A mass/pulley arrangement is used to provide a force reference for verifying the actuator calibration results. The attachment points were the same as in the spring test.

are plotted against the forcer position, as reported by the platen sensor, in the upper plots of Figure 4.15. With a perfect force model (and a perfect spring and no disturbances), these plots should be straight lines having a slope of the spring constant–any deviation indicates a probable error in the calibration. Here, the plots are scaled and offset such that a line fit to them will have the same slope as the spring and have zero offset. The deviation from the linear fit for both sets of parameters is shown in the lower plots of Figure 4.15. As expected, there is significant error for the nominal parameters, with a standard deviation of 1.08 N, but with the calibrated model, the error is reduced by 38% to 0.67 N.

Results were similar for the y axis test, which consisted of 625 positions spaced at $\rho/10$ increments, as shown in Figure 4.16. The standard deviation of the force error decreased from 1.82 N to 1.07 N. Note that at high force values, some nonlinearity can be seen, but this effect is small relative to the force noise.

As an additional verification, a mass/pulley arrangement was used to provide a constant external force, as shown in Figure 4.17. A series of different masses was

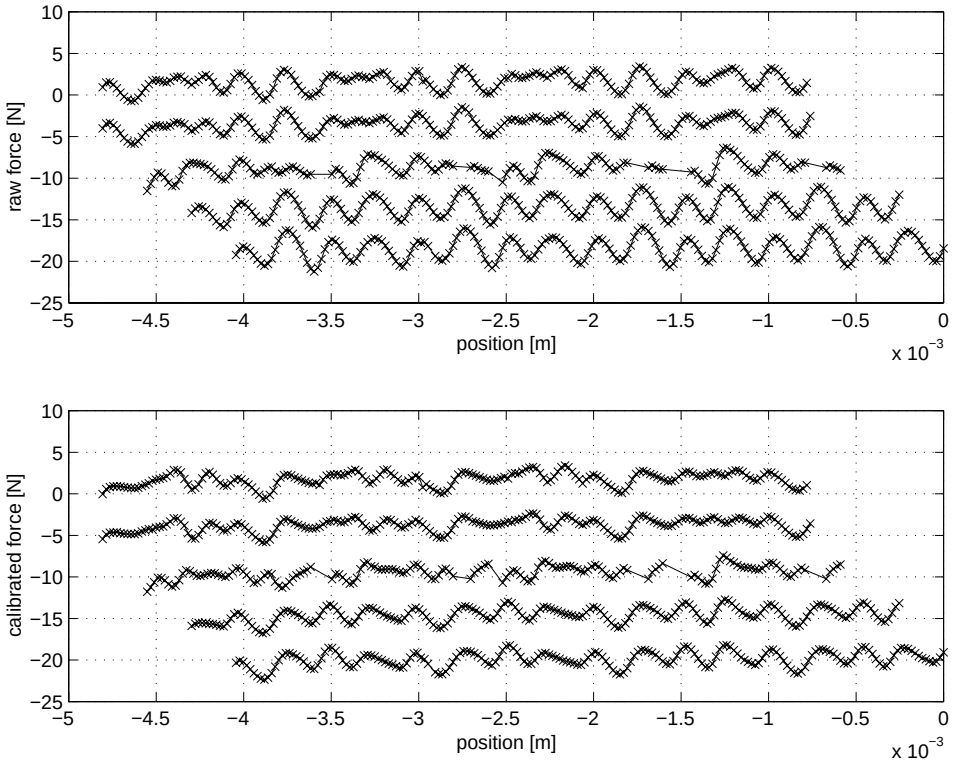


Figure 4.18: For the x axis mass/pulley test, the reconstructed force for nominal (upper) and calibrated (lower) are shown for masses of 0, 0.5, 1.0, 1.50, and 2 kg. Deviations from horizontal indicate errors in the force model.

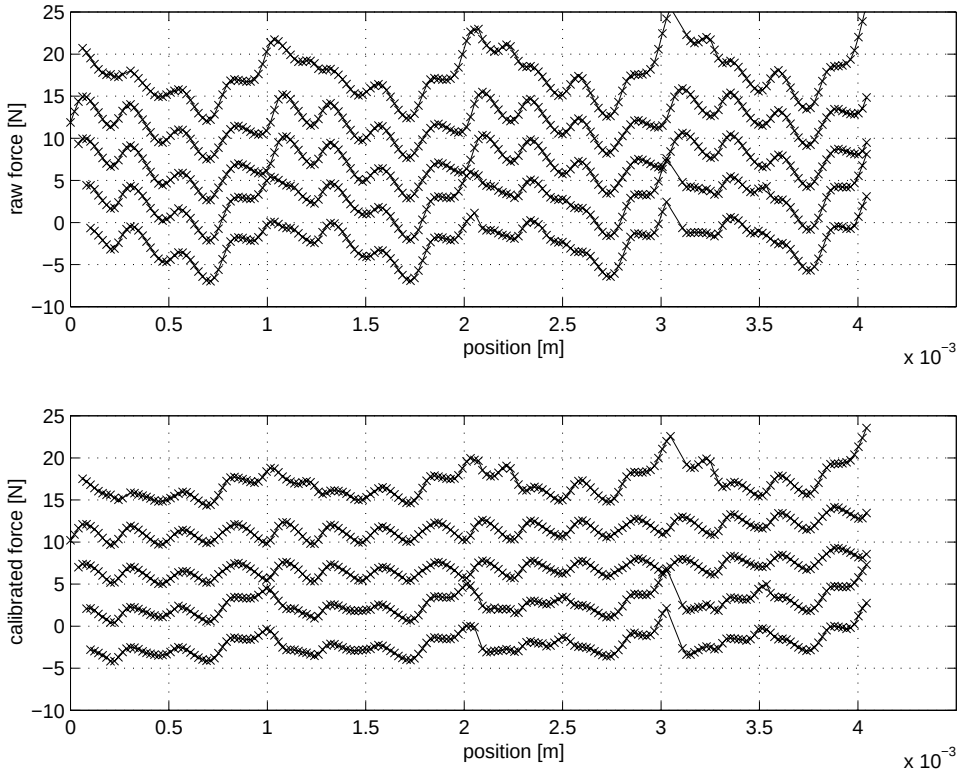


Figure 4.19: For the y axis mass/pulley test, the reconstructed force for nominal (upper) and calibrated (lower) are shown for masses of 0.0, 0.5, 1.0, 1.5, and 2.0 kg. Deviations from horizontal indicate errors in the force model.

suspended, with the forcer moving to a sequence of positions under PID control. An integral term is used here to compensate for the large, constant force disturbance. A test pulling the mass in the x direction recorded the amplifier currents at approximately 200 locations spaced at $\rho/50$. The force output of the forcer, which should have been constant, was computed based on these data for both the nominal and calibrated parameters. As shown in Figure 4.18 for multiple test masses ranging from 0 to 2 kg, calibration removes a portion of the ripple at all force values, but there remain significant errors in the force. The standard deviations of each trace are reported in the top half of Table 4.3 and demonstrate a quantitative improvement of between 17 and 31 percent, bringing all of the values under 1 N. The remaining errors may be from an imperfect actuator model, noise in the calibration data, or non-uniformities in the platen teeth.

The mass/pulley experiment was repeated for the y axis for approximately 200 points taken with a spacing of $\rho/50$. As shown in Figure 4.19, results are similar to the case for the x axis, with an improvement after calibration but a significant error remaining. The standard deviations of each trace are reported in the lower half of Table 4.3 and demonstrate a quantitative improvement of between 37 and 51 percent, reducing the force errors to below 1.26 N, except for the 2.0 kg test. The y axis has larger nominal errors in the force than the x axis and appears to be the dominant source of error. One difficulty with the calibration of these actuators is that they are permanently mounted in a single housing, and cannot be tested individually. The permanent magnets of the other actuators will generate detent forces even if no currents are in their coils. In addition, if any actuator has large unmodeled errors, the calibration procedure will tend to spread the error between all the actuators.

Both the spring and mass tests confirm the results predicted by the residuals–the autonomous calibration reduces the actuator force error by approximately 30-50%. Further improvements may be possible with a data collection procedure that isolates tether forces better, a platen with more uniform teeth, a richer data set, or a more precise model of the actuator force output, especially in how ripple changes with the size of the nominal force magnitude and the robot skew angle.

Nevertheless, the calibration procedure can be used to significantly improve the force output accuracy of the planar robot without requiring any external devices. The data collection procedure is fully automated and is completed within 20 min.

Axis	mass [kg]	Nominal	Calibrated	Improvement [%]
x	2.0	1.316	0.907	31
x	1.5	1.229	0.879	28
x	1.0	0.975	0.795	19
x	0.5	0.945	0.768	19
x	0.0	0.986	0.815	17
y	2.0	2.740	1.700	38
y	1.5	1.930	1.005	48
y	1.0	1.888	0.923	51
y	0.5	2.028	1.258	38
y	0.0	1.989	1.245	37

Table 4.3: Standard deviations of the computed force required to support the listed test masses indicate how much the autonomous calibration is smoothing the force output.

Parameter identification uses the well-known and stable singular value decomposition which takes less than 20 s on a Sparcstation 20. Examination of the recovered singular values and residuals can also provide some automated verification of the calibration. If it appears that some of the parameters are not being properly identified or there is a large amount of unmodeled effects, the system can warn the operator before proceeding to install the new parameters. In summary, the results presented in this section support the conclusion that autonomous calibration can be effective in reducing the ripple of the planar robot actuators.

4.4 Real-time implementation

The models developed in this chapter are used by the commutator to compute the forcer position based on the sensor input signals, and the actuator output currents based on the latest force commands from the controller. As this commutator will run at kHz rates, there must be an efficient real-time implementation of the models.

The platen sensor error model (4.1) must be combined with the nominal sensor model to compute the calibrated sensor outputs. For each sensing element,

$$\begin{aligned}\hat{p}_{s,i} &= \text{unwrap}(\check{p}_{s,i} + e_i, \check{p}_{s,i,prev} + e_{i,prev}, \hat{p}_{s,i,prev}) \\ &= \text{unwrap}(\text{atan2}(s_{2i-1} - z_{2i-1}, s_{2i} - z_{2i}) + k_{i,0} + \\ &\quad k_{i,1}\sin(\omega_\rho \check{p}_{s,i}) + k_{i,2}\cos(\omega_\rho \check{p}_{s,i}) + k_{i,3}\sin(2\omega_\rho \check{p}_{s,i}) + k_{i,4}\cos(2\omega_\rho \check{p}_{s,i}) + \\ &\quad k_{i,5}\sin(4\omega_\rho \check{p}_{s,i}) + k_{i,6}\cos(4\omega_\rho \check{p}_{s,i}) + k_{i,7}\sin(6\omega_\rho \check{p}_{s,i}) + k_{i,8}\cos(6\omega_\rho \check{p}_{s,i}), \\ &\quad \check{p}_{s,i,prev} + e_{i,prev}, \hat{p}_{s,i,prev})\end{aligned} \tag{4.26}$$

where s_j, $j \in \{1, \ldots 8\}$, are the raw sensor outputs in ADC converter counts, and z_j, $j \in \{1, \ldots 8\}$, are the raw sensor offset values. The offset values are used to compensate for the sensor output voltage offset levels so that the quadrature circle is centered at zero. The offsets are determined by a simple procedure where the mean raw sensor outputs are computed over a low velocity diagonal motion covering several pitches. The unwrap function is used to keep track of motions that move beyond a single tooth, and is defined (using x, y as placeholder variables) as

$$y = \text{unwrap}(x, x_{prev}, y_{prev}) = \begin{cases} x - x_{prev} > \pi & y_{prev} + (x - y) - 2\pi \\ x - x_{prev} < -\pi & y_{prev} + (x - y) + 2\pi \\ |x - x_{prev}| \leq \pi & y_{prev} + (x - y) \end{cases} . \tag{4.27}$$

The computation of most of the transcendental functions of (4.26) is avoided by storing the eight sine and cosine functions in a lookup table indexed by the sensor position $\check{p}_{s,i}$ expressed as a phase in the range $0 \leq \check{p}_{s,i} < 2\pi$. The lookup table has 6000 entries and is used without interpolation. This size is sufficient given the inherent sensor resolution. The arctangent function cannot be easily avoided without adding conditionals and divides, but only four are required per sensing sample. The remainder of the computation consists of multiplying the sine and cosines by the appropriate parameters and summing the results. These operations are efficiently pipelined on the real-time PowerPC machine. The complete sensor computation including ADC transfers, nominal and error model computation, bookkeeping to compute phase wraparounds, and combining the four sensor outputs to compute the forcer position takes approximately 30 μs, compared to the 200 μs cycle time of the commutator running at 5 kHz.

The actuator force model as presented computes the output force based on the actuator position and current input. This model must be inverted with respect to

the current to be useful to the commutator. Note that the force model (4.18) can be rewritten in the form

$$f_i = (a_{1,i}(p_{a_i}) + a_{2,i}(p_{a_i})u_i)\cos(k_\theta q_\theta) \ , \tag{4.28}$$

where $a_{1,i}(\cdot)$ collects the offset force terms and $a_{2,i}(\cdot)$ collects the current dependent terms. This relation can be easily inverted

$$I_i = \left(\frac{f_i}{\cos(k_\theta q_\theta)} - a_{1,i}(p_{a_i})\right)/a_{2,i}(p_{a_i}) \ , \tag{4.29}$$

so that the current can now be computed. Because $a_{1,i}$ and $a_{2,i}$ are functions of exclusively p_{a_i}, a one-dimensional lookup table is used for their computation. To save a divide, the inverse of $a_{2,i}$ is stored. A table with only 100 entries and linearly interpolated outputs is used to reduce memory requirements. The computation of the calibrated current output takes less than 28 μs, including computation of the actuator phase positions based on the forcer position, computation of the calibrated current outputs for each of four channels, and scaling and writing the results to the DAC outputs.

The autonomous calibration techniques of this chapter, though limited by the uniformity of the platen structures, demonstrate how the experimental measurements of the previous chapter can be exploited by the robot for improving its own performance. It is also interesting to note that well-conditioned calibration with a more than dozen parameters per DOF are possible with this low-DOF robot, allowing the few DOFs to be calibrated in more detail than usually possible with long serial chain robot manipulators.

Chapter 5

Control

The commutation, modeling, and calibration of the sensors and actuators discussed in the previous three chapters all are intended to make control easier. Commutation allows for fewer control variables and allows the actuator models to take the familiar form of current input to force output. Modeling and autonomous calibration improves the fidelity of the actuator and sensor models, reducing disturbances to the controller.

The *planar robot control problem* is then to choose forces f_i $i \in 1...4$ based on the position measurements q and knowledge of the dynamic model of the robot. Many of the problems that make the control of conventional (i.e. serial chain manipulators) robots difficult are eliminated by the planar robot design. There is only a single moving part, so the kinematics and dynamics are trivial, the performance won't vary with its joint angles, and because the robot is effectively rigid there are no link dynamics to worry about. The actuators are direct-drive and rely on air bearings, so that drive-train friction and compliance will not complicate the control. The large force-to-mass ratio of the robot combined with the high-bandwidth of the platen sensor allows for a high dynamic range.

However, the design of the planar robot also creates some problems. Although great pains, as detailed in the previous two chapters, have been taken, the realities of platen fabrication and error modeling are that there will always be some residual error in the sensor position measurements and actuator force outputs. These errors, which are a function of the robot location relative to the teeth, reflect the stepping motor origins of the planar robot, designed so that force is a strong function of position. During motions, these errors will act as a bounded disturbance with

fundamental frequencies at multiples of the robot speed, when measured in teeth per second. At rest, these errors can cause steady-state errors in the robot position.

Another problem is the limited rotational range of the robot. If it should rotate beyond this range, it will be unable to recover. Most devices would have mechanical limit stops to prevent such an occurrence, but the simple design of the robot makes this impossible and the controller must take full responsibility. Note that the same actuators are used for generating torques and forces, so that the controller must carefully balance the need to maintain the angle limits and the need to perform the task at hand.

Unlike most robots, which have to derate their actuators to prevent the excitation of link or drivetrain dynamics and the excessive wear of drivetrain components, the mechanical simplicity of the planar robot allows the use of the full capabilities of its actuators. While this characteristic is beneficial, it does require a more sophisticated controller to handle the nonlinearities that result when the robot is operated with the peak outputs of the actuators.

The robot must also drag around a tether that is of significant size and mass compared to that of the forcer body. The controller must be prepared to use a significant fraction of the actuator capabilities to reject tether disturbances for the case where the tether gets caught on the platen edge or is bumped by another robot. For normal situations, the tether will impart a relatively small and quasi-static force and torque disturbance, which will also need to be rejected. It is also possible to excite the dynamics of the flexible tether, but the size and frequency of the motions required would be unusual for most applications.

Multiple planar robots can share the same platen. Rather than a global position sensor, each has a platen sensor that measures position relative to the position where they are turned on. It would be desirable to allow the robot to collide (at least at moderate speed) with other planar robots or stationary workspace obstacles during the startup exploration without it leading to unrecoverable skew angles.

This chapter contains two PD/PID strategies for closed-loop control of the planar robot. Although they do not address all of the above issues, experiments in this and later chapters show that they are sufficient for a range of tasks. The two controllers differ mainly in their treatment of actuator saturation. In the following section, a block diagram for a saturated PD controller is presented and discussed. In Section 5.2, a domain-based approach is presented that limits the operation of

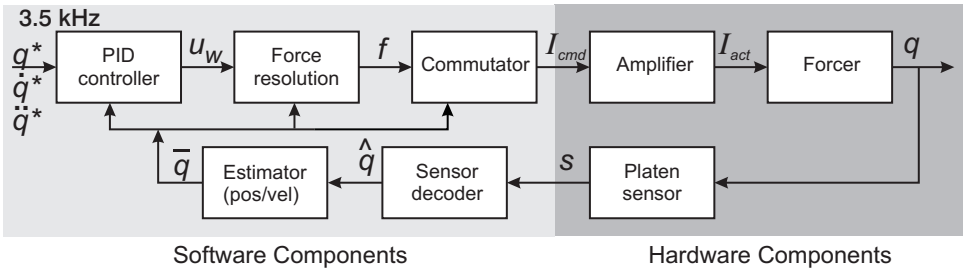


Figure 5.1: Saturated PD controller block diagram

the controller to an invariant region around the goal to prevent actuator saturation from occurring. In Section 5.3 experiments using these controllers demonstrate performance improvements stemming from closed-loop operation. Early experiments using the saturated PD controller demonstrate resolution, repeatability, and trajectory tracking performance, and more recent results using the domain-based PD controller demonstrate advantages related to power consumption and thermal dissipation.

5.1 Controller Formulation

A block diagram of the overall system is shown in Figure 5.1. There are five software blocks required. The *sensor decoder* processes the four raw sensor signals to produce a measure of the positions q_x and q_y and skew angle q_θ of the robot. The *estimator* is used to compute a velocity and position estimate based on the sensor position output and the system dynamics. The *controller* computes a wrench ($u_w = [f_x f_y \tau]^T$) command to be generated by the robot, based on the estimator outputs and desired trajectory. The *force resolution* block is needed to handle the redundancy of the force generation, as the four actuators are used to generate a 3×1 wrench. Finally, the *commutator* computes currents to send to the motor coils based on the forcer position and the desired forces.

It will simplify the presentation to specify wrenches and positions at the center of mass or center of actuation for different blocks. A wrench at the center of actuation, w_{ca}, and an equivalent wrench at the center of mass, w_{cm}, will be related by

$$w_{ca} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -d_{gy} & d_{gx} & 1 \end{bmatrix} w_{cm}, \tag{5.1}$$

where $p = [d_{gx}\ d_{gy}]^T$ is the location of the center of mass of the forcer expressed relative to the (x, y) coordinate system shown in Figure 1.16. Similarly, the positions and velocities are related by:

$$q_{ca} = \begin{bmatrix} 1 & 0 & d_{gy} \\ 0 & 1 & -d_{gx} \\ 0 & 0 & 1 \end{bmatrix} q_{cm} - \begin{bmatrix} d_{gx} \\ d_{gy} \\ 0 \end{bmatrix}, \text{and} \tag{5.2}$$

$$\dot{q}_{ca} = \begin{bmatrix} 1 & 0 & d_{gy} \\ 0 & 1 & -d_{gx} \\ 0 & 0 & 1 \end{bmatrix} \dot{q}_{cm}, \tag{5.3}$$

where the small rotation range of the motor justifies linearization of the equations about the zero angle.

In this remainder of this section, each software block is examined, with relevant motor modeling introduced as needed.

5.1.1 Sensor decoding

The raw sensor outputs will be voltage measurements from the quadrature pairs of each sensor segment, as described in Chapter 1. The calibrated position of each sensor segment along its direction of motion, $\hat{p}_{s,i}$, is computed using (4.26). The position of the forcer q_{cm} can then be computed

$$q_{cm} = \begin{bmatrix} q_x \\ q_y \\ q_\theta \end{bmatrix} = \begin{bmatrix} 0.5 & 0 & 0.5 & 0 \\ 0 & 0.5 & 0 & 0.5 \\ 0.5/d_s & 0.5/d_s & -0.5/d_s & -0.5/d_s \end{bmatrix} \hat{p}_s. \tag{5.4}$$

Note that there is a redundancy in this computation, which was exploited for the autonomous calibration in Chapter 4. The redundancy also allows the forcer position to be computed in the case where one of the x axis sensor outputs is unusable. For example, if the first sensor is unusable, the forcer position can be computed as

$$q_{cm} = \begin{bmatrix} q_x \\ q_y \\ q_\theta \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0.5 & 0 & 0.5 \\ 0 & 1/d_s & 0 & -1/d_s \end{bmatrix} \hat{p}_s. \tag{5.5}$$

The most common situation where one of the sensor segments becomes unusable is when the platen has a magnetic discontinuity, either because it was constructed from

two pieces, as in Figure 1.12(d), or for the boundaries between the field-joinable platens used in the minifactory application presented in Chapter 7. In practice it is difficult to detect a malfunctioning sensor segment when crossing a platen discontinuity at high speed, so the locations of such discontinuities must be mapped ahead of time so that the appropriate sensor segments can be ignored while they are over a discontinuity. Note that when the sensor segment is re-enabled, it must be re-initialized based on the readings of the other sensor segments so that its tooth count is correct.

5.1.2 State estimation

The estimator exploits the linear dynamic model of the forcer to produce a filtered position and velocity signal without excessive lag. The discrete time estimator takes the form:

$$\bar{Q}_{cm}(k+1) = \begin{bmatrix} \Phi & 0 & 0 \\ 0 & \Phi & 0 \\ 0 & 0 & \Phi \end{bmatrix} \bar{Q}_{cm}(k) + \begin{bmatrix} L & 0 & 0 \\ 0 & L & 0 \\ 0 & 0 & L \end{bmatrix} \tilde{q}_{cm}(k) + \begin{bmatrix} \frac{T^2}{2m} & 0 & 0 \\ \frac{T}{m} & 0 & 0 \\ 0 & \frac{T^2}{2m} & 0 \\ 0 & \frac{T}{m} & 0 \\ 0 & 0 & \frac{T^2}{2I_z} \\ 0 & 0 & \frac{T}{I_z} \end{bmatrix} u_{w,cm} \tag{5.6}$$

with

$$\Phi = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \quad L = \begin{bmatrix} l_1 \\ l_2 \end{bmatrix}, \text{and}$$
$$\tilde{q}_{cm}(k) = \hat{q}_{cm}(k) - \bar{q}_{cm}(k).$$

Here, $\hat{q}_{cm}(k) = [\hat{q}_x \hat{q}_y \hat{q}_z]^T$ is the calibrated forcer position at time k, $\bar{q}_{cm}(k) = [\bar{q}_x \bar{q}_y \bar{q}_z]^T$ is the estimated forcer position at time k, the complete state estimate is given by $\bar{Q}_{cm}(k) = [\bar{q}_x\ \bar{\dot{q}}_x\ \bar{q}_y\ \bar{\dot{q}}_y\ \bar{q}_\theta\ \bar{\dot{q}}_\theta]^T$ for time k, and T is the sample time.

Note that the estimator is decoupled and the estimator gains l_1 and l_2 can be computed based on the desired estimator pole locations using well-known pole-placement techniques (*i.e.* MATLAB's[1] `PLACE` command).

5.1.3 Commutation

Commutation was fully discussed in Chapter 2. A *fixed phase* commutator is used for all experiments in the following chapters. The inverted force model (4.29), $I_i = \left(\frac{f_i}{\cos(k_\theta q_\theta)} - a_{1,i}(p_{a_i})\right)/a_{2,i}(p_{a_i})$ from the previous chapter is used[2] to compute the actuator commands $I_i \in 1...4$ as a function of the desired force outputs f_i and positions p_{a_i}. The actuator positions are computed based on the processed sensor outputs as

$$\begin{bmatrix} p_{a_1} \\ p_{a_2} \\ p_{a_3} \\ p_{a_4} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -d_a \\ 1 & 0 & d_a \\ 0 & 1 & -d_a \\ 0 & 1 & d_a \end{bmatrix} (\hat{q}_{ca} + \dot{\hat{q}}_{ca} t_a), \tag{5.7}$$

where t_a is the phase advance time, as defined in Section 2.1, $\dot{\hat{q}}_{ca}$ is the *estimated* forcer velocity, and $\hat{q}_{ca}$ is the calibrated forcer position, both expressed at the center of actuation.

5.1.4 Force resolution

The force resolution function must consider the force kinematics and force saturation properties of the forcer. The force kinematics are given by:

$$w_{ca} = \begin{bmatrix} f_x^d \\ f_y^d \\ \tau^d \end{bmatrix}_{ca} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ -d_a & d_a & -d_a & d_a \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}, \tag{5.8}$$

where w_{ca} is a wrench applied at the center of the forcer, and d_a is the distance from the center of the motor to the center of the forcer, as shown in Figure 1.8.

[1] Product of MathWorks, Inc.

[2] Actually, some of the experiments in this chapter were performed before the autonomous calibration was developed so that the nominal force model parameters were used.

The actuators will be limited in the forces that they can generate. The exact nature of the limitation depends on the system design. A well-designed flux-steering motor will be designed such that the magnetic flux of the coil is enough to cancel the permanent magnetic flux when the rated input current is flowing through the coil. There will also be significant magnetic saturation of the forcer and/or platen teeth at this current level. This current level also will generate a significant but manageable temperature rise due to resistive losses in the coils. For these reasons, increasing the current above this design point has very limited utility. Note that this situation is different than other actuators that have current limits based primarily on thermal limitations, in which case it is possible to momentarily generate forces larger than the steady-state force output rating. For simplicity, the actuators of each particular forcer are assumed to all have the same force output limits chosen somewhat arbitrarily to roughly correspond to the results of the autonomous calibration or the manufacturer's ratings.

The actuator constraints will be of the form

$$f_i^- \leq f_i \leq f_i^+, \ i \in \{1...m\}, \tag{5.9}$$

with $f_i^- < 0$, and $f_i^+ > 0$. For a controller that operates in the planar workspace, these constraints on the actuator outputs must be mapped into wrench output space of the robot. A procedure for determining the domain, $\mathcal{F}$, of achievable output wrenches is presented in Appendix A. The result of this procedure is a norm-like function

$$\Phi(w) \ = \ \max\left(p_{\mathcal{V}_1}^T w, p_{\mathcal{V}_2}^T w, \ldots, p_{\mathcal{V}_k}^T w\right), \tag{5.10}$$

such that the domain is given by

$$\mathcal{F} \ := \ \{w \in I\!R^n \mid \Phi(w) \leq 1\}. \tag{5.11}$$

If the actuators all have identical output constraints, as is assumed here, the surface defined by $\Phi(w) = 1$, the *wrench envelope* $\mathcal{E}$, can be represented in $I\!R^3$ as a rhombic dodecahedron, as depicted in Figure 5.2. For example, at point p_1, the forcer is generating the maximum possible force in both the $+x$ and $-y$ directions and cannot generate any torque. Alternatively, at point p_2, the forcer is using its full capabilities to generate torque, and cannot generate any forces.

The input to the force resolution function is the desired wrench $u_{w,ca}$, which may be outside, inside or on $\mathcal{E}$ leading to over-, under-, and uniquely-constrained cases.

In the under-constrained case, an infinite number of solutions for f can be found that will generate the desired wrench. Here, we choose the solution:

$$\begin{bmatrix} f_{x1} \\ f_{x2} \\ f_{y1} \\ f_{y2} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 & -\frac{a}{a+b}\frac{1}{2d_a} \\ \frac{1}{2} & 0 & \frac{a}{a+b}\frac{1}{2d_a} \\ 0 & \frac{1}{2} & -\frac{b}{a+b}\frac{1}{2d_a} \\ 0 & \frac{1}{2} & \frac{b}{a+b}\frac{1}{2d_a} \end{bmatrix} u_{w,ca}\,, \tag{5.12}$$

where $u_{w,ca} = [f_x^d \;\; f_y^d \;\; \tau^d]$, $a = 2f_{max} - |f_x^d|$, and $b = 2f_{max} - |f_y^d|$. Note that a and b are a measure of the remaining force capability of the forcer in the x and y directions. Note that a or b are both non-negative, and because the wrench is inside the force envelope, at least one is non-zero. Therefore, (5.12) will always be defined, and (by examining the derivative) can also be shown to be continuous. Continuity is important to avoid the chattering that may result if similar output wrenches result in dramatically different actuators forces, which is possible given the actuator redundancy. The particular solution (5.12) was chosen primarily with an eye towards making it compatible with the result for the single-solution case described below so that there isn't a discontinuity between the underconstrained and single-solution cases.

If the desired wrench lies on envelope $\mathcal{E}$, there is a single solution. This solution is identical to the under-constrained case except that it becomes undefined when $a = b = 0$. In this special case the desired wrench is at one of the corners in the f_x, f_y plane (*e.g.* p_1) in Figure 5.2, so that $\tau = 0$, and (5.12) is replaced by:

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} = \begin{bmatrix} 1/2 & 0 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 1/2 & 0 \end{bmatrix} u_{w,ca}. \tag{5.13}$$

In the over-constrained case, $u_{w,ca}$ lies outside the wrench envelope. In this case, the forcer is saturated, and there are no solutions to (5.8) that satisfy the constraints. However, if we redefine the problem for this case to be mapping $u_{w,ca}$ back onto the

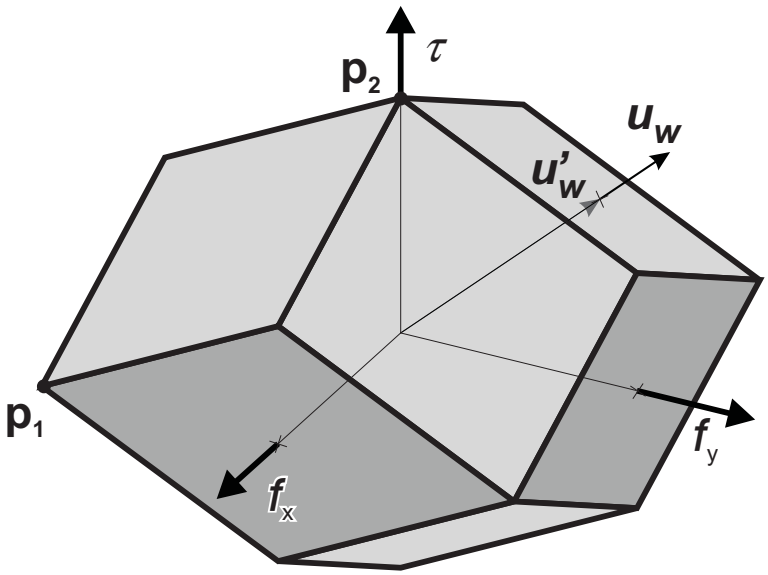


Figure 5.2: A wrench envelope represents the force/torque limits of the forcer.

wrench envelope, there are once again infinite solutions. One simple solution is to linearly scale the desired wrench vector back to the point where it pierces the wrench envelope. For example, $u_{w,ca}$ in Figure 5.2 is mapped to $u'_{w,ca}$. They are related by a scale factor s_l:

$$s_l = \max\left(\frac{|f_x|}{2f_{max}}, \frac{|f_y|}{2f_{max}}, \frac{|f_x| + |f_y| + |\tau|/d_a}{4f_{max}}\right), \tag{5.14}$$

such that $u'_{w,ca} = u_{w,ca}/s_l$.

Although this solution has been implemented, the experiments that use this controller are designed to avoid saturation cases. Saturation in general is a difficult non-linearity to deal with effectively. In this case it also acts to couple the axes, requiring a more complicated controller design if it is to be treated effectively. The controller discussed in Section 5.2 has a built-in mechanism for avoiding saturation cases.

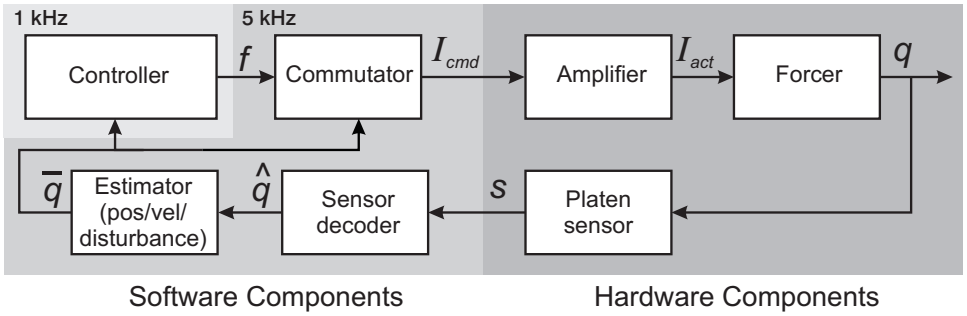


Figure 5.3: Restricted domain controller block diagram

5.1.5 Control

Given the commutator and force resolution functions, the controller can be designed around a simple linear model. The mechanical dynamics of the forcer are simply those of a mass moving in the plane with viscous damping:

$$u_{w,cm} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I_z \end{bmatrix} \ddot{q} + \begin{bmatrix} b_v(\dot{q}_x) & 0 & 0 \\ 0 & b_v(\dot{q}_y) & 0 \\ 0 & 0 & b_\omega(\dot{q}_\theta) \end{bmatrix}, \tag{5.15}$$

where m is the forcer mass, I_z is the rotational inertia, and $u_{w,cm}$ is the net wrench acting on the system. The b_v and b_ω functions model the damping and rolloff, but can be neglected for simplicity. To express the dynamics in this decoupled form, I_z and $u_{w,cm}$ are defined relative to the center of mass of the forcer. The gains of a PID control law,

$$u_w = -K\left(e + T_d\dot{e} + \int \frac{1}{T_i}e\right) + M\ddot{q}^*, \tag{5.16}$$

can then be chosen independently for each axis. The feedforward acceleration term, $M\ddot{q}^*$ is added to the controller forces if a trajectory acceleration command $\ddot{q}^*$ is available.

5.2 Restricted domain PD controller

The block diagram for a modified PD controller is shown in Figure 5.3. This diagram differs from that of Figure 5.1 in that the internals of the controller block are modified as described in the following section, and the estimator is augmented to provide an estimate of the disturbance force acting on the forcer.

5.2.1 PD domain formulation

To improve robustness in conditions where saturation may occur with the controller of the previous section, the idea of a *domain* for a controller is introduced. The controller will only execute when the system is inside an *invariant set* that includes the goal. Invariance here means that the system under the influence of the controller will never leave the set.

If a controller is not always allowed to execute, other controllers with their own domains must be provided such that the union of the controller domains covers all possible system states. At a minimum, there should be an *apotropaic* controller that has a domain covering the entire state space of the robot, encoding an appropriate fail-safe behavior for the system. The candidate controllers are ordered so that even if they have overlapping domains, only one controller will be active for any state of the system.

The controller has the form

$$u_w = -K(e + T_d\dot{e}) + M\ddot{q}^* + \bar{w}_e, \tag{5.17}$$

which is the same as controller (5.16) except that a disturbance estimate $\bar{w}_e$, described in the following section, replaces the integral term. Appendix B presents a method for deriving a large invariant domain for this controller under the assumption that the feedforward terms are bounded

$$D_i^- < M\ddot{q}_i^* + \bar{w}_{e,i} < D_i^+. \tag{5.18}$$

To see if the system is in the domain, the computation

$$\eta = \frac{1}{2}E^T K E, \tag{5.19}$$

is performed, where $E := [e \;\; \dot{e}]^T$. Note that testing if the system is in this domain requires very few operations so that many controller domains of this type can be evaluated at every controller cycle without affecting the real-time performance of the system.

5.2.2 Disturbance estimator

If it is necessary to compensate for disturbance forces, a complication arises. If the individual candidate controllers use integral terms, it is desirable to maintain the

state of the integral across controllers to eliminate integral resets when switching between controllers. For this reason, the estimator of the previous section is augmented to provide a disturbance estimate that will converge to a constant disturbance force.

The estimator exploits the linear dynamic model of the forcer to produce a filtered position and velocity signal without excessive lag. The estimator of the previous section (5.6) is replaced by

$$\bar{Q}_{cm}(k+1) = \begin{bmatrix} \Phi & 0 & 0 \\ 0 & \Phi & 0 \\ 0 & 0 & \Phi \end{bmatrix} \bar{Q}_{cm}(k) + \begin{bmatrix} L & 0 & 0 \\ 0 & L & 0 \\ 0 & 0 & L \end{bmatrix} \tilde{q}_{cm}(k) + \begin{bmatrix} \frac{T^2}{2m} & 0 & 0 \\ \frac{T}{m} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{T^2}{2m} & 0 \\ 0 & \frac{T}{m} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{T^2}{2I_z} \\ 0 & 0 & \frac{T}{I_z} \\ 0 & 0 & 0 \end{bmatrix} u_{w,cm} \tag{5.20}$$

where the following variables are redefined

$$\Phi = \begin{bmatrix} 1 & T & \frac{T^2}{2m} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}, \quad L = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix}, \text{and}$$

$$\tilde{q}_{cm}(k) = \hat{q}_{cm}(k) - \bar{q}_{cm}(k).$$

where the state estimate at time k, $\bar{Q}_{cm}(k) = [\bar{x}\ \dot{\bar{x}}\ \bar{w}_{e,x}\ \bar{y}\ \dot{\bar{y}}\ \bar{w}_{e,y}\ \bar{\theta}\ \dot{\bar{\theta}}\ \bar{w}_{e,\theta}]^T$, is augmented with the disturbance estimate $\bar{w}_e := [\bar{w}_{e,x}\ \ \bar{w}_{e,y}\ \ \bar{w}_{e,\theta}]$ of a constant external wrench acting on the system.

As before, the estimator is decoupled and the estimator gains l_1, l_2, and l_3 can be computed based on the desired estimator pole locations.

5.3 Experimental results

Sections 5.3.1, 5.3.2, and 5.3.3 describe early experiments using the saturated PD controller of Section 5.1. The software blocks in Figure 5.1 were implemented using the *normag1* system, as described in detail in Section 1.4.2. Briefly, the LynxOS real-time operating system running on a Motorola PowerPC 133 MHz computer was used to implement the controller and was interfaced to a Zygo 2-axis laser interferometer, capable of measuring the differential skew angle and one translational axis of the forcer, providing a position measurement independent of the magnetic platen sensor.

The software is structured with a single high-priority thread running at 3500 Hz that includes the commutation, sensor I/O, force resolution, controller, observer, and trajectory generator functions. This set of functions takes approximately 150 μs to complete, leaving adequate time for lower-priority user interface and network communications threads to execute. The fast update rate is necessary to provide a reasonable number of updates per pitch. With a forcer pitch of 1.016 mm, and a peak speed above 1 m/s, even with 3500 Hz updates there will be less than four updates per pitch.

Section 5.3.4 describe more recent experiments using the domain-based PD controller of Section 5.2. These experiments used the *vole* system. The controller software was structured to separate the control thread from the commutator thread. The control thread executed at 1000 Hz, while the commutator executed at 5000 Hz.

5.3.1 Resolution tests:

To characterize the position resolution under closed-loop control, the PID controller was used to regulate the forcer to zero position and angle. The gains of the controller were $K = 220$ N/mm, $T_d = 0.0053$ s, and $T_i = 0.028$ s. The controller poles were underdamped with natural frequency of 40 Hz. Estimator poles were placed at 80 Hz, which was near the upper limit for the controller rate of 3500 Hz. Above this rate, an unacceptable level of noise from the sensor was passed into the controller, causing the forcer to vibrate *audibly*.

Readings for the x translation and θ skew angle from the platen sensor, estimator, and laser interferometer were recorded at 3500 Hz for 1000 samples. The amount of

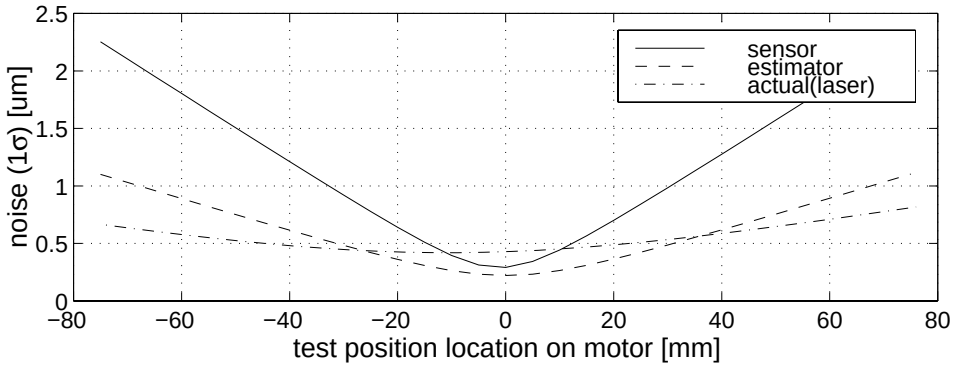


Figure 5.4: The controller regulates points near the sensor better than points on the periphery of the forcer.

motion in the x direction of a particular point on the forcer was computed using the simple differential kinematic equation:

$$\delta x_t = \delta x - \delta\theta \; y_t, \tag{5.21}$$

where δx_t is the x differential motion of a test point (x_t, y_t) on the forcer given a differential motion at the middle of the forcer of δx and $\delta\theta$. Figure 5.4 shows the standard deviation of δx_t as recorded by the sensor, estimator, and laser interferometer as y_t is varied. Note that points with $|y_t| = 75$ mm correspond to the edge of the forcer. Because the sensor measures angle by differencing two parallel position measurements that are close together (see [22]), there is a low-noise *sweet spot* in the middle of the forcer, where the sensor is located. However, Figure 5.4 indicates that even at the edge of the forcer, the controller maintains micron-level resolution (1σ), which is sufficient for many applications.

5.3.2 Repeatability

Repeatability is the ability of an actuator to return to the same position. The difference in the forcer positions after moving to a reference location from two different directions was measured with the laser interferometer. This process was repeated with varying move distances. The controller was started several minutes before testing began, and the test was designed to be completed in under a minute to minimize

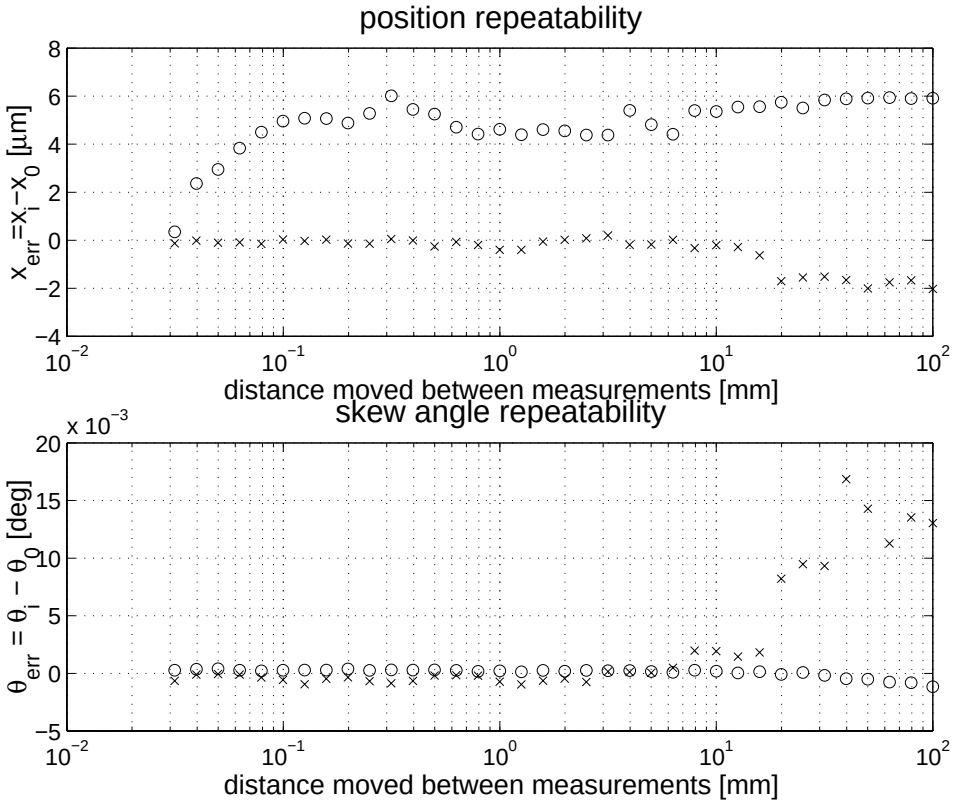


Figure 5.5: Bi-directional repeatability: errors in moving to a reference position from two approach directions are shown for open-loop (circles) and PID control (crosses) for a range of motion distances.

thermal effects. The crosses in Figure 5.5 show this bi-directional position repeatability for the PID controller to be under 2.5 μm peak-to-peak, and the skew angle repeatability to be under 0.02° peak-to-peak over 36 motions. Note that the error increases at a travel distance of about 20 mm, which is when the linear motors start to overlap the reference position, so it appears likely that the motors are leaving a residual magnetic field in the platen that is slightly affecting the sensor operation. The exact mechanism for this interaction has not been investigated further. For comparison, the bi-directional repeatability tests are repeated under open-loop control. The error here is probably due to a combination of tether disturbance forces and the same magnetic hysteresis that affects the sensor. Note that these effects can be avoided in practice by approaching a goal position from the same direction.

5.3.3 Trajectory commands

The closed-loop controller was used to track a trajectory with a bang-bang acceleration of 10 m/s^2, maximum velocity of 0.8 m/s, and a position change of 0.1 m. Integral gains were disabled for this experiment to prevent integral windup during the motion.

As shown in the dark traces in Figure 5.6, the PD controller tracks the trajectory to within 50 μm, and settles to 1 μm within 20 ms. The tracking error comes from motor modeling errors (i.e. eddy-current damping, errors in the mass or peak force model parameters, or unmodeled actuator nonlinearities.). The tracking error is also shown when a mass of 240 g is attached to the edge of the forcer, causing errors in in the mass, inertia, and center of mass of the model. Even in this case, the controller settles just as well, although the tracking error increases significantly during one part of the trajectory.

For comparison, the tracking experiment was repeated under open-loop control. However, it would be unfair to have the open-loop controller attempt to track the same bang-bang acceleration trajectories. Instead, a trajectory suggested by Nordquist and Smit [31] was implemented. This trajectory is identical to the bang-bang acceleration trajectory, except for *burst* (step changes applied to the desired position) and *acceleration rolloff* as the velocity increases. These changes are designed so that the open-loop forcer dynamics will not be excited and, in the ideal

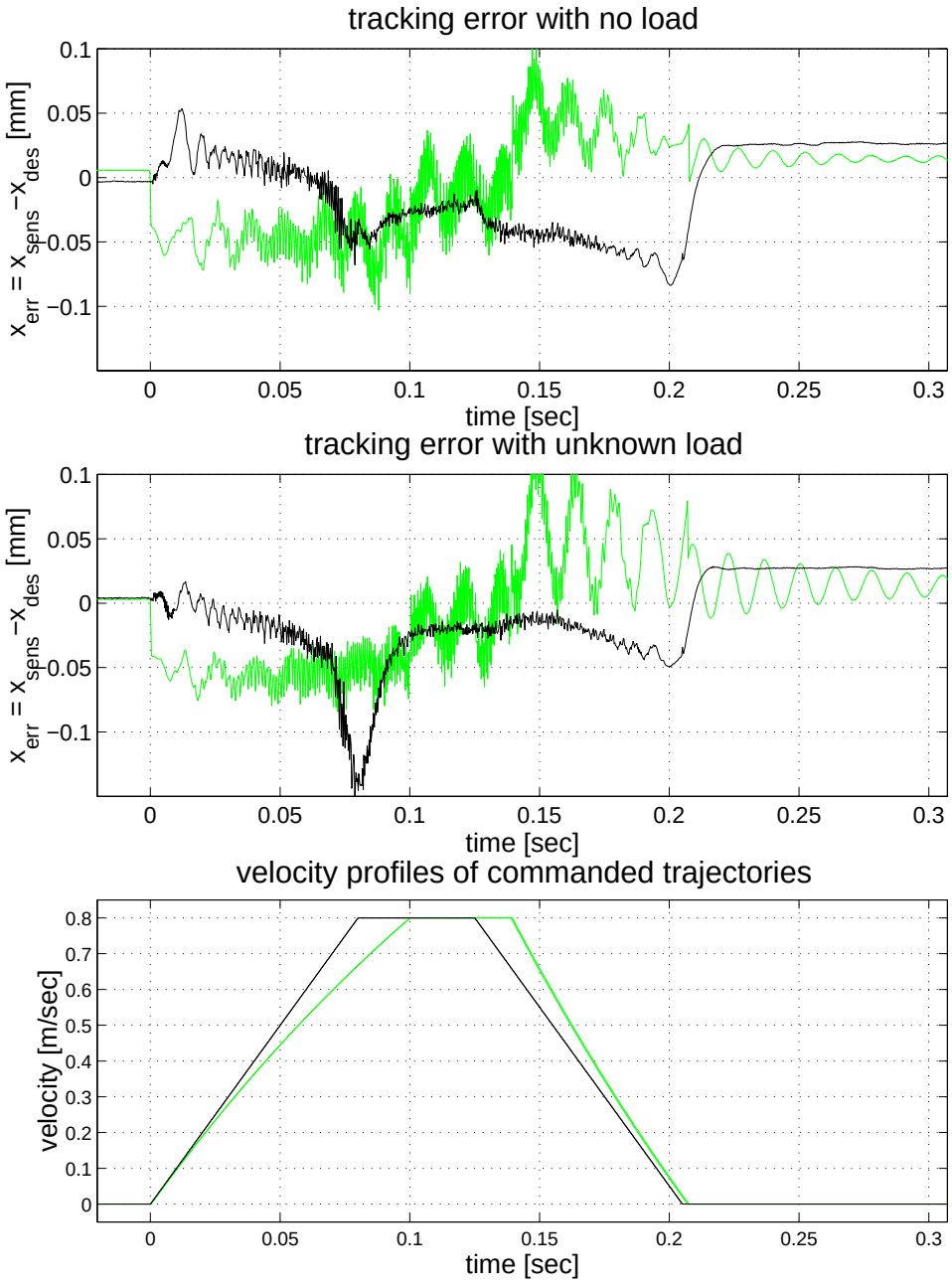


Figure 5.6: Trajectory tracking: tracking errors are shown for PD (dark trace) and open-loop (light trace) control given no load (upper plot) and an "unknown" load of 240 g (middle plot). The lower plot shows the velocity profile of the trajectories.

case, there will be a constant tracking error during each phase (acceleration, slew, and deceleration) of the trajectory.

After a reasonable amount of tuning of the trajectory, tracking errors were as shown in the light traces of the top plot of Figure 5.6. Although the tracking error levels are reasonably good, there is a significant oscillation, indicating that some dynamic parameters had errors or there were unmodeled nonlinearities. This oscillation grew worse when the extra load was added, as shown in the middle plot. Furthermore, in both cases, the open-loop controller takes much longer to settle at the end of the trajectory.

5.3.4 Power consumption and thermal effects

The use of closed-loop operation with a fixed phase commutator (see Chapter 2) only requires currents to be in the motor coils during motions and to reject disturbances. In contrast, open-loop operation requires currents to be in the coils at all times. Otherwise, any disturbance will cause the forcer to drift from its position, which cannot be detected without added sensing.

It has been shown by others [75] that heating from open-loop operation can cause forcer temperatures as high as 47°C (measured on the center of the inside of the forcer housing), platen temperatures above 40°C, and local thermal deformations of the platen as high as 60 μm. These tests involved very little motion of the robot, and reflects heating that occurs from resistive losses when it is stationary for tens of minutes.

For comparison, closed-loop operation of the *vole* system was used to regulate the forcer at a fixed location, with the tether in a natural position, which imposes a small disturbance force on the device. The sensor electronics makes the inside of the forcer inaccessible, so the center of one of the outside walls of the forcer body was used as a temperature test point with a Tektronix P6602 temperature probe. A PD controller was used to servo the forcer to a fixed position with gains (5.16) $K_{1,1} = K_{2,2} =$ 160000 N/m, $K_{3,3} = 250$ Nm/rad, and $T_{d1,1} = T_{d2,2} = T_{d3,3} = .011$ s, corresponding to natural frequencies of 50 Hz on the translational axes and 35 Hz on the rotational axis with all axes critically damped. The temperature probe read a value of 25.2°C for room temperature and only 26.5°C for the forcer body temperature after over 30 minutes. While these temperature numbers are not directly comparable with those

from prior work [75], it is clear that under closed-loop operation, thermal effects from resistive losses are negligible.

In an efficiently-designed system, one would expect forcers to spend much of their time moving. During motions, eddy currents are formed when the large magnetic fields of the forcer sweep across the platen surface. These electrical currents are mostly a concern in the large continuous top surface of the platen, because the forcer pole pieces are formed from laminations that help to minimize eddy current effects. Dynamic temperature measurements are difficult, but it is relatively easy to monitor the power usage of the amplifiers. This power gives an indication of the amount of energy dissipated in the forcer and platen as eddy current heating, although it also includes heating of the amplifiers themselves. The power measurement also allows a comparison of the relative efficiency of open-loop and closed-loop operation of planar motors, which is especially important for enabling tetherless operation of these devices.

To this end, the forcer was commanded to move using a time optimal trajectory with a distance of 0.4 m, peak velocity of 1.0 m/s, and acceleration of 10 m/s^2 for both open-loop and closed-loop operation (using the same gains as in the above temperature test). A Tektronix A6303 non-contact current probe was used to measure the current draw of the Copley 4212 amplifiers during the motion, with the current probe output sampled using an extra input channel of the A2D converter for the coordination sensor in the *vole* system. The power usage of the amplifiers in Watts is computed by taking the product of the current measurement and the 120 V supply voltage of the amplifiers. Open-loop operation results are shown in Figure 5.7 and closed-loop operation results are shown in Figure 5.8. The raw power measurements sampled at 1000 Hz are shown with digitally filtered versions that use a low-pass filter with two poles at 100 Hz to help eliminate some of the noise from the amplifier switching. The sensed velocity profiles are also shown for reference.

For open-loop operation (Figure 5.7), the power usage rises with velocity, peaking at 600 W for velocities of 1 m/s. For comparison, the mechanical power usage of the forcer generating its nominal maximum force of 50 N at a velocity of 1 m/s will require only 50 W of power. Most of the power usage must be going into thermal heating of the platens, forcers, or amplifiers. Note also that the steady-state power usage of open-loop operation is 250 W. For comparison, the expected thermal dissipation of the motor coils can be computed. The resistance of the 8

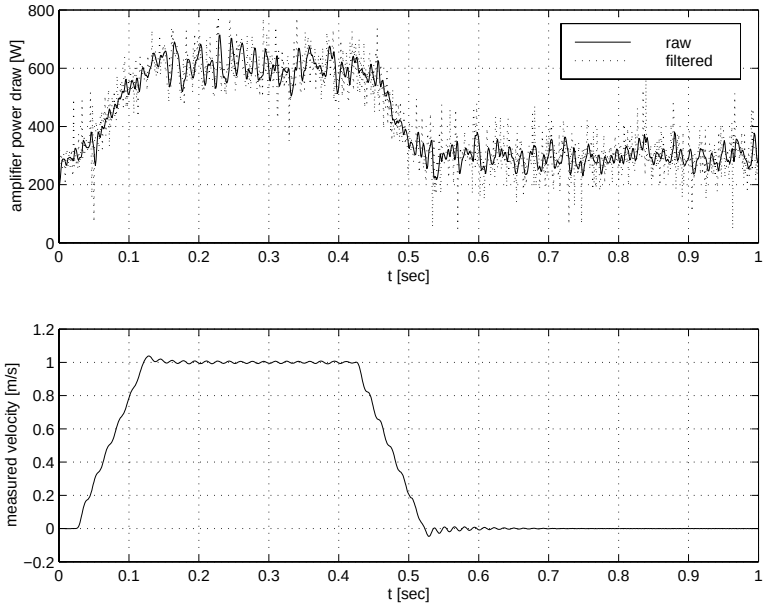


Figure 5.7: Amplifier power usage and forcer velocity for open-loop trajectory tracking.

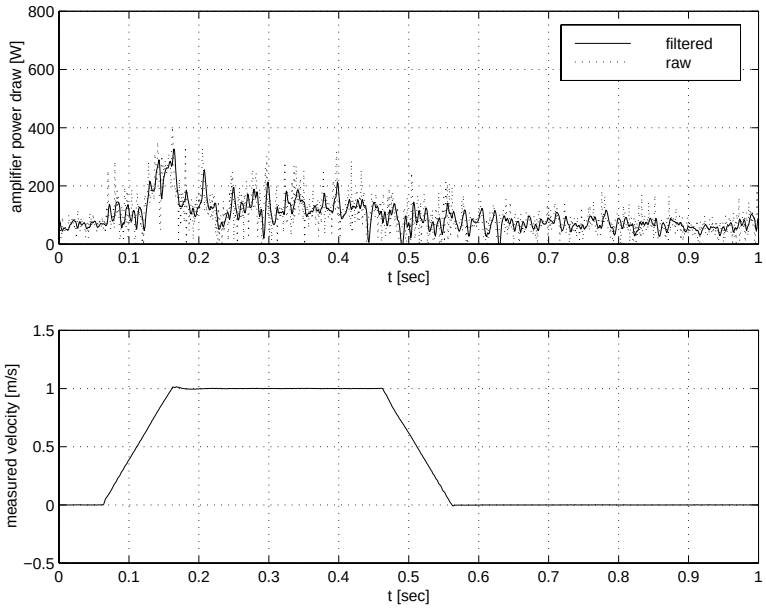


Figure 5.8: Amplifier power usage and forcer velocity for closed-loop trajectory tracking.

actuator coils is roughly 1 Ω, and the nominal current in the coils is 3 A, giving a power dissipation of 9 W per coil or 72 W total. In this case there is no significant motion of the forcer, so that the extra power must be dissipated in the amplifier or in eddy currents induced by high-frequency components of the PWM amplifier outputs.

In contrast, the closed-loop operation power requirements shown in Figure 5.8 are much lower. Less than 70 W are used in the steady-state condition. For comparison, the Copley amplifiers require 65 W even when their outputs are disabled. The peak power usage during the motion is 400 W, which is not much larger than the steady-state power usage of open-loop operation. In addition, during the slew and deceleration portion of the trajectory, only about 100 W more than the steady-state operation appear to be required.

The reduced power usage of closed-loop operation allows for more compact amplifier packaging and reduced thermal heating and thermal expansion of the forcer and platen. Reduced energy consumption also partially enables the design of tetherless planar robots. Note that the energy used by each mode of operation is the area under the power curves of Figure 5.7 and Figure 5.8. Numerically integrating the filtered power curves over the 1 s of data yields an energy usage of 420 J for open-loop operation and 95 J for closed-loop operation, demonstrating the dramatic reduction in energy usage possible with closed-loop operation.

While the performance gains shown in this chapter are important to consider, the main advantage and motivation for closed-loop operation is in enabling new modes of behavior for the planar robot. Instead of carefully scripted motions in highly structured environments, closed-loop operation allows the robot to assuredly explore and interact with unknown or uncertain environments without the danger of losing track of its position.

Chapter 6

Hybrid control

> It is most gratifying when a goal is
> achieved through one's efforts.
> Lucky Numbers 4, 6, 9, 10, 27, 29

Robot motion planning and control have traditionally been treated independently, with the controller having no knowledge of workspace boundaries and obstacles, and the planner having no knowledge of the details of control. Although this approach can often lead to simple solutions with acceptable performance, if the planning and control problems are coupled such that the plans would benefit greatly from information available only at runtime, this separation will yield substandard results.

The controllers of the previous chapter require trajectories to move large workspace distances if actuator saturation is to be avoided. This approach treats the trajectory planning problem separately from the controller design problem. However, these problems are interrelated. The controller domain computation required the trajectory's desired accelerations to be bounded, as these bounds are used to compute a predicate for the controller that will ensure stability. A disadvantage of this approach is that the time-slaved trajectory cannot be modified once it has begun. While the controller may dynamically respond to disturbances, if it falls significantly behind the trajectory command it may not be able to catch up again without leaving the workspace path implied by the trajectory. This chapter explores how motion planning can be included in the controller. Although this work has not advanced beyond successful simulations (further work is required to properly handle the limited-range rotation axis), it does demonstrate one method for combining planning and control for solving the planar robot control problem.

Recently, such *hybrid* control techniques have been used to move higher-level decisions into the controller in a provably stable manner. Time-varying system dynamics [116], state and control constraints [117, 100], disturbances [118], and task-level programming of dynamic tasks [119] have been incorporated into controllers.

The main complication for motion planning of planar robots arises when multiple robots share a single platen. In this case, a collision avoidance scheme [66] is necessary. The task considered in this chapter is the transportation of products between overhead devices. The traditional approach would be to consider this as a planning problem and choose a collision-free set of trajectories for the robots. However, disturbances may make it impossible to track an aggressive time-parameterized trajectory, resulting in controller saturation and overshoot which could lead to collisions. These problems arise because an exact spatial and temporal path for collision-free motion is an overspecific solution since generally any efficient collision-free motion would be acceptable.

An alternative, conceptually simple approach for collision avoidance is to predefine *reservation areas* on the platen surface, where a robot would only move into a new region if it can obtain exclusive access by negotiating with its peers. If a controller can then be designed that will provide efficient motion towards a goal state in a region while guaranteeing that the robot will not leave the region, explicit trajectories can be avoided, yielding feedback-based motion that will robustly respond to a large variety of disturbances. The bulk of the work of this chapter concerns the development of such a controller, restricted to a convex domain. While convex domains are not sufficient for complex reservation area layouts, multiple controllers can be used, as detailed more formally elsewhere [100], to form polygonal domains by overlapping them such that the goal of one controller lies in the domain of the next controller, eventually leading to the goal of the overall polygonal domain.

The planar robot control problem is now restated for this task. Assuming an unoccupied convex region on the platen, find a controller that takes the robot from any starting position within the region and a large selection of velocities to the goal position of that region without leaving the region or violating the velocity, acceleration, or actuator limits. The controller must regulate the robot's rotation to prevent reductions in the force output. The limited actuator capabilities must be carefully shared between these two tasks.

6.1 Controller

For simplicity, the planar robot dynamics are modeled as

$$\ddot{q} = u, \tag{6.1}$$

with $u, q \in I\!R^3$. The actuator and velocity limits are

$$\Upsilon(u) \leq 1 \tag{6.2}$$

$$\|\dot{q}\|_V \leq 1, \tag{6.3}$$

where $\|q\|_V := (q^T V q)^{1/2}$. The function $\Upsilon(u)$ is a norm-like function with the property that $u/\Upsilon(u)$ gives the largest achievable acceleration in the direction of u, computed based on the equivalent norm-like function for forces (A.12),

$$\Upsilon(u) := \Phi(Mu), \tag{6.4}$$

where M is the mass matrix of the robot.

A controller will also have an associated convex domain that restricts the robot position

$$q(t) \in \mathcal{P} \ \forall t \geq t_0 \tag{6.5}$$

$$\mathcal{P} := \{q \in I\!R^3 \mid \beta_i(q) \geq 0\} \ \forall i \in \{1...N\} \tag{6.6}$$

$$\beta_i := l_i^T q - c_i, \tag{6.7}$$

where the l_i and c_i parameters encode the direction and distance to the half-planes that make up the boundary of the domain. They are defined such that $\|l_i\| = 1 \ \forall i \in \{1...N\}$, $l_i^T l_j \neq 1 \quad \forall i \neq j \in \{1...N\}$, and $N \geq 4$. Note that $\mathcal{P}$ includes both translation and angle constraints.

For reasons described below, the boundary directions l_i and velocity norm $\|\cdot\|_V$ must be related by

$$(D_q \|q\|_V)^T l_i = 0 \quad \forall q \in I\!R^3 \, | q^T l_i = 0, \tag{6.8}$$

where D_q is the partial derivative operator relative to q. Examples that satisfy (6.8) include a standard 2-norm with any boundaries, and a scaled 2-norm with its principle axes parallel to the boundaries.

The strategy for the controller design is to start with a distance function γ such as

$$\gamma(q) := \sum_{i \in 1...3} \frac{\|q_i - q_i^*\|^2}{\|q_i - q_i^*\| + \alpha_i}, \tag{6.9}$$

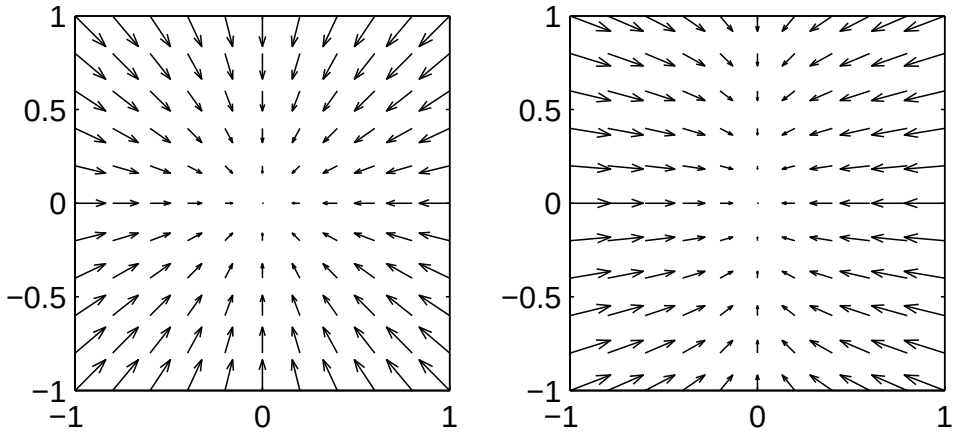


Figure 6.1: A vector field $\hat{v}$ derived from (6.9) defines a desired velocity at each position for the controller. (left: $\alpha_1 = \alpha_2 = 5$, right: $\alpha_1 = 2, \alpha_2 = 8$)

where q^* is the desired rest position for the system. Other choices for γ are possible, provided the function is positive definite about q^*, has a bounded gradient, and a positive definite Hessian, $D_q^2\gamma$. The transposed Jacobian (gradient) of γ, $D_q\gamma^T$, can then be used as a position dependent velocity command

$$\hat{v}(q) = -V^+ D\gamma^T(q), \tag{6.10}$$

where the symmetric, positive definite matrix V^+ is used to scale $D\gamma$ such that $\|\hat{v}(q)\|_V < 1 \ \forall q \in \mathcal{P}$. Figure 6.1 shows an example 2-D vector field, $\hat{v}(q)$, for two different choices of the α parameters.

Three controllers are combined such that $(q^*, 0)$ is reached without violating constraints (6.2)-(6.5). A *velocity* controller regulates the velocity to that given by $\hat{v}(q)$, but requires that the velocity already points roughly in the correct direction. This controller is sufficient if the overall controller is always activated with $\dot{q} = 0$, but it is desirable for the controller to activate at non-zero velocities for more efficient transitions between sequences of convex controllers.

Thus, an additional controller is necessary. Ideally, this controller would start from any $q \in \mathcal{P}$ and any velocity and bring the system into the domain of the velocity controller. However, there will be starting conditions that will inevitably lead to boundary violations. It is necessary to define the *savable set*, $\mathcal{S}$, as those $(q, \dot{q})$ for which a controller exists that brings $\dot{q}$ to zero without violating constraints (6.2)-(6.5). It is difficult to design a single controller that has a domain of $\mathcal{S}$, as it

requires careful consideration of the boundaries, while also efficiently bringing $\dot{q}$ to $\hat{v}(q)$. However, a *stop* controller designed to bring $\dot{q}$ to zero for any starting condition in $\mathcal{S}$ can be combined with a *join* controller that seeks to take over from the stop controller as soon as possible and bring $\dot{q}$ into the domain of the velocity controller.

The domain of the stop controller is the entire savable set $\mathcal{S}$, which in turn contains the domain of the join controller, which contains the domain of the velocity controller. The sequencing of these controllers is designed to be monotonic–the stop controller leads to the join controller leads to the velocity controller, with no "backwards" transitions. The remainder of this section presents these controllers more formally.

The use of such a control strategy was proposed previously [100], but as detailed below, the join controller has been modified to have a much larger domain, allowing it to take over from the stop controller much sooner. The implications of the more complicated actuator bounds of the planar robots are also addressed.

6.1.1 Stop controller (Φ_S)

Unfortunately, the form of the acceleration constraint $\Upsilon(u)$ complicates the formulation and demonstration of a controller with a domain of $\mathcal{S}$. The arguments in prior work [100] rely on having 2-norm acceleration limits. While these arguments have been extended to include scaled 2-norms, their extension to a general convex limit such as $\Upsilon(u)$ remains in progress. Note that the stop controller's primary purpose is to extend the domain of the overall system to $\mathcal{S}$ to improve transitioning performance between convex controllers in certain cases. However, for the planar robot control problem, this mode of operation is not typically excited and the stop controller can be omitted.

6.1.2 Velocity Regulation (Φ_V)

The velocity field given by $\hat{v}(q)$ is used as a position dependent velocity command for (6.1) in the controller

$$u = -k\sigma(q,\dot{q})(\dot{q} - \hat{v}(q)) + \zeta(q,\dot{q})D\hat{v}(q)\dot{q}, \tag{6.11}$$

where k is a positive scalar gain for the "velocity regulation" term, which is scaled by $\sigma(q,\dot{q})$, as defined below. The second term acts as a feed-forward acceleration to

track the desired velocity surface $\mathcal{V} := \{(q,\dot{q}) \in T\mathcal{P} | \dot{q} - \hat{v}(q) = 0\}$, where $T\mathcal{P}$ is the tangent bundle of $\mathcal{P}$.

The domain of the controller consists of the union of two sets, the first a neighborhood of $\mathcal{V}$,

$$\mathcal{N}_v := \left\{ (q,\dot{q}) \in T\mathcal{P} \left| \frac{1}{2}(\dot{q}-\hat{v})^T(\dot{q}-\hat{v}) < \epsilon_v \right. \right\}, \tag{6.12}$$

and the second the set

$$\mathcal{N}_\omega := \{(q,\dot{q}) \in T\mathcal{P} | 0 \leq \zeta \leq 1, \|\omega\| < \epsilon_\omega\}, \tag{6.13}$$

where $\zeta := \frac{\hat{v}^T\dot{q}}{\hat{v}^T\hat{v}}$ and $\omega := (I - \frac{\hat{v}\hat{v}^T}{\hat{v}^T\hat{v}})\dot{q}$ are used to decompose $\dot{q}$ as $\zeta\hat{v} + \omega$. Note that $\mathcal{N}_\omega$ is a neighborhood of the states where $\dot{q}$ is "aligned with" and "smaller than" $\hat{v}$. The parameter ϵ_ω is chosen such that $\Upsilon(\zeta D\hat{v}\dot{q}) < 1$ for $(q,\dot{q}) \in \mathcal{N}_\omega$. The scale parameter σ is set to 1 for $(q,\dot{q}) \in \mathcal{N}_v$; otherwise it is dynamically chosen to scale the feedback term of (6.11) by the range $0 < \sigma \leq 1$ so that $\Upsilon(u) = 1$. Parameter ϵ_v is chosen such that $\Upsilon(u) \leq 1\ \forall (q,\dot{q}) \in \mathcal{N}_v$. It is then straightforward to show[1] that (6.2) is satisfied over the controller domain $\mathcal{N}_v \cup \mathcal{N}_\omega$.

Stability of this controller in the absence of any external constraints can be seen by considering

$$\eta_\gamma = \gamma(V^+kq) + \frac{1}{2}\dot{q}^T\dot{q} \tag{6.14}$$

as a candidate Lyapunov function for (6.1) under the influence of (6.11) about q^*. Taking the time derivative of (6.14) along the trajectory of the system yields

$$\dot{\eta}_\gamma \;\; = \;\; D\gamma V^+ k\dot{q} + \dot{q}^T u, \tag{6.15}$$

and substituting (6.10) and (6.11) gives

$$\dot{\eta}_\gamma = -k(1-\sigma)\hat{v}^T\dot{q} - k\sigma\dot{q}^T\dot{q} + \zeta\dot{q}^T D\hat{v}\dot{q}. \tag{6.16}$$

The first term is identically zero for $(q,\dot{q}) \in \mathcal{N}_v$ (because $\sigma = 1$) and negative semidefinite in the state for $(q,\dot{q}) \in \mathcal{N}_\omega$ (because $\hat{v}^T\dot{q} \geq 0$). The remaining terms are also negative semidefinite in the state, and we can conclude that $\lim_{t\to\infty}\dot{q} = 0$. Finally, by recourse to LaSalle's Invariance Principle [120], $\lim_{t\to\infty}\gamma(V^+kq) = 0$ and q converges to q^*.

[1] One necessary assumption is that the parameters of (6.9) and (6.10) are chosen such that $\Upsilon(u) < 1\ \forall (q,\dot{q}) \in \mathcal{V}$.

Proposition 1 *Under the influence of (6.11) the surface* $\mathcal{W}$:=$\{(q,\dot{q}) \in T\mathcal{P} | \omega = 0\}$ *is both attractive and invariant over* $\mathcal{N}_v \cup \mathcal{N}_\omega$.

This can be easily seen by considering the Lyapunov function

$$\eta_\omega = \frac{1}{2}\omega^T\omega. \tag{6.17}$$

After taking the derivative of (6.17) along the trajectories of the system under the control (6.11), it can be shown (Appendix C) that

$$\dot{\eta}_\omega \;=\; -k\sigma\omega^T\omega, \tag{6.18}$$

which allows us to conclude that ω converges to zero.□

Proposition 2 *Under the influence of (6.11) the surface* $\mathcal{D}$:=$\{(q,\dot{q}) \in T\mathcal{P} | \zeta = 1\}$ *is both attractive and invariant over* $\mathcal{W}$.

Similarly, this can be seen by considering the Lyapunov function

$$\eta_\zeta = \frac{1}{2}(\zeta - 1)^2. \tag{6.19}$$

After taking the derivative of (6.19) along the trajectories of the system under the control (6.11), it can be shown (using algebra similar to that in Appendix C) that

$$\dot{\eta}_\zeta \;=\; (\zeta - 1)\frac{\dot{q}^T D\hat{v}}{\hat{v}^T\hat{v}}\omega - (\zeta - 1)^2 k\sigma. \tag{6.20}$$

The first term is zero for $(q,\dot{q}) \in \mathcal{W}$ (because w will be zero), and the second is negative definite in $(\zeta - 1)$, so we can conclude that ζ converges to 1.□

Combining these propositions, it is easy to show that the controller (6.11) will drive the state to $\mathcal{W}$, and then to $\mathcal{W} \cap \mathcal{D} = \mathcal{V}$, and we can conclude that $\mathcal{V}$ is both attractive and invariant for the entire controller domain, $\mathcal{N}_v \cup \mathcal{N}_\omega$.

The design of this controller requires choosing the velocity limit V^+, and any parameters of the particular γ form such that constraints (6.2)-(6.5) are not violated. The acceleration constraint (6.2) was discussed earlier. The velocity constraint (6.3) will be obeyed if the time derivative of $\|\dot{q}\|_V$ decreases when $\|\dot{q}\|_V = 1$. Details are omitted here, but it can be shown that (6.3) will not be violated under this control policy.

The final constraint on position (6.5) must also be examined. As the velocity controller has no knowledge of the boundaries, the designer must ensure *a priori*

that they are obeyed. One problem occurs if the "trajectory" implied by $\hat{v}$ starting at any $(q, \dot{q}) \in \mathcal{V}$ leaves $\mathcal{P}$. This case can be avoided if the parameters are chosen such that

$$\hat{v}(q)^T l_i > 0 \ \ \forall i \in \{1...12\}, \{q|\beta_i(q) = 0\}, \tag{6.21}$$

i.e. $\hat{v}$ always points towards the interior of $\mathcal{P}$ along its boundaries. In addition, system trajectories within the domain must also not leave $\mathcal{P}$. A Lyapunov-based derivation of a suitable condition for parameters was sketched in [100], although formal demonstration of this constraint has not been developed. In practice, reasonable values for V^+ and α do not exhibit this defect.

6.1.3 Join controller (Φ_J)

The join controller has three components

$$u = \sigma_a u_a + \sigma_b u_b + \sigma_s u_s. \tag{6.22}$$

As defined below, u_a is designed to avoid boundaries while using a time-varying fraction σ_a of the total acceleration capability, u_b is designed to push $\zeta = \frac{\hat{v}^T \dot{q}}{\hat{v}^T \hat{v}}$ into the range $0 \leq \zeta \leq 1$, and u_s is designed to steer the velocity by reducing any undesirable components. As detailed below, the σ values are chosen somewhat conservatively so that (6.2) is not violated, with the *avoidance* taking priority over *braking*, which has priority over *steering*. Each of the three u's will be designed with the properties

$$u^T \dot{q} \ \leq \ 0 \tag{6.23}$$

$$u^T V \dot{q} \ \leq \ 0. \tag{6.24}$$

With these properties it is easy to show that the join controller will (in the worst case) bring $\dot{q}$ to zero, which is in the domain of the velocity controller. With property (6.24) it is easy to show that the velocity constraint (6.3) will not be violated.

Boundaries are avoided by computing the ratio δ_i of the required stopping distance to each boundary to the distance to that boundary

$$\delta_i = \frac{\max(-l_i^T \dot{q}, 0)^2 / 2u_{l,i}}{\beta_i}, \tag{6.25}$$

where $u_{l,i} = \frac{l_i}{\Upsilon(l_i)}$ is the maximum acceleration that can be generated normal to the ith boundary. The net direction of the avoidance control is given by $u_a = \sum_i \delta_i u_{l,i}$,

with magnitude $\sigma_a = \sum_i \delta_i$. It is easy to show that (6.23) is satisfied, but u_a may act to increase the scaled velocity norm $\|\dot{q}\|_V$ in certain cases, possibly leading to a violation of (6.3). However, (6.8) ensures that the velocity norm and the boundaries are "aligned" such that this violation is impossible.

The braking control is simply defined with direction

$$u_b = \begin{cases} 0 & 0 \leq \zeta \leq 1 \\ -\frac{\dot{q}}{\Upsilon(\dot{q})} & \text{otherwise,} \end{cases} \tag{6.26}$$

and magnitude

$$\sigma_b = \begin{cases} 1 - \sigma_a & u_b \neq 0 \\ 0 & u_b = 0. \end{cases} \tag{6.27}$$

The steering control uses the remaining acceleration capability, if any, to reduce velocities normal to the local desired velocity. This normal may be defined based on a Euclidean norm or a scaled norm that reflects the velocity limits

$$u_{s1} = -\left(\mathrm{I} - \frac{\hat{v}\hat{v}^T}{\|\hat{v}\|^2}\right)\dot{q} \tag{6.28}$$

$$u_{s2} = -\left(\mathrm{I} - \frac{\hat{v}\hat{v}^T V}{\|\hat{v}\|_V^2}\right)\dot{q}. \tag{6.29}$$

Note that u_{s1} will never increase $\|\dot{q}\|$ while u_{s2} will never increase $\|\dot{q}\|_V$. Also, (we claim) at any given instant, either u_{s1} will decrease $\|\dot{q}\|_V$ or u_{s2} will decrease $\|\dot{q}\|$. In this case, the steering direction can be chosen as

$$u_s = \begin{cases} u_{s1} & \dot{q}^T V u_{s1} < 0 \\ u_{s2} & \dot{q}^T V u_{s1} \geq 0 \end{cases} \tag{6.30}$$

to guarantee that the steering control will not violate the velocity constraint and not increase the Euclidean velocity norm. The steering magnitude is chosen to use any available acceleration (reduced near $\Upsilon(u_s) = 0$ to prevent chattering) as

$$\sigma_s = \begin{cases} \frac{1}{\Upsilon(u_s)}(1 - \sigma_a - \sigma_b) & \Upsilon(u_s) > \epsilon_s \\ \\ \frac{\Upsilon(u_s)}{\epsilon_s^2}(1 - \sigma_a - \sigma_b) & \Upsilon(u_s) \leq \epsilon_s. \end{cases} \tag{6.31}$$

The domain of the overall join controller is given by constraints (6.2), (6.3), and (6.5) together with one additional constraint, $\sum_i \delta_i < 1$, to guarantee that the avoidance controller will not activate if it might be overwhelmed. The goal of the

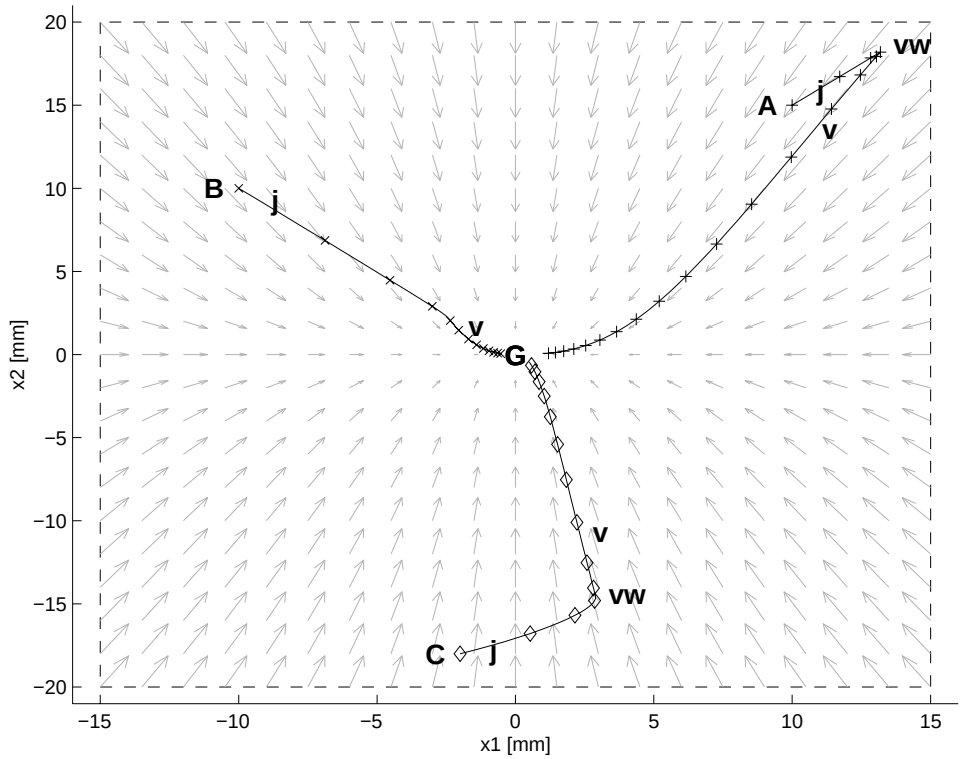


Figure 6.2: Position traces for three simulations A, B, C with different starting conditions but identical boundary $\mathcal{P}$ (dashed lines). Controller switches are indicated by codes (j=join, vw=velocity controller in $\mathcal{N}_\omega$, v=velocity controller in $\mathcal{N}_v$). Symbols ($+$, $\times$, $\diamond$) demarcate 5 msec time intervals.

join controller, expressed in terms of ζ and ω from the previous section, is the set $\{(q, \dot{q}) \in T\mathcal{P} | \omega = 0, (0 \leq \zeta \leq 1)\}$, which is a subset of the domain of the velocity controller, Φ_V.

6.2 Results

Simulations with varying starting conditions were performed to demonstrate reasonable controller behavior, at least anecdotally. All simulations use the planar robot and controller parameters given in Table 6.1.

The first simulation shows a case where the controller becomes active with the robot rapidly approaching the boundary of $\mathcal{P}$. The translational position domain is shown in Figure 6.2 along with the vector field $\hat{v}$ and the position trace (marked with $+$). The position trace begins (A) with the join controller (j) being activated. In this

	x	y	θ
k	800 s^{-1}	800 s^{-1}	800 s^{-1}
α	0.05 m	0.02 m	0.01 rad
V^+	1.0 m/s	1.0 m/s	0.5 rad/s
diag(V)	1.0 s^2/m^2	1.0 s^2/m^2	4.0 s^2/rad^2

Table 6.1: Simulated controller parameters

case, the robot is headed in the wrong direction, so the join controller brings it to a stop. Next, the velocity controller with the system state in $\mathcal{N}_\omega$ (vw) increases the velocity, bringing the system into $\mathcal{N}_v$ (v), which leads the robot to the goal (G) at the origin. Note that controller switches occur in the proper sequence. Although not shown here, the angle also starts with an initial error and a velocity of the incorrect sign. However, the controller smoothly handles both the angle and translational errors.

The second simulation shows a case where the controller becomes active with the robot headed towards the goal with a velocity larger than $\hat{v}$. The position sequence (B) in Figure 6.2 shows the join controller starting with its braking component active to reduce ζ. After sufficient braking, the system state directly enters $\mathcal{N}_v$ and the velocity controller (v) takes over and moves the robot to the origin.

The final simulation shows the robot headed roughly perpendicular to the goal with a large velocity. Position sequence (C) in Figure 6.2 shows the join starting with its braking component active for a short time followed by the steering component. In this case, a component of the initial velocity pointed towards the goal, so that complete stopping was unnecessary. In practice, we expect this case to be the most common when multiple convex region controllers activate one another in succession.

6.3 Discussion

Simulation results demonstrated good performance in a wide range of situations, simultaneously avoiding region boundaries, angle limits, and actuator limits. However, experiments using the *normag1* and *normag2* systems with this controller have uncovered some deficiencies of the controller design. The main implementation difficulty was the tuning of controller parameters of the angle axis. This axis is primarily

used in a regulation mode during robot motions, but the controller parameters are related to motion and velocity profiles that are not directly relevant. The angle inaccuracies are also a significant fraction of the rotational limits of the device, which caused the monotonicity of the controller sequencing to break. The controller would reach the goal, but the monotonicity of the controller sequences was violated due to the angular errors, resulting in some momentary chattering that slowed the progress towards the goal.

For a more successful implementation of this controller, the design of the angle axis needs to be different than that of the translational axes. Rather than parameters that are used to design workspace motions, parameters for this axis should be related to regulation performance. It is relatively straightforward to redefine the distance function γ (6.9) to have a more appropriate parameterization for the rotational axis or, more generally, any axis that will be operated in regulation mode during the active use of the controller.

While the experimental performance of these controllers was not impressive, the experiments did demonstrate, at least anecdotally, the robustness of the approach. Despite the persistent disturbance in the angle measurement, the robot did reach its goal without over-rotating or violating the other constraints on position and velocity. However, the important result from this chapter is that it is possible to build controllers for the planar robot that can respond naturally to uncertainties without intervention by higher-level processing. Instead of only knowing how to tightly track a moving or stationary setpoint, this controller knows how to actually move the robot from any initial state to a goal position in a provably stable manner. What is usually a function implemented in a robot control language becomes a servo-level behavior for the planar robot. Moving is perhaps the simplest manipulation behavior. The techniques of this chapter can possibly be used to develop controllers for other manipulation behaviors. It is important to note that the simplicity of the low-DOF planar robot is instrumental in keeping the math of this chapter manageable.

Chapter 7

Minifactory

The primary application for the planar robot is the Architecture for Agile Assembly (AAA) [66, 121], developed in the Microdynamic Systems Laboratory concurrently with the work described in this dissertation. AAA is designed to enable the development of rapidly deployable assembly systems through the use of physically, computationally, and algorithmically modular robotic agents. The primary research goal of the ongoing AAA research project is to demonstrate that this modular, distributed approach can greatly simplify and accelerate the tasks of designing, programming, constructing, calibrating, debugging, operating, and reconfiguring assembly systems.

The *minifactory* [64, 67, 68, 69] serves as an instantiation of AAA for a particular domain: the assembly of small mechatronic products that have short market lifetimes, such as pagers, cellular phones, cameras, and wearable computers. In the minifactory, shown in Figure 7.1, a set of field-joinable platens are clamped together to form an extended, continuous workspace. Modular bridges are positioned above the workspace where overhead devices, such as 2-DOF manipulators, parts feeders, welders, screwdrivers, glue dispensers, etc. can be quickly mounted. Planar robots operating on the extended platen surface serve as sub-product *couriers* in the minifactory, which have a dual-role of transporting the products between the overhead devices and cooperating with the overhead devices during part placement and other tasks. Each platen is supported by a base-unit, which also provides the *minifactory bus* for the robotic agents, with air, vacuum, power, and network facilities available in a single modular connector.

When the two translational DOFs of the courier are used cooperatively with a 2-DOF z, θ manipulator, they can together perform any of the broad class of assembly operations that SCARA robots are typically called upon to perform. However, as

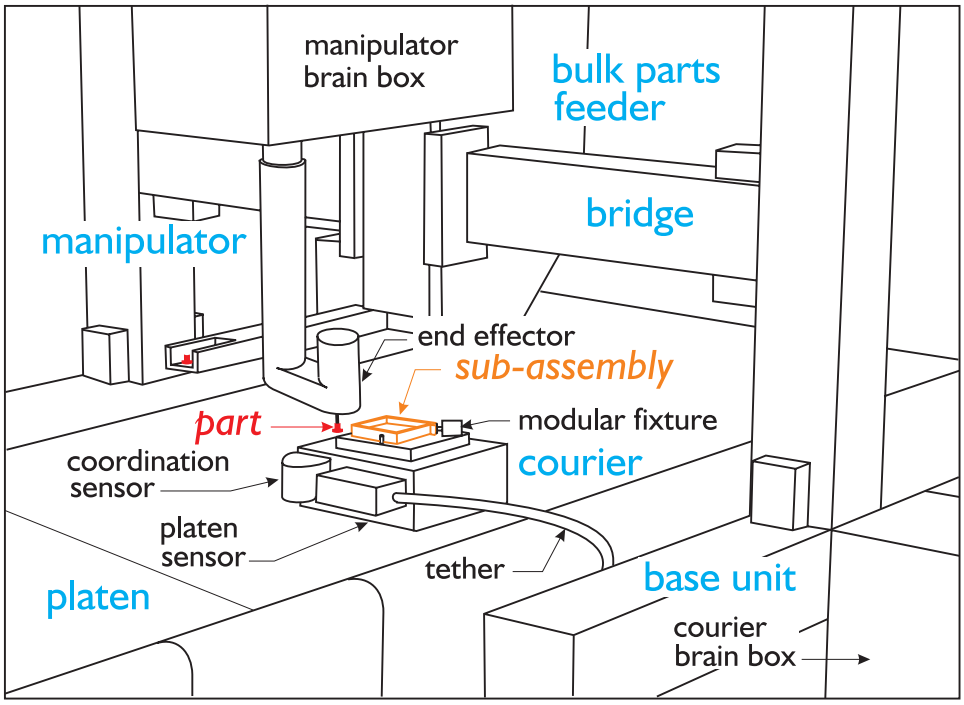


Figure 7.1: The minifactory is a rapidly-deployable assembly system composed of low-DOF robotic agents.

detailed in [65], the use of cooperative 2-DOF robots has advantages over typical robotic assembly systems. Splitting the DOFs allows *pipelining* of tasks – the courier can transport the sub-assembly to the next manipulator while that manipulator picks a part from a nearby feeder and moves it into placement position, whereas a SCARA would need to dedicate all of its DOFs to one of these tasks. This split also enhances system modularity – overhead devices for many types of operations and levels of performance can be simply clamped to the bridge, each one relying on the courier as a precision stage as needed. Each robot will have a short path to mechanical ground, minimizing transmission and link dynamics, and simplifying kinematics, dynamics, and control. By eliminating the mass associated with additional links, the robots can be sized more appropriately for the size of the products, reducing floorspace requirements, which is especially important for cleanroom applications. This reduction in robot and actuator size combined with the limited and well-defined robot workspaces also makes the system much safer, which is important when people are required for filling parts feeders and system debugging and maintenance.

Another important minifactory attribute is ubiquitousness of high-precision manipulation capabilities. Operations requiring precisions as low as a single micron

can be performed using the standard minifactory agents. It is not necessary to design special workstations isolated from the main assembly line to perform precision operations. The use of low-DOF robots makes it easier to provide this level of precision by eliminating long serial chains and transmissions. Planar robot technology provides high-precision operation over a large workspace at a reasonable cost.

There is no master factory "program." Each agent has its own program where it will try to accomplish a sequence of part or product transport and cooperative assembly operations, negotiating with its peers for the use of shared resources. This distributed programming eliminates the need for the factory programmer to develop a program that must consider dozens of interacting robots. Instead, when developing an agent program the factory programmer can limit his attention to a small section of the factory.

7.1 Minifactory agents

The rapid deployment requirements of the minifactory impose strong conditions on the robotic agents used in the system. A designer must be able to have a high-fidelity simulation of the factory for early factory design, so there must be good models for the geometry and performance of the agents. To speed factory setup, agents must be physically modular, quickly attaching to the system with clamps, and getting all their required power, pneumatic, and communications services from the single minifactory bus connector.

For scalability reasons, the minifactory avoids the use of central resources. Agents must interact on a peer-to-peer basis. Without a global overseer to verify each agent's proper operation, each agent must be *trustworthy* – it must faithfully report its state to other agents, and not do anything that it hasn't told its relevant neighbors about. Each agent must then not only know its own state, but must be able to limit its future states to those states that their peers have agreed are acceptable. If a courier says it will stay out of region $\mathcal{A}$, it must not enter that region or a collision may occur. If an courier agent tells a manipulator agent that it is at position A when it is really at position B, a part may be placed incorrectly or dropped onto the platen. These types of failures can easily cascade, requiring the entire factory to be reset and restarted.

An implication of distributed programming in the minifactory is that agent interactions are asynchronous and dynamically negotiated. Without the lockstep, pre-programmed motions seen in workcells and fixed automation systems, it is more difficult to foresee all possible courier motions and disturbances. The robustness guarantees of closed-loop operation become important for ensuring the trustworthiness of the courier agents in the presence of this added uncertainty.

With the use of low-DOF robotic agents, efficient cooperation between agents becomes very important. Cooperation may be best accomplished by several different methods, depending on the task. One agent may simply request that another move to a rendezvous position and wait while it performs some operation. An example is a simple placement operation where an overhead manipulator places an object on a sub-assembly that is positioned appropriately by the courier. A slightly more complex method for cooperation is a coordinated trajectory between the two agents. An example would be the assembly of a key-like part that needs to be inserted then twisted. If this were a low-precision operation with little or no contact forces, it could be performed by pre-planning appropriate trajectories for a courier and manipulator agent and using the minifactory network to synchronize the starting time of the trajectories. Couriers under open-loop operation would be suitable for these types of coordination, but closed-loop operation would allow improved precision, higher velocities, and robustness to uncertainties.

There are other richer methods of agent coordination. One method is coordination by physical interaction, where the agents coordinate their actions based on forces transmitted between them. A compliant insertion operation is a simple example, where a courier would assume the proper compliance relation and a manipulator would perform the insertion. This type of operation could not be performed by an open-loop courier, as its only operating mode has a very large stiffness. Even if an open-loop courier were augmented with a force sensor, only low-bandwidth operations could be performed to avoid exciting the underdamped harmonics of the open-loop controller.

Another method of coordination is the use of non-contact sensing between the agents. This sensing could come in the form of the upward-facing coordination sensor on a courier, a downward-facing camera on a manipulator, or a proximity sensor on either agent. In this case, one agent would process the sensing information and transmit the resulting error measures to the other agent over the minifactory

network. Both agents would then respond appropriately to the error signal, based on their programs for this task. These types of coordination could be performed with a courier under open-loop operation in principle, but would have to either be implemented as a look-and-move operation or limited to very low velocities, neither of which would permit dynamic interactions or coordination between agents.

Although the minifactory project is still a work-in-process, the courier agents have been substantially implemented. The following sections describe several minifactory related operations designed to highlight both the usefulness of the planar robot and the benefits of closed-loop operation.

7.2 Factory initialization

When the minifactory is assembled, it would take too much time to precisely mount the overhead robots such that they match the factory design. It is also inconvenient to have to manually move all the couriers to known positions when starting or restarting the factory. Instead, the minifactory requires agents that can automatically initialize and collectively discover the precise mounting location of every agent and platen boundary in the system.

Because the overhead devices have small workspaces, the couriers are required to do most of the work of factory initialization. Couriers must first initialize themselves. Their platen sensors can only measure positions relative to their arbitrary starting position, so the couriers must explore their environment to discover enough features to localize their position. In addition, the skew angle of the courier must be manually aligned to within a few degrees or the actuators will not function properly.

7.2.1 Skew angle initialization

In commercial Sawyer motor systems, the forcers must be manually aligned with the platens during system startup. This operation is easier if there are currents in the actuators that correspond to a fixed open-loop position command. In this case, there will be several rotational detents near the zero skew angle, spaced by approximately 0.6°, depending on the size of the actuator teeth. It requires good lighting and eyesight to align the edge of the motor with the vertiginous grid of platen teeth, and requires good dexterity to twist the motor into the proper 0° detent. However, the

platen sensor is operational over a rotational range of several degrees and has been used to detect a misalignment at startup. If the misalignment is within the working range of the actuators, it can also be automatically corrected, eliminating the need for the factory operator to struggle with courier alignments. Manual alignment of the couriers to within the operating range of the actuators (approximately 1.8° for the Normag forcers) is extremely easy.

A realignment routine has been implemented that locks the courier position using an open-loop controller and reads the sensor to determine whether the forcer is aligned, misaligned, or unrecoverably misaligned. If it is unrecoverably misaligned the user is alerted. If it is misaligned, an open-loop rotation command is used to correct the motor alignment. This process repeats until the motor is successfully aligned, at which point the tooth counts for each sensor segment are reset to zero, and closed-loop operation can commence.

7.2.2 Startup exploration

After the couriers have determined and corrected their skew angles, it is necessary for them to explore their environment. The two main tasks are to locate the platen boundaries (which will have bumpers) by bumping into them, and to locate the overhead devices by detecting their LED beacons with the coordination sensor. In addition, there may be multiple couriers exploring the same portions of the factory simultaneously, so the couriers also need to be able to detect and recover from collisions. An open-loop courier would need some form of contact sensing around its perimeter and would have to proceed very cautiously to minimize collision impacts which can easily excite the stiff, underdamped dynamics, resulting in an unrecoverable rotation.

In contrast, closed-loop operation can be used for exploration without extrinsic force sensing. The disturbance estimator of Section 5.2 can be used to provide a measure of the external forces acting on the forcer at bandwidths of 50 Hz. In addition, the instabilities that arise when a stiff robot transitions from free-space motion to contact motion can be eliminated by using an impedance controller.

To demonstrate this mode of closed-loop operation, a simple exploration task was designed. The courier, starting at an arbitrary location, is to map out the boundaries of its workspace. For simplicity, we assume the boundaries are within

the range of the courier's tether, boundaries are orthogonal to the tooth grid on the platen, and there are no obstacles internal to the boundary.

Starting in the workspace, the courier first must find a boundary. Assume, without loss of generality, that the initial search is in the direction of the courier's first translational axis. An impedance controller of the form

$$u_{w1} = -b_1 \dot{q}_1 + f_{d1} \tag{7.1}$$

is applied to the courier's first translational axis. Note that for unconstrained motion with no disturbances, the courier will not exceed a speed of $\dot{q}_1 = b_1/f_{d1}$. The damping parameter b_1 is chosen based on this desired free-motion speed. When contact is made with a boundary, the controller will push with a force of f_{d1}, which is chosen high enough so that tether disturbances will not be able to pull the forcer away from the boundary. Note that both free-space and contact modes are handled by a single controller so that chattering instabilities are avoided. The second translational axis uses a simple PD regulation controller

$$u_{w2} = -k_2(q_2 - q_2^*) + k_{d2}\dot{q}_2 \tag{7.2}$$

with the desired position q_2^* fixed at the initial courier position, and proportional and derivative gains k_2 and k_{d2} chosen appropriately. The rotational axis uses a non-linear regulation controller of the form

$$u_{w3} = -k_3(q_3 - q_3^*) - k_{c,3}(q_3 - q_3^*)^3 - k_{d3}\dot{q}_3, \tag{7.3}$$

where $k_{c,3}$ is a non-linear position gain designed to prevent large motor rotations. This non-linear gain allows the proportional position gain k_3 to be smaller, which reduces audible noise caused by the angle servo responding to small disturbances stemming from angle accuracy errors. These three axis controllers can be combined into a single hybrid[1] impedance *bump* controller.

The bump controller was implemented on the *normag2* system, with gains given by k_2 =58000 N/m, k_3 =100 Nm/rad, k_{d2} =616 N/m/s, and k_{d3} =1.5 Nm/rad/s. To allow sufficient force to overcome tether forces, f_{d1} =20 N, and to specify a maximum velocity of 0.5 m/s, b_1 =40 N/m/s. The courier was positioned so that moving in the $-y$ direction will cause it to bump into an effectively rigid obstacle.

[1]Hybrid in this context refers to the use of different types of controllers in different directions.

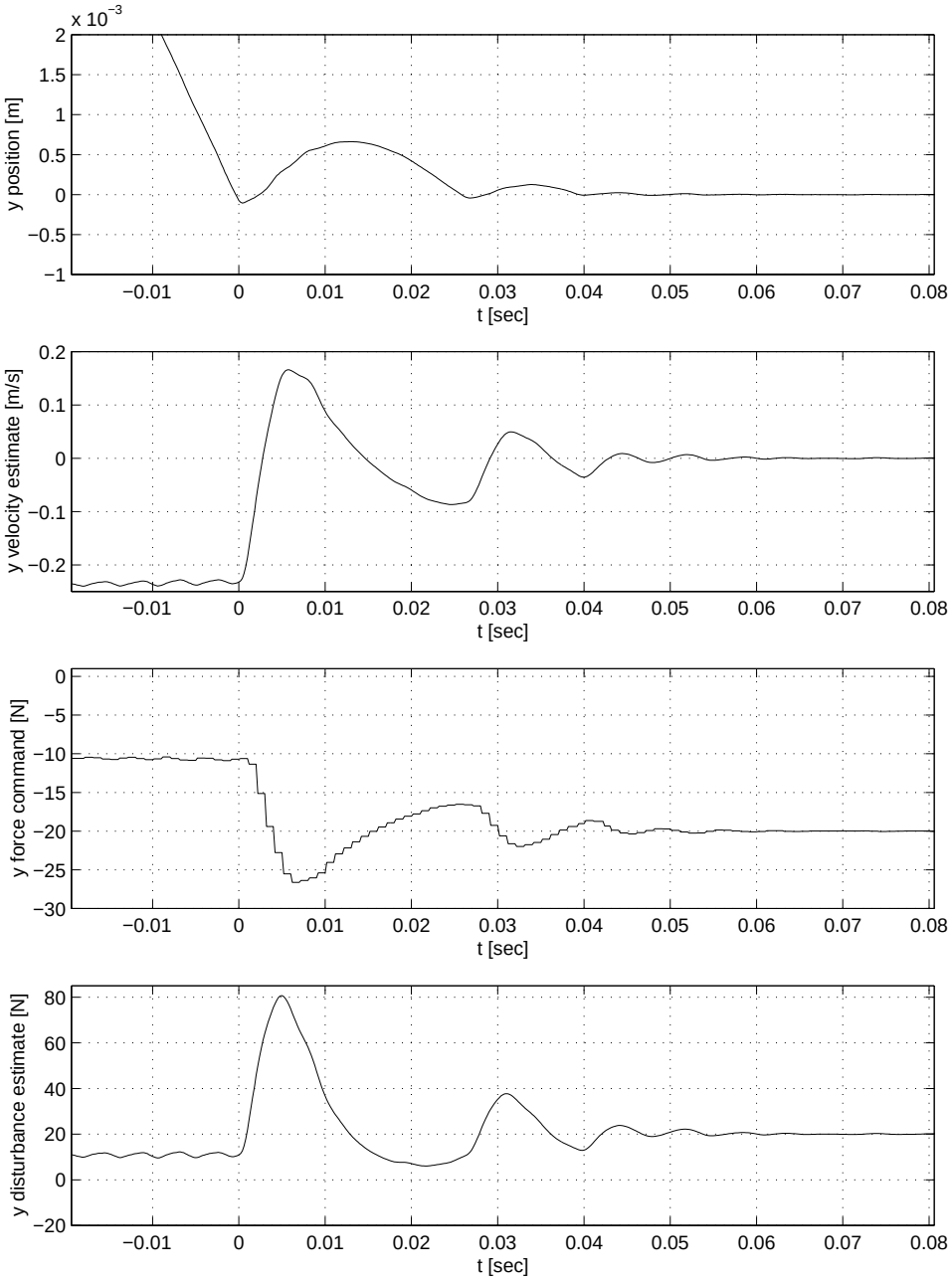


Figure 7.2: An impact with a rigid obstacle produces bouncing, but no controller instability.

The obstacle was also centered so that the impact would not generate significant torques on the forcer. The commanded forces, disturbance estimates, and positions before, during, and after the collision are shown in Figure 7.2. For $t < 0$, the forcer approaches the boundary at a velocity of 0.24 m/s. This velocity is slower than the maximum 0.5 m/s design velocity because the controller does not feed-forward the damping effects discussed in Chapter 3 or the tether disturbance forces. After the impact, there is an oscillation in the x_2 direction of motion, which is to be expected. Note that the commanded force, u_{w2}, also oscillates, but is smooth[2]. With a single controller handling both cases, there is no instability caused by the transition from the free-space motion to contact, and the bouncing is dissipated by the impedance controller and other damping in the system within 60 ms.

After the higher-level controller detects that the courier has stopped moving, the bump controller will be pressing against the boundary with a force of f_{d1}. For this boundary exploration, the next step is to find the extent of the current boundary. A *slide* controller which is similar to the bump controller above, but has impedance controllers in both the translational axes, is used to slide along the wall without losing contact. Two other appropriate controllers are used to notify the higher-level controller that the end of the boundary was found, and whether it is an internal or external corner. The process is then repeated so that the courier will discover the entire boundary. This boundary exploration was implemented on the *normag2* system using obstacles set up on one half of the Normag platen. Results are shown in Figure 7.3. With the maximum velocity set to 0.4 m/s, the 3.48 m perimeter is explored within 18.5 s[3], for an effective exploration velocity of 0.18 m/s, including localization of the first boundary.

The courier also needs to find the overhead devices. An exploration algorithm has been developed by Butler, et. al. [70, 71] that creates a map of the overhead device locations and platen boundaries using multiple couriers. Experimental implementation of this technique is in progress using the controllers presented above. Preliminary experiments suggest that closed-loop operation allows exploration to

[2]The discretization visible in the u_{w2} plot is because the controller command is only updated at 1 kHz rather than the 5 kHz commutation rate.

[3]Any form of cognition is beyond the scope of this experiment, so completion of the boundary loop is not detected.

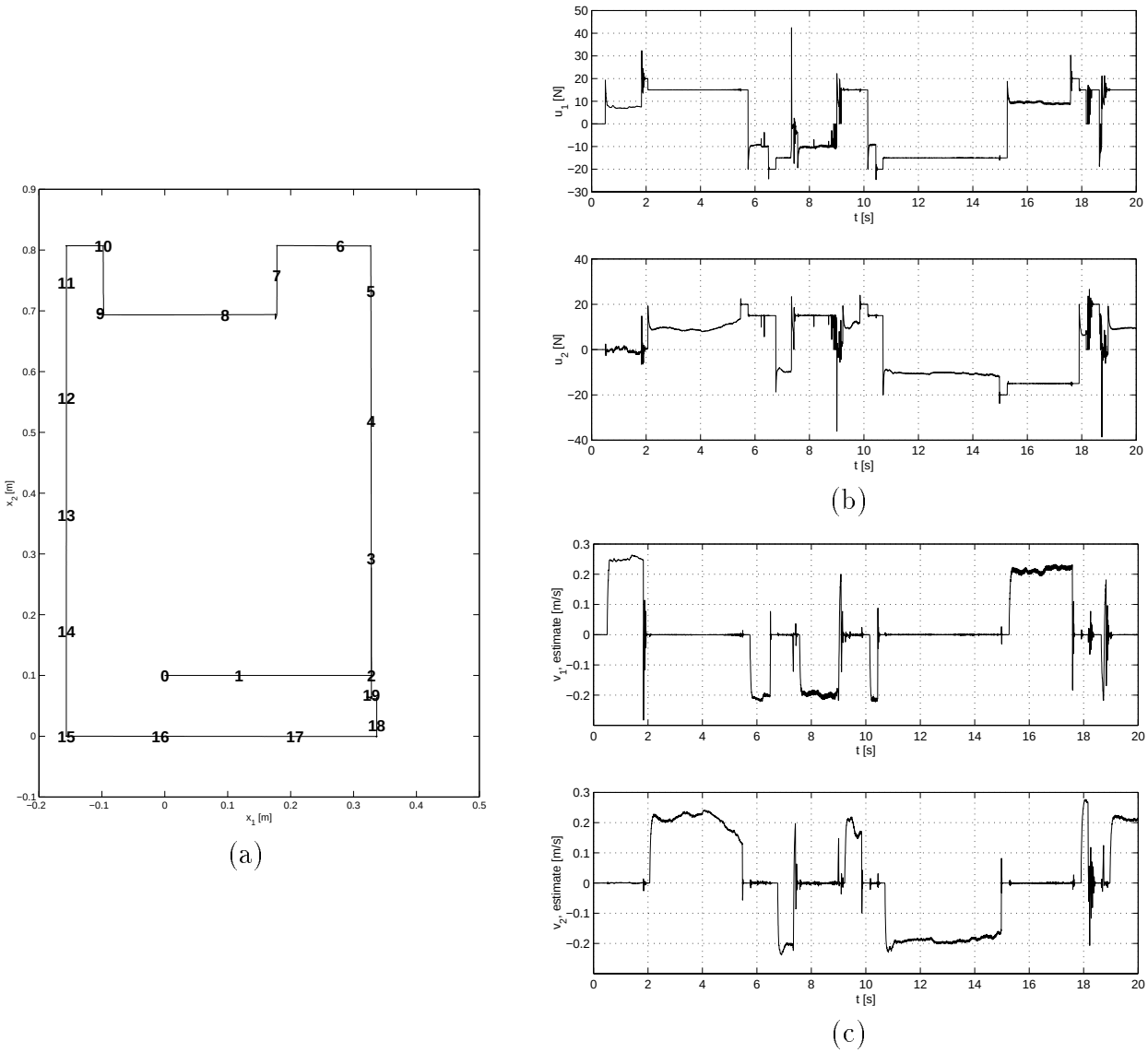


Figure 7.3: Perimeter following results (a) configuration space courier position trace (numbers indicate elapsed time) (b) courier force commands (c) courier velocities.

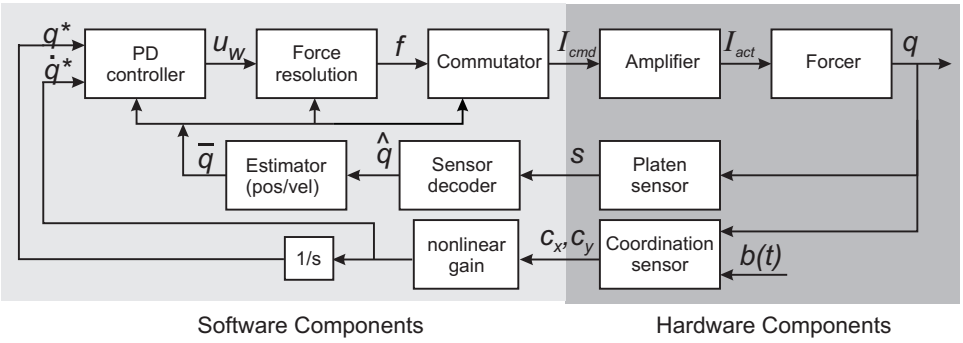


Figure 7.4: Block diagram for tracking of overhead beacons

proceed more than twice as fast compared to an open-loop courier augmented with contact sensors[4].

7.3 Multi-robot coordination

Experiments have been performed using both the coordination sensor and an end-point camera on an overhead manipulator to demonstrate both high-speed tracking and precision visual positioning, both which are expected to be important for cooperative minifactory tasks.

7.3.1 High-speed tracking

To demonstrate the types of high-speed, dynamic interactions that are possible with closed-loop operation, the coordination sensor is used for simple coordination between a courier and an overhead manipulator. The manipulator and courier begin in positions such that the LED beacon on the manipulator's end-effector is within view of the courier's coordination sensor. The courier's task is to track the position of the manipulator's LED beacon. There is no communication between the agents.

A block diagram of the courier's beacon tracking controller is shown in Figure 7.4. A PD controller is used for the basic control of the courier. A nonlinear function of the coordination sensor position output, which is itself proportional to the offset between the courier position, q, and the beacon position b, serves as a velocity command for the courier controller. This velocity command is also integrated and

[4]Actually, the platen sensor output was used to simulate contact sensing

used as the position command of the courier controller. The nonlinear tracking function is used to balance the tradeoff between responding strongly to large tracking errors and stable operation at rest.

A (z, θ) overhead manipulator was first commanded to follow a sinusoidal trajectory using its rotational axis. An LED beacon on the end effector of the manipulator 72 mm from its center of rotation was used as the tracking target. The manipulator was commanded to track a sinusoidal trajectory with a total travel of π rad at a frequency of 6 rad/s, which was near the limits of the manipulator velocity limits. The beacon height was approximately 80 mm above the coordination sensor. Figure 7.5 shows the coordination sensor outputs, converted to position tracking errors in the x, y plane of the courier, together with the position and velocity of the courier. Note that the tracking error is rather large, but is smaller than the field of view of the coordination sensor (about 30 mm) for the height of the beacon for this test, which was approximately 80 mm above the coordination sensor, so that significantly higher tracking velocities are possible. If tracking error is important, stiffer tracking gains or integral gains can be used.

To test the limits of the tracking ability, a handheld beacon was manually waved above the courier. The resulting beacon motion is difficult to characterize, but the tracking results of Figure 7.6 show courier velocities above 1 m/s while tracking errors do not rise much above 20 mm.

7.3.2 Precision visually guided positioning

Others [122] have used the closed-loop courier to perform visually-guided positioning of a small medical device. The results of their work are summarized in this section.

An overhead manipulator holds a 3 mm square medical device, to be positioned relative to a similarly sized part held in a palette on the courier. The manipulator has a camera, shown in Figure 7.7, and the relative positions of the two parts are tracked at frame rates. The errors in the part placement are translated from camera coordinates to courier coordinates and transmitted to the courier using the networking infrastructure of the minifactory. A simplified block diagram of the cooperative visual controller is shown in Figure 7.8, where H represents the visual processing algorithm, G_c contains a position and integral gain for the visual controller, and J maps from image coordinates to courier coordinates.

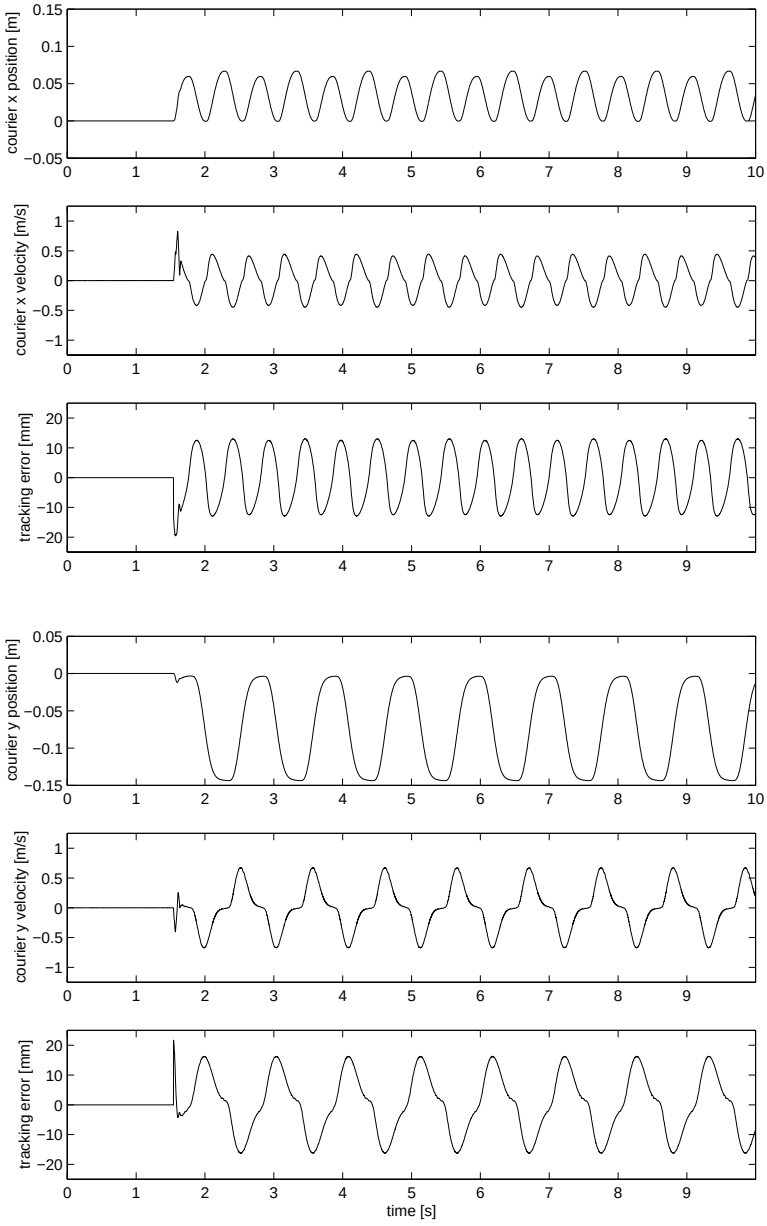


Figure 7.5: Closed-loop tracking of overhead manipulator beacon

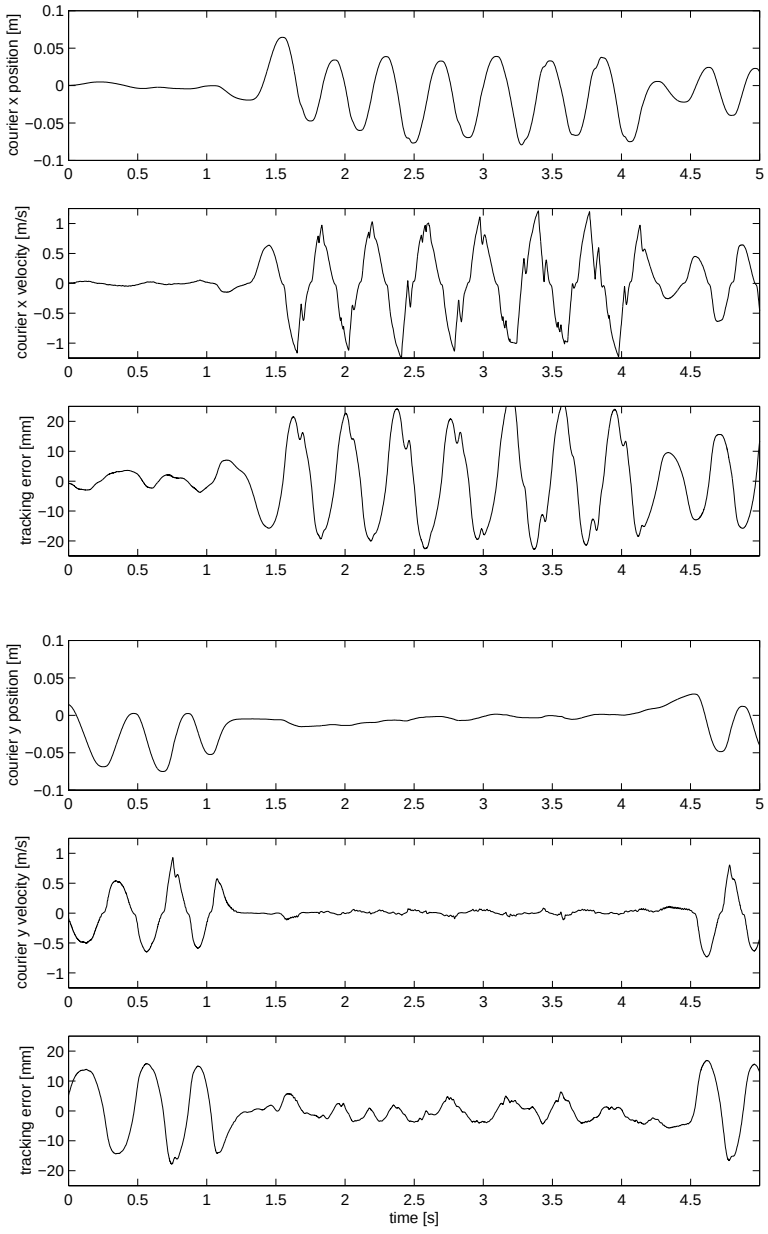


Figure 7.6: Closed-loop tracking of handheld beacon

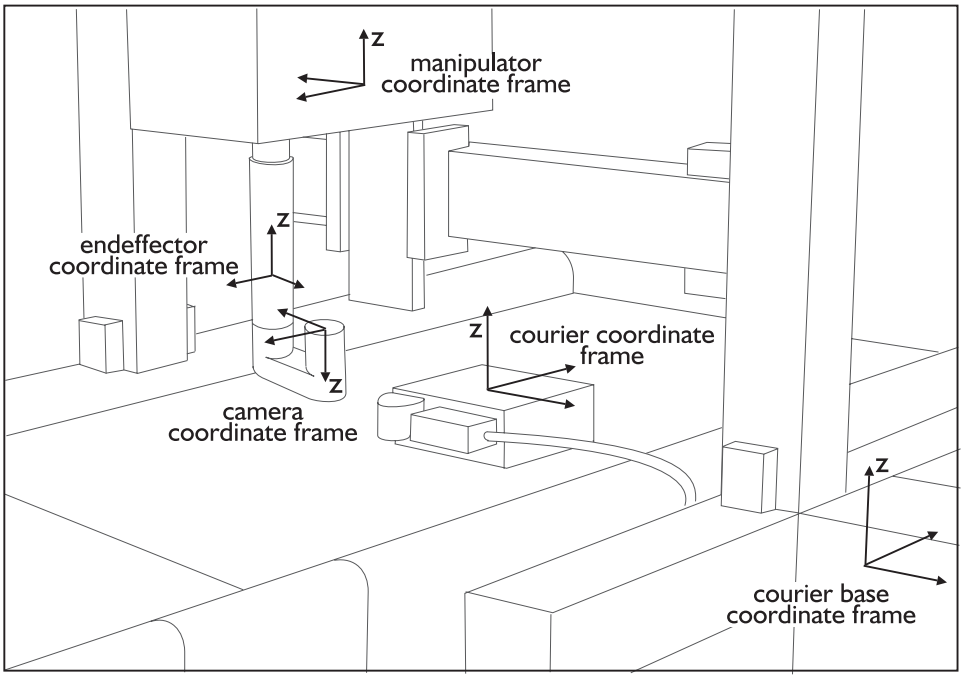


Figure 7.7: Robotic system for visually guided positioning (from [122])

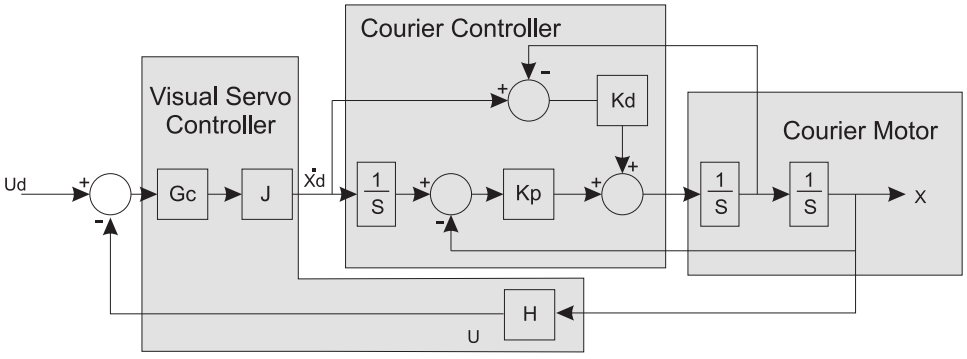


Figure 7.8: Block diagram for visually guided positioning (from [122])

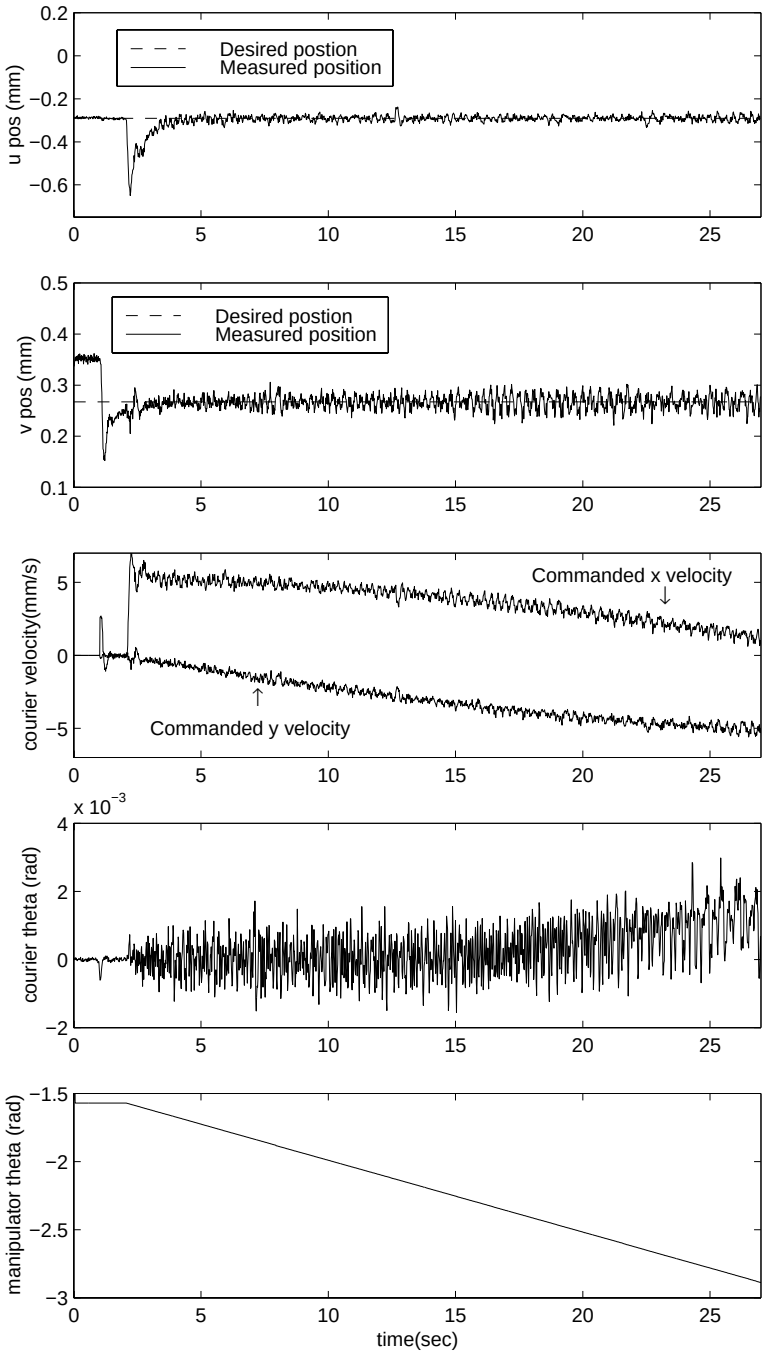


Figure 7.9: Results for visually guided positioning (from [122])

Starting from a rest position, the manipulator was rotated at a velocity of 0.052 rad/s to act as a disturbance to the visual servoing. The resulting courier velocity is approximately 5 mm/s. The resulting tracking errors over 25 s of motion converged to mean errors of 1 μm and standard deviations of 14 μm, as shown in Figure 7.9. For reference, one pixel of the camera corresponds to approximately 10 μm. After motion began, it took approximately 2 s for the tracking error to converge to the steady-state condition. Platen sensor angle measurement inaccuracies as the courier moved were blamed for the small remaining 14 μm of tracking noise.

7.4 Discussion

Planar motors are a key technology in the minifactory instantiation of AAA. The ability to provide fast motions over a large workspace is critical for providing reasonable performance of the minifactory system. Frictionless operation allows for precise positioning, and the platen acts as a repeatable position reference over the entire factory. The large peak force to mass ratio allows for fast motions and large stiffnesses. The compact size dramatically decreases the overall factory size, and makes them very safe. Note that all these arguments apply to the planar motor operated in stepping mode.

However, without the work detailed in this dissertation, planar motors would be unacceptable for use in the minifactory. Although the performance advantages of closed-loop control detailed in Chapter 5 are helpful in providing quantitative improvements to factory performance that will expand its economically feasible domain, the qualitative changes in operation that allow for dynamic interactions with the environment and other robotic agents are absolutely critical for ensuring the reliable performance of the overall minifactory system.

Chapter 8

Application: A miniature mobile parts feeder

Although the experiments of the previous chapter demonstrate a number of ways that the planar robot can interact with its environment and other robots, explicit manipulation tasks were limited to positioning.

In this chapter the planar robot is used for a form of distributed manipulation [105] – the use of a *single* rigid body to simultaneously manipulate *multiple* parts. The manipulation occurs in multiple locations over the surface of the body, enabling efficient parallel and/or pipelined operation while requiring only a single set of actuators–those needed for moving the body itself. The theoretical problem is to find a shape and motion for the body so that multiple parts can be manipulated in a useful manner. A related practical problem is the design of the actuation system to realize the desired body motions.

Many have considered combining relatively simple body shapes with vibratory motions, as depicted in Figure 8.1. Traditional vibratory feeders (a) use inclined vibrations to move parts along a feed track or up a spiral track on the inside of a bowl [123]. Traps or gates are often placed along the feed path to filter out incorrectly oriented parts. Rotational vibrations (b) of a plate [124] have also been used to orient and localize parts. A bouncing strategy (c) where fast vertical motion of a plate are combined with transverse motion during part impacts has been explored for parallel part reorientation [125, 126]. The Dyna-Glide system (d) combines vertical vibrations with a carpet of inclined fibers [127]. Sections of carpets with different directions of inclination can be used over an area to create a desired feed path for parts. It has also been shown [104] that horizontal vibrations of a horizontal plate (e) can be used to move parts in the plane. Surprisingly, this technique was extended

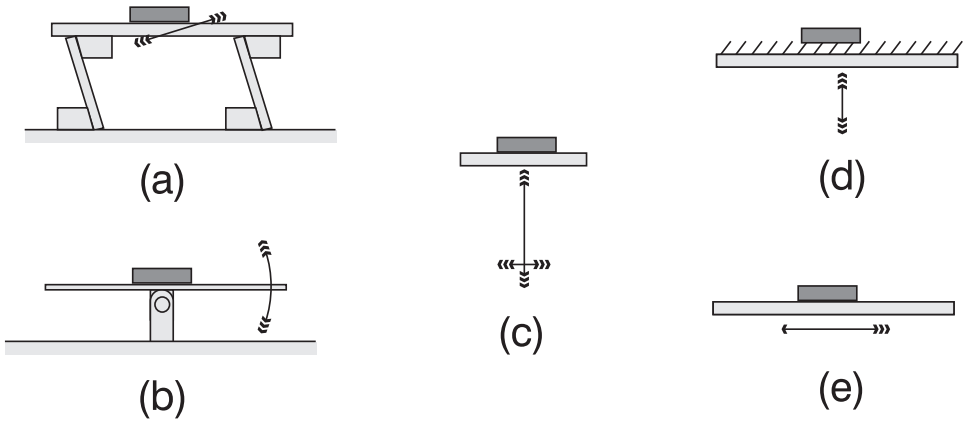


Figure 8.1: Researchers have explored a variety of distributed manipulation techniques using oscillating plates.

[13] to independently control the feed directions of multiple parts sitting on the same horizontal plate.

It is important to specify the scope of a distributed manipulation system, which may include repositioning, singulating, reorienting, sorting, or even assembling parts. This chapter is concerned with the bulk parts feeding problem: starting with a pile of parts, singulate, position, and orient them for presentation to an assembly device. Of course, parts feeding is of critical importance in automated assembly, and has received much attention. Successful commercial bulk feeders include vibratory feeder bowls [123], the Adept FlexFeeder [128, 129], and the Sony APOS system [130]. It is interesting to note that these systems have in common a recirculation path, reorientation facility, and sorting capability. They provide mechanisms for some of the parts to assume the desired orientations, and allow the rest of the parts to be recirculated. Ensuring the parts are in the proper orientation may be done by mechanical means, such as bowl feeder gates or the APOS tray detents, or through sensing, as in the Adept FlexFeeder vision system. A major advantage of these systems is that every part does not have to be correctly oriented in its first trip through the feeder, allowing simple designs that do not sacrifice robustness.

The remainder of this chapter presents the operating principles and simulation results of a novel bulk parts feeder. This feeder, introduced in earlier publications [14, 131] seeks to provide a recirculatory feed path in an extremely compact space. A significant novelty is the feeder's mobility, allowing it to supply multiple overhead robots with parts, even if their workspaces do not overlap. The following section

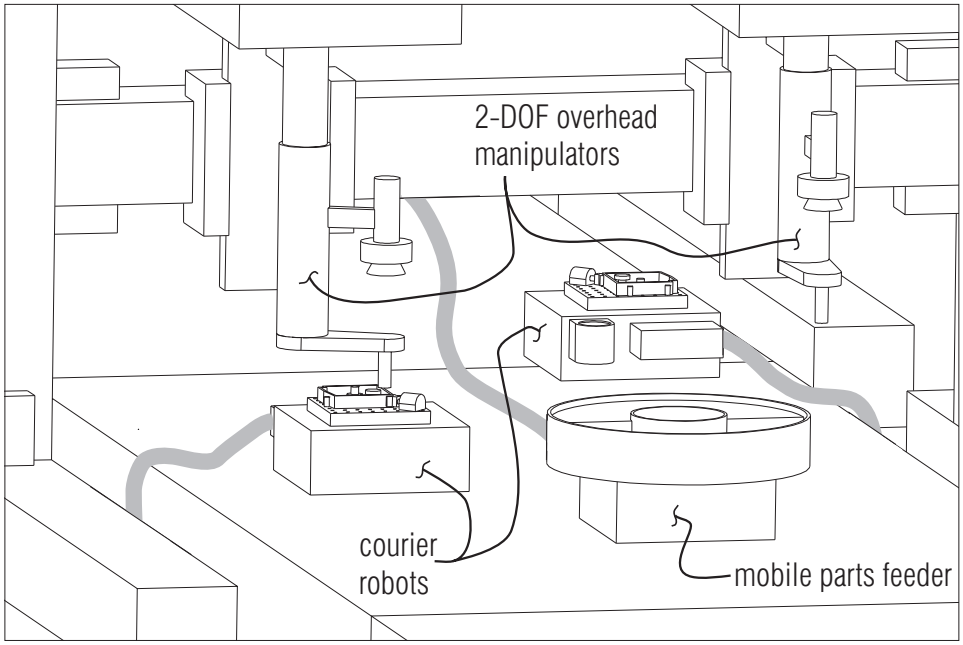


Figure 8.2: A miniature mobile parts feeder in a minifactory setting: a single feeder supplies parts to multiple low-DOF overhead manipulators.

provides an application to demonstrate how this mobility can be useful and also presents the basic concept for the feeder.

8.1 Application example

Figure 8.2 shows the proposed feeder in a minifactory setting. Recall that in the minifactory, the low-DOF overhead manipulators do not have large workspaces, so parts *must* be fed close to the assembly locations. In this context, it is very useful to have a mobile parts feeder that can move under the manipulator with an oriented part, allow the manipulator to pick it up, and then move out of the way.

The proposed mobile parts feeder is depicted in Figure 8.3. Physically, it consists of a special feed tray rigidly attached to a planar motor. The tray has an annular feed path for parts, with a sloped *ramp* section, and a flat *plateau* section. The motor performs a rotational vibration, resulting in a counter-clockwise motion of the parts. When bulk parts are loaded at the bottom of the ramp, parts slowly climb the ramp, but only near the outside edge, resulting in a single-file line. Once in the plateau section, the parts speed up and spread out. They continue to move around the plateau, where an overhead vision system can be used to identify parts

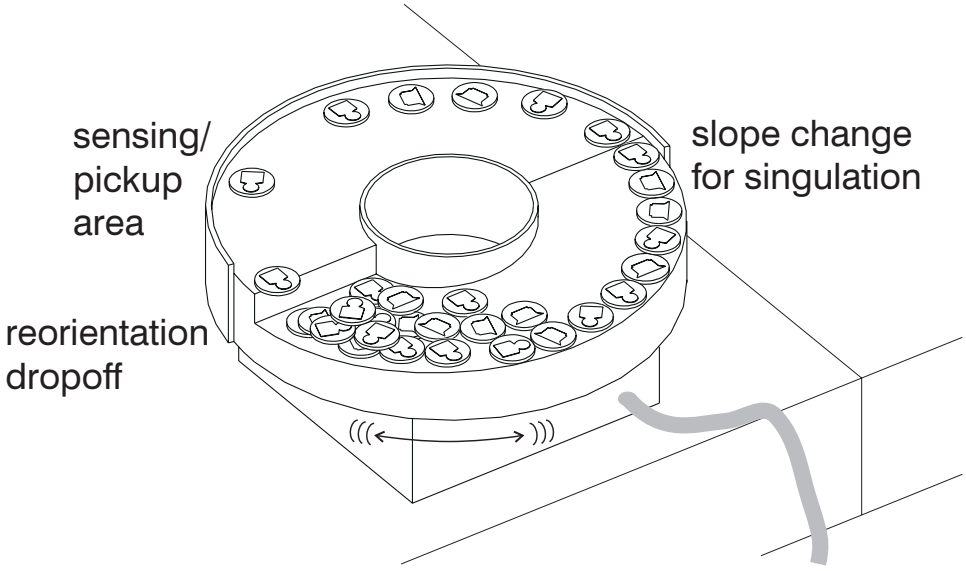


Figure 8.3: A miniature mobile parts feeder, consisting of a partially sloped annular feed tray (outside edge partially omitted for illustrative purposes) mounted on a planar robot.

in the correct orientation.[1] Incorrectly oriented parts are reoriented as they pass over the dropoff and return to the pile of bulk parts. The next section discusses the details of using horizontal vibrations to move parts. Section 8.3 describes how ramps can be used for singulation. The complete feeder design is then described and simulated in Section 8.4.

8.2 Moving parts with horizontal vibrations

In this section, the basic technique for generating part motion is presented. A model is derived and experimental results are presented. As this feeding technique is related to that used by other researchers [104], differences between the approaches are highlighted.

[1]It is also possible to place indentations, fences, etc. in the plateau region to separate parts in the correct orientation without the use of vision. However, unless a removable insert contains all the part specific features, the feeder's flexibility will be somewhat compromised.

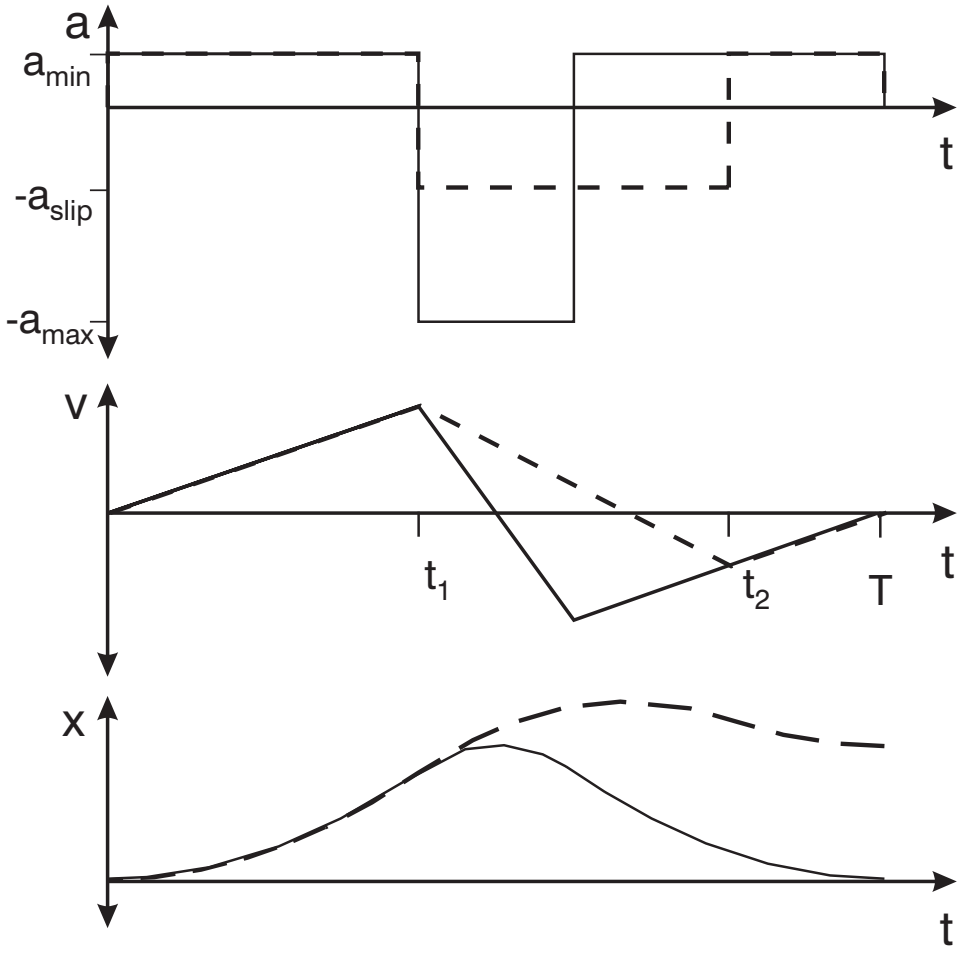


Figure 8.4: Planar vibratory feeding: at each cycle, the part (dashed trace) moves forward but the tray (solid trace) returns to the original position.

8.2.1 Stick-slip waveform

There is a common magician's trick where a tablecloth is removed from under a table setting by quickly jerking the cloth. This trick demonstrates how planar motion can cause relative motion between two objects, and inspired this exploration of the application of closed-loop planar motors to parts feeding.

Using a flat plate for the feed tray, motion of parts relative to the tray is achieved by accelerating the feeder rapidly in one direction such that the part slips on the tray, followed by a return to the original position with accelerations slow enough that the part "sticks" to the tray. A periodic waveform with such a *stick-slip* nature is shown in Figure 8.4.

This waveform is defined as:

$$a(t) = \begin{cases} a_{min} & 0 \le t \le t_1 \\ -a_{max} & t_1 < t \le T - t_1 \\ a_{min} & T - t_1 < t \le T, \end{cases} \tag{8.1}$$

$$v(t) = \begin{cases} a_{min}t & 0 \le t \le t_1 \\ a_{min}t_1 - a_{max}(t - t_1) & t_1 < t \le T - t_1 \\ a_{min}(t - T) & T - t_1 < t \le T, \end{cases} \tag{8.2}$$

where t is the time within the current cycle. The acceleration switch time t_1 is computed as:

$$t_1 = \frac{a_{max}}{a_{min} + a_{max}} \frac{T}{2}, \tag{8.3}$$

so that $x(T) - x(0) = \int_0^T v(t) = 0$, and the feeder has no net motion.

The remaining parameters of the waveform to be chosen are the period T, the acceleration during the slip phase a_{max}, and the acceleration during the stick phase a_{min}. The choice of T should keep the waveform frequency low enough to be in the dynamic range of the robot, but high enough to limit the velocities and displacements required of the feeder. For slipping and sticking to occur, a_{max} and a_{min} should be chosen to meet the constraints $a_{max} > \mu g$ and $a_{min} < \mu g$, where g is the gravitational acceleration and μ is the coefficient of friction between the part and feed tray.

Assuming the part is sticking to the tray at the start of the waveform and its motion is restricted to the direction of tray motion (i.e. no rolling or transverse motion), the part will move as:

$$a_p(t) = \begin{cases} a_{min} & 0 \le t \le t_1 \\ -\mu g & t_1 < t \le t_2 \\ a_{min} & t_2 < t \le T, \end{cases} \tag{8.4}$$

$$v_p(t) = \begin{cases} a_{min}t & 0 \le t \le t_1 \\ a_{min}t_1 - \mu g(t - t_1) & t_1 < t \le t_2 \\ a_{min}(t - T) & t_2 < t \le T. \end{cases} \tag{8.5}$$

The part catches up to the tray at time

$$t_2 = t_1 + \frac{a_{min}T}{\mu g + a_{min}}, \tag{8.6}$$

and the average part velocity over one waveform, $\bar{v}_p$, is computed as:

$$\bar{v}_p = \frac{T a_{min}}{2} \left(\frac{-1}{1 + \frac{a_{min}}{\mu g}} + \frac{1}{1 + \frac{a_{min}}{a_{max}}} \right). \tag{8.7}$$

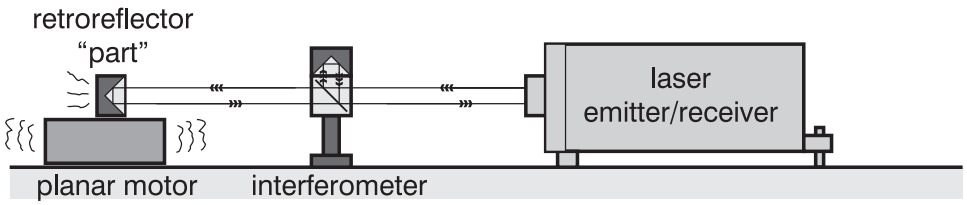


Figure 8.5: Experimental setup for verifying feed operation: a laser interferometer measures the position of a retroreflector "part," while the integral planar motor sensor measures the motor position relative to the platen.

To verify the feeding principle, the waveform of (8.1) and (8.2) was used as the input to one axis of a 3-DOF PD controller controlling a planar motor. Parts such as coins, rubber grommets, and plastic pieces with varying friction coefficients were placed on a flat feed tray attached to the motor. Although it was possible to find waveforms that would feed the parts well, not all theoretically acceptable waveforms worked. To investigate further, a stainless steel block containing a glass cube corner was used as the "part," whose position relative to the stationary platen could be measured using a laser interferometer, as shown in Figure 8.5. The motor's position relative to the platen was measured by the integral planar motor sensor. Both position measurements were precise and at a high enough bandwidth to allow for velocity measurements by simply taking the difference of consecutive measurements.

Waveforms that worked well appeared as in Figure 8.6, with the motor tracking the commanded velocity fairly well. The part velocity deviates from the motor velocity during the slip phase and tracks it closely during the stick phase, as expected. Given the feeder waveform parameters and estimated friction coefficient, $\bar{v}_p$ is computed using (8.7) as 11.8 mm/s. In the experiment, the part traveled 2.48 mm over 0.2 s, for an average velocity of 12.4 mm/s, in close agreement with the computed value.

In waveforms that did not work well, the motor velocity had large deviations from the commanded velocity. These experiments were performed before the actuator calibration of Chapter 4 were completed and errors in the actuator force outputs contributed to these deviations.

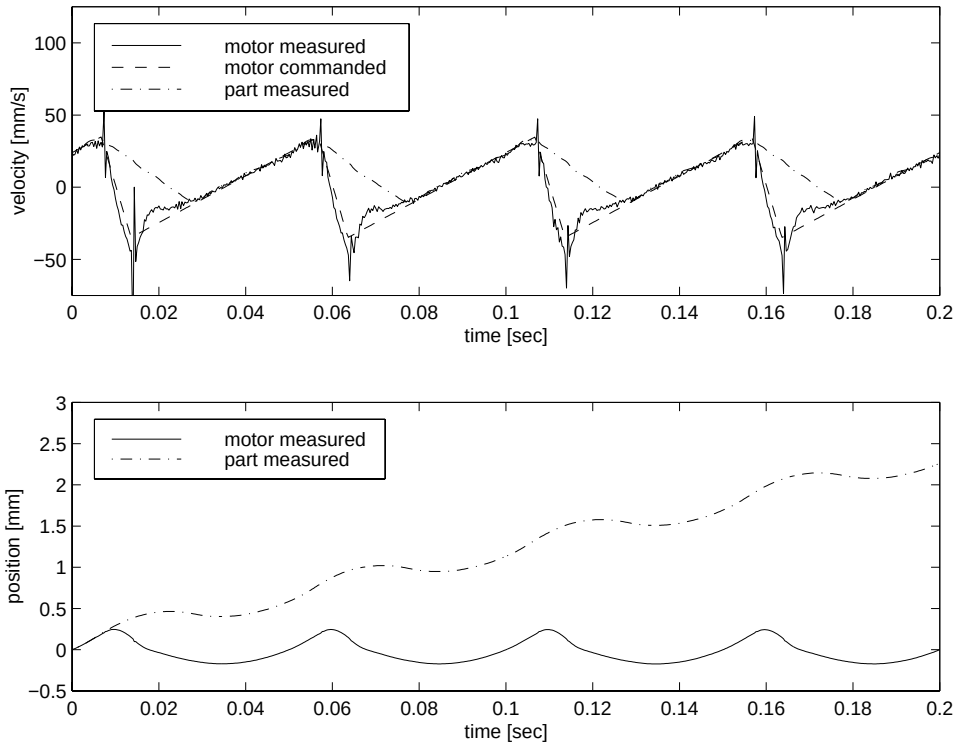


Figure 8.6: Experimental measurements of feeder and part motion using a stick-slip waveform with $T = 0.05$ s, $a_{min} = 1.6$ m/s^2, $a_{max} = 10.1$ m/s^2. The part is stainless steel and the tray is aluminum; based on the part motion during the slip phase, the coefficient of friction appears to be approximately 0.2.

8.2.2 Coulomb pump waveform

A *Coulomb pump* waveform has also been used [13, 104] to achieve part motion in the plane. This waveform is given by:

$$a_t(t) = \begin{cases} a_{max} & 0 \le t < t_1 \\ 0 & t_1 \le t < t_2 \\ -a_{max} & t_2 \le t < T, \end{cases} \tag{8.8}$$

$$v_t(t) = \begin{cases} a_{max}\left[\frac{T}{4}(z^2-1)+t\right] & 0 \le t < t_1 \\ a_{max}\frac{T}{4}(z-1)^2 & t_1 \le t < t_2 \\ a_{max}\left[\frac{T}{4}(z^2+3)-t\right] & t_2 \le t < T, \end{cases} \tag{8.9}$$

where $t_1 = (1-z)T/2$, $t_2 = (1+z)T/2$, and z is a parameter controlling the fraction of a cycle used by the constant velocity portion of the waveform.

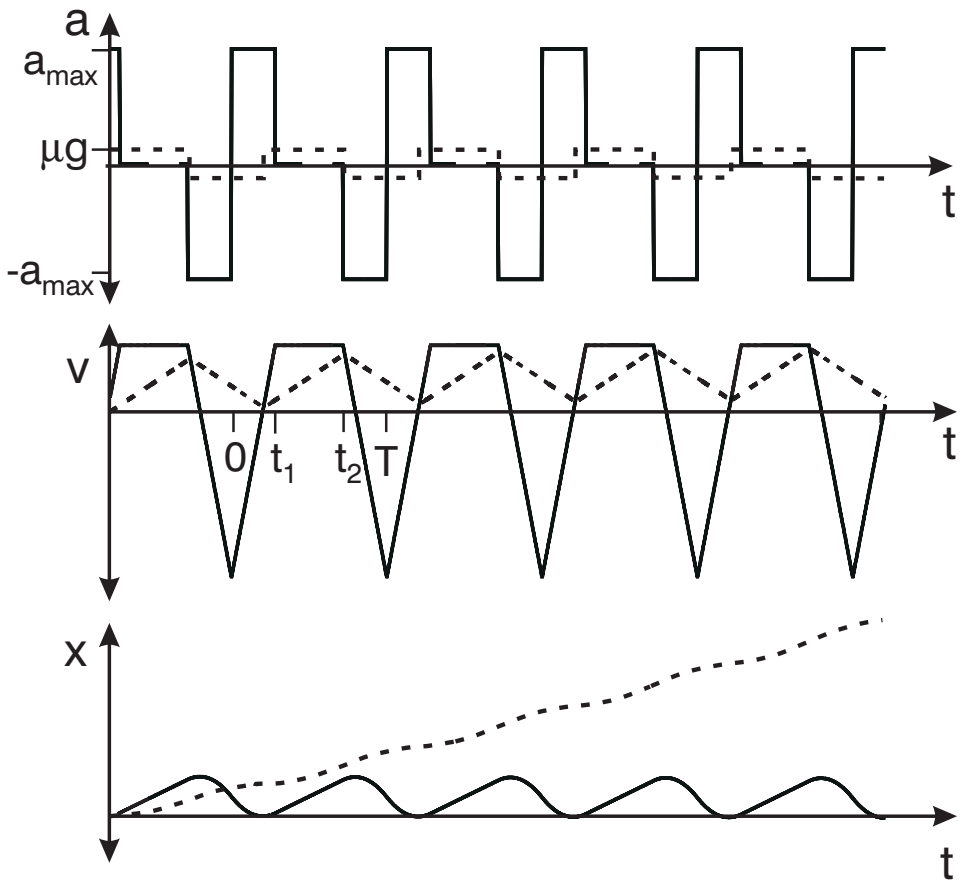


Figure 8.7: The Coulomb pump waveform can achieve higher part velocities over multiple cycles.

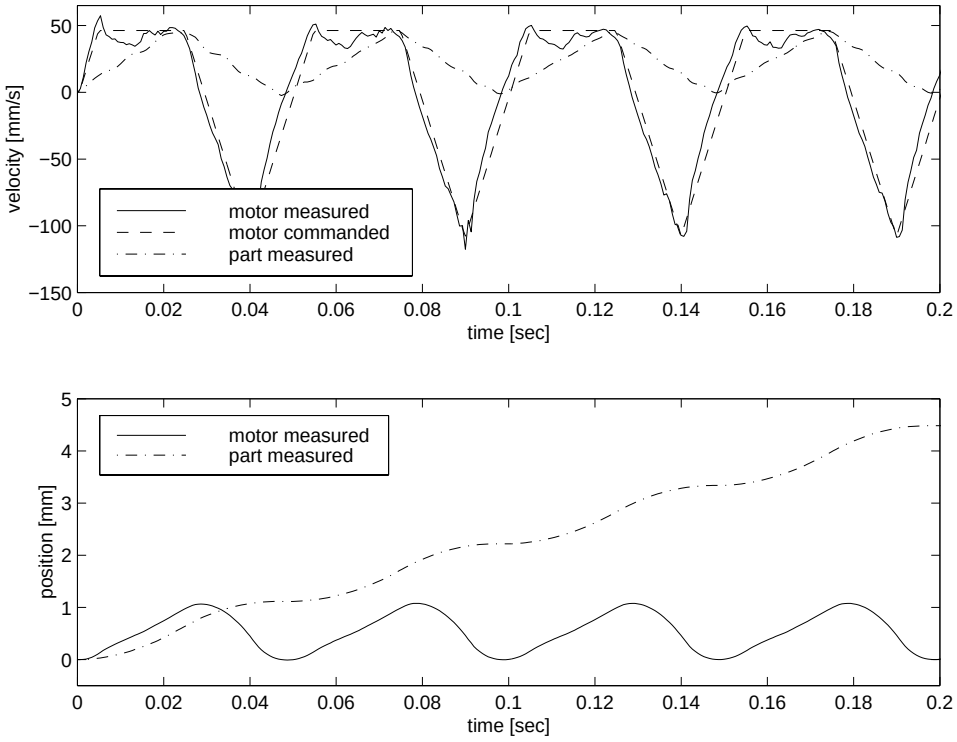


Figure 8.8: Experimental measurements of feeder and part motion using a Coulomb pump waveform with $T = 0.05$ s, $a_{max} = 10.3$ m/s^2, and $z = 0.37$.

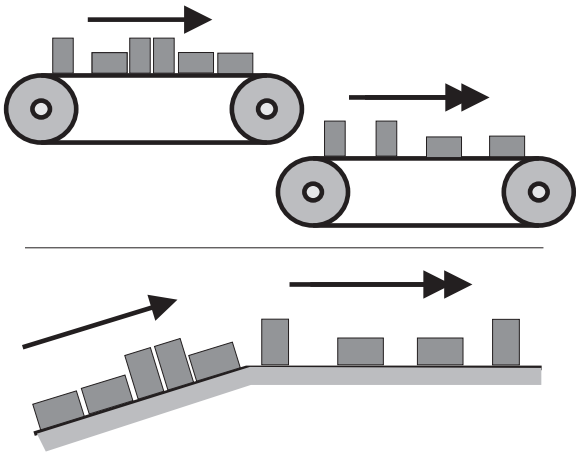


Figure 8.9: A change of feed velocity based on part position can singulate parts in the direction of motion. This technique can be implemented with conveyors (above) or a sloped feed tray (below).

As shown in Figure 8.7, the part motion changes from cycle to cycle, with each cycle "pumping up" the velocity via the Coulomb friction forces. The part velocity is now best characterized by the *equilibrium velocity*, which is the value that the average waveform velocity eventually reaches, given by [104]:

$$v_{eq} = a_{max} T \frac{z^2}{4}. \tag{8.10}$$

The Coulomb pump waveform was also implemented on one axis of the planar motor. Once again, the main problem was in finding a waveform that the motor followed reasonably well. One example is shown in Figure 8.8. Applying (8.10), the computed equilibrium velocity is 17.6 mm/s, while the observed velocity in Figure 8.8 is 4.5 mm/0.2 s, or 22.5 mm/s, a fairly close match.

The examples shown in Figure 8.6 and Figure 8.8 (which both use the same part and similar waveform frequencies and maximum accelerations) show that the Coulomb pump waveform has a higher part velocity than the stick-slip waveform, but requires larger velocity and translations from the feeder. The larger forward accelerations of this waveform also appear to cause problems for sloped feed trays, as discussed in the following section.

8.3 Singulating parts with ramps

While the above waveforms result in part motion, nearby parts will tend to move as a group. Singulation of these parts requires something more. One technique for singulation is to somehow vary the feed rate of the parts based on their position. In particular, if the feed rate increases as the parts move from one region to another, the parts will tend to spread out. This technique is used in the dual conveyors of the Adept FlexFeeder [129], schematically shown in the top half of Figure 8.9, where one conveyor drops parts onto a faster conveyor. To achieve a similar effect for the proposed feeder, a sloped section may be added to the feeder tray. Assuming that parts will climb up the ramp, but at a slower rate than if they were on a flat surface, the parts on the flat plateau region at the top of the ramp will be singulated relative to the parts on the ramp section, as depicted in the bottom half of Figure 8.9. A model for parts motion on ramps is derived in this section that supports this conclusion.

For this case, the part dynamics are given by inspection of the free-body diagram in Figure 8.10:

$$m\ddot{y}_p = f_f - mg\sin(\varphi) \tag{8.11}$$

$$m\ddot{z}_p = f_N - mg\cos(\varphi), \tag{8.12}$$

where $\ddot{y}_p$ and $\ddot{z}_p$ are the part accelerations in the j and k directions defined in Figure 8.10. Assuming a Coulomb friction model and sliding contact between the part and ramp, there are two additional constraints:

$$f_f = -\mu f_N, \quad \text{and} \tag{8.13}$$

$$\ddot{z}_p = -\ddot{y}_t\sin(\varphi), \tag{8.14}$$

where $\ddot{y}_t$ is the horizontal tray acceleration.

Solving the above equations for f_N, $\ddot{y}_p$, and $\ddot{z}_p$ gives:

$$f_N = mg\cos(\varphi) - m\ddot{y}_t\sin(\varphi) \tag{8.15}$$

$$\ddot{y}_p = -g\sin(\varphi) - \mu g\cos(\varphi) + \mu\ddot{y}_t\sin(\varphi) \tag{8.16}$$

$$\ddot{z}_p = -\ddot{y}_t\sin(\varphi). \tag{8.17}$$

Note that $\ddot{y}_p$, the part acceleration, is now a function of the tray acceleration, which changes during the slip phase, complicating the part motion. For this reason, an

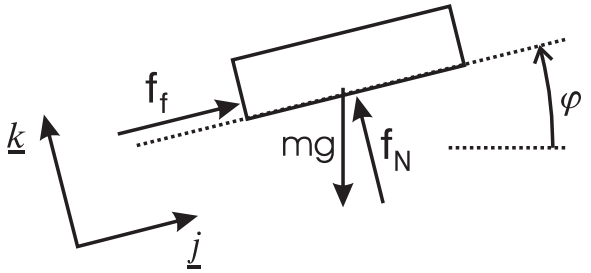


Figure 8.10: Free body diagram for part on ramp

analytic solution of the part motion is less informative than for the flat tray case, and the dynamic equations are instead used to simulate the part motion.

The part motion was simulated for a range of friction coefficients and ramp angles, with results shown in Figure 8.11. It is interesting to note that for some of the cases represented in this plot, the stick assumption does not hold, but parts still move forward. More importantly, the feed rate for parts on ramps is smaller than for parts on a flat surface, suggesting that ramps will be effective for singulation. The left half of Figure 8.13 shows an example of how the motion of a part on a sloped feed tray compares with that of a part on a flat feed tray.

Simulations of the Coulomb pump waveform for the sloped feed tray were also performed, with results shown in Figure 8.12. Here, the part velocity dropped off faster with increasing slope than for the stick-slip waveform. Note that the part velocity drops below zero at about 3°, limiting this waveform to gradual slopes. The reason for this effect is difficult to pinpoint, but appears to be related to the larger positive accelerations of the tray. Referring to the right half of Figure 8.13, when slope of the part velocity increases from $\overline{ab}$ to $\overline{cd}$, the steep slope of the tray velocity $\overline{de}$ causes their intersection to move more negative than if the slope of the tray velocity was less steep. In particular, note that $\Delta v_{cp} > \Delta v_{ss}$, indicating that the slope change has a larger effect on the Coulomb pump waveform. Although such a change is useful for singulation, in this case it is too extreme and will limit the feasible ramp angles too severely.

8.4 A miniature mobile parts feeder

In the complete feeder design (Figure 8.3), the above motion and singulation techniques are combined with a parts recirculation strategy, chosen for the reasons cited

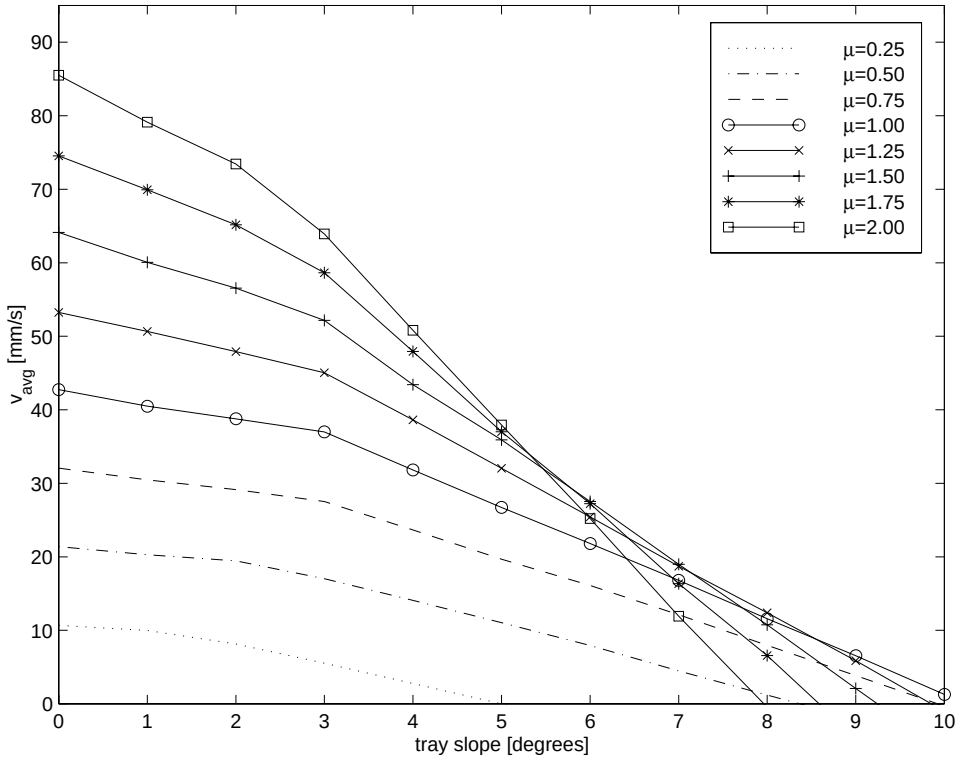


Figure 8.11: Feed rates as a function of tray slope and coefficient of friction for stick-slip waveform, with $a_{max} = 4\mu g$, $a_{min} = 0.9\mu g$, and $T = 1/30$ s. Parts feed faster on flat surfaces than up ramps.

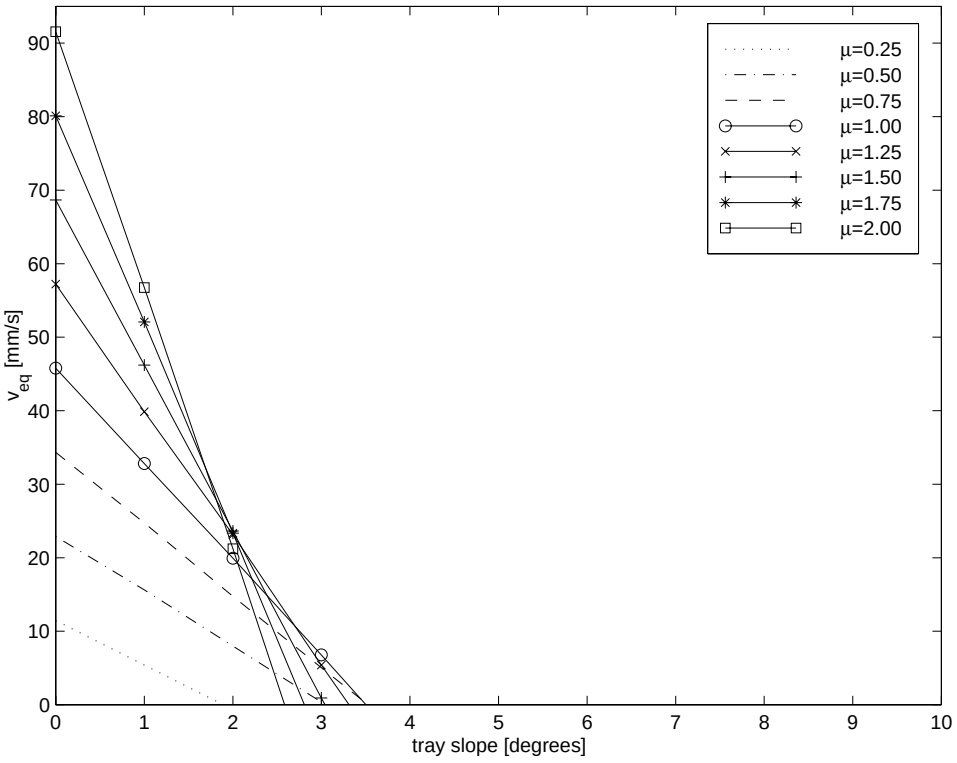


Figure 8.12: Feed rates as a function of tray slope and coefficient of friction for Coulomb pump waveform, with $a_{max} = 4\mu g$ and $T = 1/30$ s.

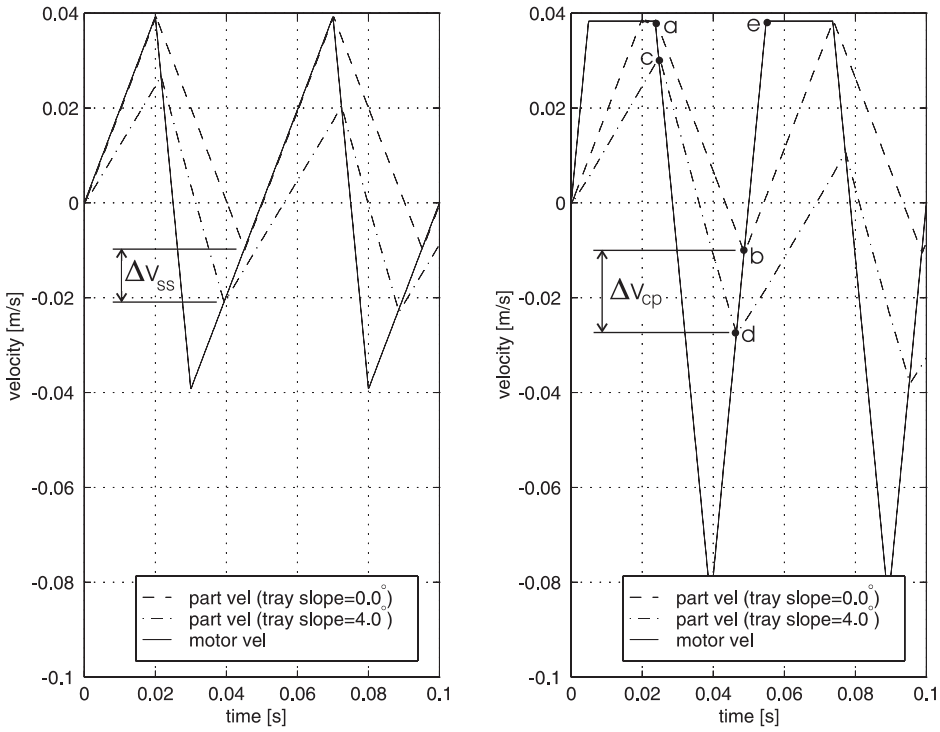


Figure 8.13: Part waveforms change based on the slope of the feed tray for both the stick-slip (left) and Coulomb pump (right) waveforms.

at the beginning of this chapter. The feeder has an annular feed path so that a single rotational vibration waveform of the feeder will suffice to keep the parts flowing around the loop. A ramp is used, as discussed above, to singulate parts along the direction of feeding. Two effects were expected to provide singulation radially across the feed path. First, the parts were expected to move locally along tangents, which would tend to make them collect towards the outer wall of the feed tray. In addition, the ramp section does not have a constant slope, but gets steeper with decreasing radius. If the waveform is selected carefully, it could be possible to cause parts to slide back down the ramp for small radii, but still climb at larger radii. This technique would allow for a variable-width feeding region at the outer wall of the feed tray. Simulation results below examine these effects.

Once singulated and in the plateau region, an overhead vision system can be used to detect parts in the correct orientation. The feeder can then move[2] to deliver the parts to an overhead manipulator for assembly.

Incorrectly oriented parts rejected by the vision system continue around the feed path and pass over the dropoff (a nice side-effect of the ramp section), which allows for a reorientation. This effect allows for reorienting parts out of the plane, despite a feeder that is restricted to planar motions. Depending on the part materials, part size and shape, and dropoff height, the part might always just flip over or might assume a more random orientation change.

Although the proposed feeder physically resembles a vibratory bowl feeder, it is important to note the major differences. First, the proposed feeder design is intended for use where feeder bowls custom-designed for parts would be impractical because of re-use requirements or long lead times, and where required feed rates are modest. Conceptually, the feeder operation actually resembles the Adept FlexFeeder (with the ramp singulation replacing the double conveyors and both using vision for parts selection) more than a bowl feeder. Second, the use of planar motor technology for generation of vibrations allows the feeder to be compact and mobile, permitting it to deliver parts directly to the assembly location, which is especially important for the minifactory application.

[2]Of course, it must move with small enough accelerations that the parts don't slide relative to the feed tray.

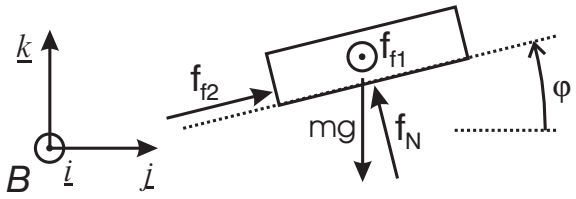


Figure 8.14: Local free body diagram for a part on an annular tray. ($\odot$ denotes a vector pointing out of the page.)

8.4.1 Dynamic model

In this section, a dynamic model is derived for a part on the feed tray discussed in the previous section. The part is assumed to be a point mass that stays in contact with the feed tray, and a Coulomb friction model is assumed. A free-body diagram of the part on the ramped section of the tray is shown in Figure 8.14. Coordinate frame B is an inertial (not accelerating) reference frame aligned with the part *at this particular instant in time*.[3] The dynamics of both a sticking case and slipping case must be considered. The dynamics problem is to determine the acceleration of the part along the tray given the part and tray positions and velocities.

By inspection of the free-body diagram, the part dynamics are given by:

$$\ddot{x} = f_{f1}/m, \tag{8.18}$$

$$\ddot{y} = f_{f2}\cos(\varphi)/m - f_N sin(\varphi)/m, \tag{8.19}$$

$$\ddot{z} = f_{f2}\sin(\varphi)/m + f_N cos(\varphi)/m - g, \tag{8.20}$$

where $f_{f1,2}$, f_N, g, and φ are defined in Figure 8.14.

For the slipping case, Coulomb's law gives:

$$f_f = \begin{bmatrix} f_{f1} \\ f_{f2}\cos(\varphi) \\ f_{f2}\sin(\varphi) \end{bmatrix} = \mu f_N \frac{v_p - v_t}{\|v_p - v_t\|}, \tag{8.21}$$

where v_p is the part velocity and v_t is the local tray velocity, both expressed in coordinate frame B.

Assuming the part stays on the surface of the feed tray, there is an additional kinematic constraint on the part acceleration:

$$\ddot{z} = \tan(\varphi)(\ddot{y} - 2\dot{r}\dot{\theta} - r\ddot{\theta}_t), \tag{8.22}$$

[3] In the simulation, each time step uses a different frame B.

where r and θ give the part position in polar coordinates relative to the center of the feeder, and θ_t is the rotation angle of the feeder.

To solve for f_N, (8.19) and (8.20) are substituted into (8.22), giving after simplification:

$$f_N = mg\cos(\varphi) - 2m\sin(\varphi)\dot{r}\dot{\theta} - m\sin(\varphi)r\ddot{\theta}_t. \tag{8.23}$$

This result for f_N can then be substituted into (8.21) to get $f_{f1,2}$, and (8.18)-(8.20) to compute the part acceleration for the slipping case at a given instant in time. Because coordinate frame B is only valid at an instant in time, the acceleration vector must be transformed to a coordinate frame fixed to the workspace before integration.

For the sticking case, the part is fixed relative to the feed tray, and the part accelerations are given based on the tray motion and part position:

$$\ddot{x} = -r\dot{\theta}_t^2 \tag{8.24}$$

$$\ddot{y} = r\ddot{\theta}_t \tag{8.25}$$

$$\ddot{z} = 0, \tag{8.26}$$

To check whether the friction forces are sufficient to keep the part stuck to the feed tray, these acceleration values are substituted into (8.18)-(8.20), which can be solved for f_N and $f_{f1,2}$:

$$f_N = -mr\ddot{\theta}_t\sin(\varphi) + mg\cos(\varphi) \tag{8.27}$$

$$f_{f1} = -mr\dot{\theta}_t^2 \tag{8.28}$$

$$f_{f2} = mr\ddot{\theta}_t\cos(\varphi) - mg\sin(\varphi). \tag{8.29}$$

Both the sticking and sliding equations are evaluated every simulation iteration. The sticking mode results are used if $\|[f_{f1}\ f_{f2}]^T\| < \mu f_N$ (which indicates that the friction forces are sufficient to maintain sticking), and the relative velocity $\|v_p - v_t\|$ is below some threshold. Otherwise, the sliding mode results are used. For both cases, the sign of f_N is examined to be sure the part does not leave the tray surface.

8.4.2 Simulation results

In this section, simulation results are presented for a feeder with an outer radius of 90 mm, an inner radius of 30 mm, and a 180° ramp with a total height change of

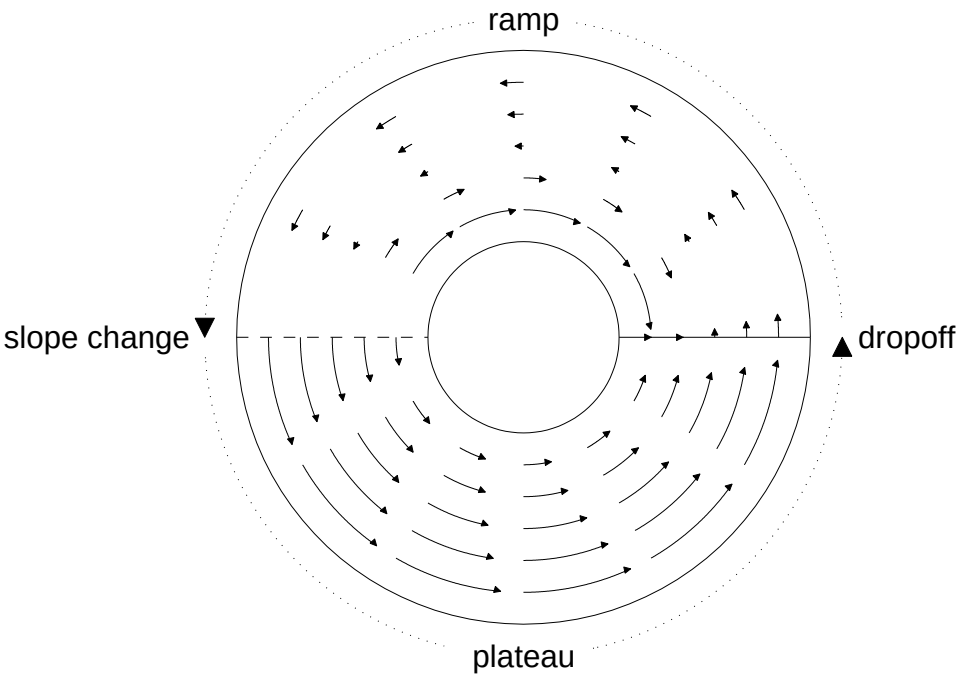


Figure 8.15: Simulation results show parts climbing the curved ramp, and singulation occurring in the radial and circumferential directions.

20 mm. The part mass is chosen as 20 g, with a coefficient of friction between the part and tray of $\mu = 0.2$. The tray is set to follow a rotational stick-slip waveform, with parameters $T = 1/30$ s, $a_{min} = 9.0\mu g$ rad/s^2, and $a_{max} = 66.7\mu g$ rad/s^2. The initial position of the part is set to one of a polar array of positions on the tray, with zero initial velocity. For each starting position, the part motion is simulated for five seconds while recording the sequence of part positions.

Results for each part starting position are shown in Figure 8.15. Note that parts only feed up the ramp if they are close to the outer radius of the feeder, with retrograde motion of parts near the center. This feed pattern should cause the parts to feed along a narrow annulus along the outside of the feeder, singulating them reasonably well in the radial direction. The part velocity on the plateau section is much faster than that on the ramp, suggesting that singulation along the direction of motion should also work well. It is surprising that the parts move along nearly perfect arcs, instead of veering off in more of a tangent direction. This effect can be understood by noting that the parts move in a series of small incremental steps that closely approximate a circle. Also, sticking resets the radial velocity of the part to

zero every cycle so that radial motions do not build up. A depiction of the parts flow in the feeder loosely based on the these simulation results is shown in Figure 8.3.

A simulation of the feeder using the Coulomb pump waveform was also attempted. However, it was difficult to get the parts to climb the ramp except for very gradual slopes, with 5 mm or less rise over the 180° ramp. This result was not surprising given this waveform's strong sensitivity to slope, as noted in Section 8.3.

8.5 Discussion

Open problems include choosing the waveform parameters given a particular feed tray and part, and designing a feed tray for a family of parts given the limitations of the planar motor actuators. In addition, waveforms other than the stick-slip or Coulomb pump can be considered. A hybrid waveform that combines their advantages or the formulation of an optimal waveform would improve performance. Reznik [104] has suggested a waveform composed of two sinusoids to limit the frequency spectrum that must be tracked by the controller.

There are a number of potential limitations of this feeder. Parts must be stable enough in their pickup orientation to survive the trip up the ramp and the vibrations without falling over. Parts that tend to nest will probably not be singulated properly. Parts may become stuck on the transition between the ramp and plateau, even if it is rounded. Feed rates may be too slow to be useful in a real automated assembly system.

A simple model of a spiral feed tray, shown in Figure 8.16, was constructed out of cardboard, but parts did not feed properly. Excessive vibration of the cardboard tray and the imprecise angle of the fabricated ramp prevented proper feeding. Construction of a precision fabricated rigid feed tray is required to completely demonstrate successful operation of the complete feeder, but the experiments in this chapter indicate the main idea that part motions from planar vibrations can be used for inducing useful motion in parts. This chapter is also intended to show that low-DOF robots with sufficient performance can be used for performing interesting manipulation tasks, though it may require some ingenuity to discover exactly how. The feeder of this chapter exploits the high-bandwidth of the planar robot. Other interesting manipulation techniques may be made possible by exploiting the dynamics, large motion capability, high-payload, and/or high-precision of the robot.

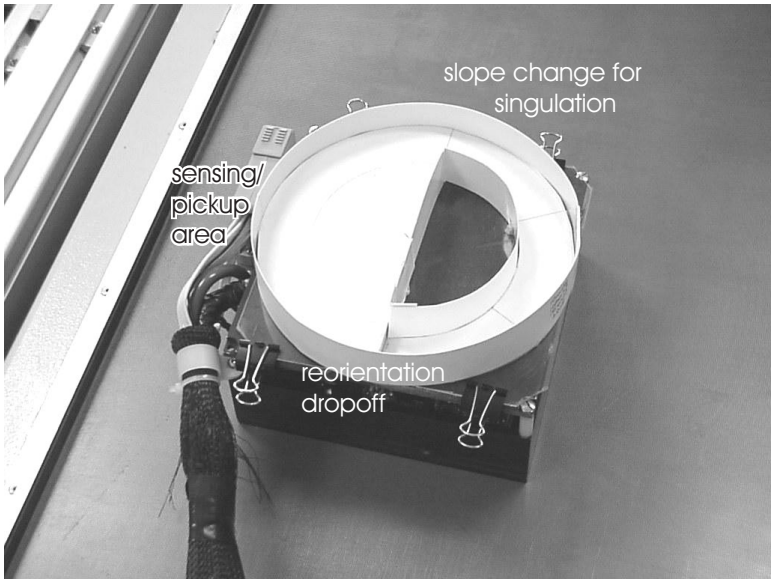


Figure 8.16: Prototype tray for spiral parts feeder

Chapter 9

Conclusions

9.1 Advantages of low-DOF closed-loop operation

This dissertation has presented a planar robot that combines the precision of a microscope stage with the speed, acceleration, and workspace of a direct drive manipulator arm. Some of these properties are inherent in the Sawyer motor technology, and can be exploited even with the open-loop stepping operation mode used in industry. However, such usage is only possible in repetitive, structured situations. Open-loop operation is fragile. Position sequences that are too fast or not smooth enough or that excite the underdamped dynamics of the system will not be tracked effectively and will lead to a fault condition. This fault condition may require a re-registration of the position of the device or operator intervention to realign and reposition the robot. The fault may not even be immediately detected by the system, which in systems with multiple robots may lead to cascading failures. The success of open-loop operation in industry has been predicated on the use of robots as programmable positioning devices that, once programmed, have very little interaction with their environments.

Even for pure positioning tasks, environmental interaction becomes important at the limits of the performance envelope. Small disturbances caused by vibrations from other robots, the magnetic state of the platen, and tether motions limit the precision of open-loop operation, and can lead to serious quality problems as the required precision for an operation approaches the limits of the device. Failed operations may not be detected until much later in the assembly process. Closed-loop operation permits higher precision via disturbance rejection, but also allows an operation to be

dynamically monitored. Even if an operation fails it can be immediately detected, minimizing the costs of the failure.

During motions, the tether generates significant disturbances that may be difficult to model, especially in multi-robot systems. Open-loop operation assumes that the tether disturbances will not change significantly from cycle to cycle and requires conservative design of trajectories such that tether disturbances will not cause failures. As the required motions becomes less structured, the trajectories must become more conservative to account for the wider possible range of tether disturbance. In contrast, in closed-loop operation, the controller can move as fast as possible to the goal subject to the actual disturbance forces it experiences.

Closed-loop operation also enables robust dynamic interactions with the environment. Instead of simple, carefully planned positioning tasks, the planar robot can have rich interactions with its environment. Tasks of this sort might involve direct physical contact, such as the exploration of platen boundaries, force-guided assembly, or collision detection, or might involve virtual couplings, such as visual servoing or coordination of multiple low-DOF robots via network communications. These kinds of tasks are especially important in overcoming the limited DOFs of the planar robot by allowing multiple low-DOF robots to cooperatively perform high-DOF operations. The key advantage of closed-loop operation in this case is the ability to respond dynamically to unstructured sources of information, enabling simple, robust methods for cooperative operations.

A number of experiments have demonstrated the superiority of closed-loop operation for various tasks. For precision positioning, the experiments of Chapter 5 showed an improvement in repeatability from 6 μm to 2 μm for motions larger than 20 mm and an improvement to sub-micron levels for motions smaller than 20 mm. Repeatability on a longer timescale will also be improved by the dramatic decrease in power dissipation, with the power usage for a typical pattern of moves dropping by more than 75%.

Closed-loop operation for contact tasks was demonstrated with the wall-following operation of Chapter 7, which allows for a quick, robust startup procedure for the robot to characterize its environment. The same controller was used for both free-space motions and wall contact, minimizing instabilities during the transition between these modes of operation. Internal and external corners were also immediately identified and localized in a quick and robust manner. This experiment demonstrates

that the robot can be used for force and impedance tasks without the need for an extrinsic force sensor.

The use of closed-loop operation for coordination between multiple low-DOF robots was illustrated using the visual servoing applications of Chapter 7. Under closed-loop operation, the planar robot can track overhead beacons moving faster than 1 m/s and with camera-based visual servoing can position parts to 14 μm (1 σ) even while moving at 5 mm/s.

9.2 Contributions

```
Your existence has a positive
contribution to mankind.
Lucky Numbers 5, 11, 16, 20, 28, 38
```

The high-level contribution of this dissertation is the demonstration that certain low-DOF robots can be used to provide high-performance manipulation capabilities. Their simplicity permits great improvements in performance, which for many tasks is more important than the number of DOFs. For certain tasks that do require higher DOFs, multiple low-DOF robots can be coordinated in a variety of ways to perform the task without sacrificing the performance gains from their low-DOF designs.

There are additional contributions that may be of interest to those who are interested in using the specific technology of the experimental system or in systems with similar configurations:

- The first published 3-DOF closed-loop operation of a Sawyer motor with integral position sensing.
- A thorough and well-documented re-examination of the modeling of Sawyer motors for use with closed-loop operation.
- Techniques for autonomous calibration of both the planar robot actuators and sensors, identifying redundancy as an enabler for autonomous calibration, and solving the problem of how to provide for well-calibrated devices while still allowing users to use multiple platens and forcers together in any combination.
- A demonstration of the advantages of closed-loop operation of planar motors, with an emphasis on the importance of new modes of operation that broaden the classes of applications.

There are also contributions related to applications of the planar robot:

- Development of a critical component of the minifactory system, which requires a high-performance and trustworthy robotic agent for product transport and cooperative assembly tasks.
- The invention of a novel miniature mobile parts feeder, and the first time to my knowledge that mobility has been identified as a useful feature for a parts feeder.

9.3 Planar robot design

This dissertation documents the first uses of the planar robot designed at CMU. There are several design issues that were impossible to resolve except after developing and using the prototype systems. Although the experiments of this dissertation were not designed specifically to explore these issues, the work did yield insights that are useful for validating some of the design decisions and providing insights for future designs.

The CMU platen sensor [22] is unusually small compared to other prototypes that have been introduced [26, 40, 96]. It also is unusual in that all four sensor segments are combined on a single substrate and integrated into the center of the forcer. Other sensor prototypes have attached to the side of the forcer body.

The small size was expected to have a number of advantages. It can be integrated into typical forcer designs without increasing the footprint of the forcer, making more efficient use of the platen area. It simplifies manufacturing, with only one extra part that must be aligned with the rest of the forcer. The small size also allows for high-frequency operation and a single shield to protect all four sensor segments from the fields of the actuators.

However, disadvantages to the small design were also expected. Each sensor segment only covers a few platen teeth, making them more sensitive to platen teeth irregularities. When computing the angle by differencing two of the sensor segments, there is a very short baseline (d_s in Figure 1.8) because of the compact sensor design. This effect increases both the noise and inaccuracy of the skew angle measurement, and creates a significant *sweet-spot* in the middle of the forcer where the positioning resolution will be better.

Although the advantages of the small sensor design were demonstrated experimentally on early prototypes, the disadvantages are more system-level and could not be easily determined until final system integration. The experiments presented in this dissertation have shown that the disadvantages of the sensor are not overly problematic. The sweet-spot resolution was examined experimentally in Chapter 5, showing that the robot can achieve a 1 μm (1σ) positioning resolution over the entire forcer body, which was the design target. Originally, the decrease in sensor accuracy was not expected to be a problem because for very high precisions, some amount of endpoint sensing would normally be required, because although a precision operation

may be required for a particular part, dimensions of the sub-assembly are probably not known to as high a precision. With a sensor positioning accuracy of around ten microns, this is certainly the case. However, what was not fully appreciated was the importance of the angular accuracy of the sensor. While few tasks require the robot to rotate to better than the milli-radian angular accuracy of the sensor, during motions the inaccuracy of the sensor acts is a disturbance to the angle servo with a frequency proportional to the velocity of the forcer. This disturbance causes trouble during high-stiffness motions, and necessitated the use of the non-linear angular position gains for certain controllers such as that of (7.3). The angular accuracy also causes the positioning accuracy away from the center of the forcer to be several times worse, so that there is also a *sweet-spot* for accuracy that could require endpoint sensing in cases where it would not ordinarily be required.

Most commercial planar motor systems use dedicated electronics to perform the commutation. The commutation and control systems in this dissertation were implemented in software running on real-time operating systems. Software-based commutation was chosen to ease the development of controllers and commutators, and to allow dynamic changes in the operational characteristics of the device. Although with modern computing hardware the software commutation was able to run at 5 kHz rates, this frequency is still substantially slower than the rates possible with dedicated hardware. However, this frequency level was expected to be sufficiently high.

Early experiments, documented in Chapter 2 showed that the lower commutation rates significantly reduced the speed capability of the planar robot. However, the relatively simple phase-advance strategy presented there can compensate for most of this effect. The lower commutation and control rates cause additional audible noise during fast motions, limit the bandwidth of the controller, and may also be limiting the fidelity of the dynamic force modeling experiments of Chapter 3, but these possible problems are outweighed by the convenience of the software-based development environment.

9.4 Future work

```
There will plenty of time to work
      hard, enjoy yourself!
Lucky Numbers 8, 15, 16, 20, 21, 39
```

The work described in this dissertation can be extended in many different directions. There are several obvious short-term extensions, many of which have been mentioned elsewhere in this text. The autonomous calibration work can be repeated using new couriers and platens under development in the Microdynamic Systems Laboratory. Prototypes of these platens and couriers have much more uniform teeth than the Normag systems, so the autonomous calibration results for both the sensors and actuators should be significantly better.

A design study of tetherless couriers is also underway in the Microdynamic Systems Laboratory. Eliminating the tether should improve the actuator autonomous calibration results by eliminating the dominant disturbance during the data collection procedure. Note that the power usage experiments of Chapter 7 show that closed-loop operation is crucial for reducing the power requirements of the planar robot for tetherless operation.

The coordination sensor cooperative autonomous calibration of Section 4.2 could also be revisited after calibrating the internal parameters of the coordination sensor using a benchtop setup. This should allow improved on-site calibration of the platen sensor using a sensor that is already available on the minifactory couriers.

The improvements to the hybrid controller suggested in Chapter 6 can be implemented to eliminate the chattering and the sensitivity to angle inaccuracies. Design aids and heuristics for choosing the hybrid controller parameters would also be helpful.

Complete verification of the mobile parts feeder application requires a precision fabricated spiral feed tray to eliminate feeding problems caused by tray vibrations and imprecise ramp angles. The feed tray can also include backlighting to aid visual identification of the parts prior to pickup, and vacuum chucks to keep the parts in the pickup area from moving when the planar robot moves across the platen. This vibration method is especially compatible with traps [132] because the part never loses contact with the feed surface, unlike typical vibratory parts feeders where parts bounce as they feed.

Much of the work of this dissertation also applies to higher-DOF robots that use planar motor technology, such as the robotic positioning heads of the RobotWorld and Megamation workcells, provided they are augmented with appropriate position sensors. However, the performance will be reduced by the extra mass and relatively poor precision of the added DOFs. Other types of robots can also benefit from the autonomous calibration and control results. Examples include underwater [133, 134] and simulated space [135, 136] robots that have multiple fixed-direction thrusters mounted on a single rigid body.

There are also several unexplored uses for the planar robots. One idea is that instead of carrying subproducts, planar robots can be used to cooperatively manipulate objects sitting directly on the smooth platen surface (or sitting on appropriate low-friction, passive pucks). This system would be a smaller-scale, higher-performance version of cooperative manipulation using traditional mobile robots [11, 137]. The manipulation can be performed by pushing, ropes, or docking mechanisms. This approach would eliminate the need to hand-off subassemblies when the couriers of the minifactory reach the ends of their tethers, but also allows the subassemblies to be dynamically "passed" to other couriers, even if they do not share a common workspace.

Another unexplored application is using planar robots to automate product transfer in human-stationed assembly lines or hybrid assembly lines that have both manual and automated stations. Often, rapidly-deployable assembly dictates manual assembly. In this case, planar robots can provide a programmable, intelligent system for product transfer. Planar robots are very compatible with humans. The small size and small moving mass of the planar robots make them very safe. The table-top platen configuration also allow for ergonomic seating or standing positions for human operators, depending on the mounting height of the platen. The disturbance estimator provides enough haptic ability for simple human-robot interaction. During the course of this work, it was discovered that by adding a high-frequency waveform as a disturbance to either the open-loop or closed-loop position commands causes the forcer to emit a loud audible noise. The reproduction of the waveform is good enough that digitized speech that is generated in this manner is understandable. This audio capability allows another channel of communication from the robot to the human. In an example scenario, an assembly-line worker would be sitting in

front of the platen with appropriate tools hanging from above to perform any of several operations. A courier would drive to her station, verbally describe the operation needed for the product, and adjust its impedance appropriately to aid in the task. The worker would perform the operation, and then push the courier away, which it would detect using either force or position information, and then proceed to the next automated or manual station. This usage of planar robots could be very important if a product requires a few assembly operations that are difficult to automate in a timely or economic manner.

There are probably many other applications for planar robots. The work presented in this dissertation will be most successful if it informs and inspires others to invent and implement new, interesting, and useful applications for this technology.

Reference List

[1] B. Carlisle, "Robot mechanisms," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, (San Francisco, CA), pp. 701–708, April 2000.

[2] J. K. Salisbury and J. J. Craig, "Articulated hands: Force control and kinematic issues," *International Journal of Robotics Research*, vol. 1, no. 1, pp. 4–17, 1982.

[3] S. C. Jacobsen, E. K. Iversen, D. F. Knutti, R. T. Johnson, and K. B. Biggers, "Design of the Utah/MIT dextrous hand," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, (San Francisco, CA), pp. 1520–1532, April 1986.

[4] R. L. Hollis, S. E. Salcudean, and A. P. Allen, "A six-degree-of-freedom magnetically levitated variable compliance fine-motion wrist: Design, modeling, and control," *IEEE Transactions on Robotics and Automation*, vol. 7, pp. 320–332, June 1991.

[5] A. Sharon, N. Hogan, and D. E. Hardt, "The macro/micro manipulator: an improved architecture for robot control," *Robotics and Computer-Integrated Manufacturing*, vol. 10, no. 3, pp. 209–222, 1993.

[6] H. Makino and N. Furuya, "Selective compliance assembly robot arm," in *Proceedings of the First International Conference on Assembly Automation*, pp. 77–86, 1980.

[7] J. Hauser and R. M. Murray, "Nonlinear controllers for non-integrable systems: the acrobot example," in *Proceedings of the 1990 American Control Conference*, (San Diego, CA), pp. 669–671, May 1990.

[8] M. W. Spong and D. J. Block, "The Pendubot: a mechatronic system for control research and education," in *Proceedings of the 34th IEEE Conference on Decision and Control*, (New Orleans, LA), pp. 555–556, December 1995.

[9] N. Hogan, "Stable execution of contact tasks using impedence control," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, (Raleigh, NC), pp. 1047–1054, March 1987.

[10] A. Bicchi and K. Goldberg, eds., *Minimalism in Robot Manipulation, Workshop WS1, IEEE Int'l Conf. on Robotics and Automation*, April 1996.

[11] K.-F. Böhringer, R. Brown, B. Donald, J. Jennings, and D. Rus, "Distributed robotic manipulation: experiments in minimalism," in *Experimental Robots IV: the 4th International Symposium*, (Stanford, CA), pp. 11–25, July 1995.

[12] S. Akella, W. H. Huang, K. M. Lynch, and M. T. Mason, "Sensorless parts orienting with a one-joint manipulator," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, vol. 3, pp. 2383–2390, April 1997.

[13] D. Reznik and J. Canny, "A flat rigid plate is a universal planar manipulator," in *Proc. Int'l Conf. on Intelligent Robots and Systems*, pp. 1471–1477, 1998.

[14] A. E. Quaid and R. L. Hollis, "Design and simulation of a miniature mobile parts feeder," in *Distributed Manipulation* (H. Choset and K. F. Böhringer, eds.), pp. 127–146, Kluwer, 2000.

[15] B. A. Sawyer, "Magnetic positioning device." U.S. Patent 3,376,578, April 2, 1968.

[16] B. A. Sawyer, "Linear magnetic drive system." U.S. Patent 3,735,231, May 22 1973.

[17] B. A. Sawyer, "Multi axes linear movement positioning system." U.S. Patent 3,836,835, September 17 1974.

[18] Northern Magnetics, Inc., "Dual axis high performance linear stepper motors." Linear motor sales catalog, rev. 1-96, Santa Clarita, CA 91355.

[19] V. Scheinman, "RobotWorld: a multiple robot vision guided assembly system," in *Robotics Research, the Fourth International Symposium* (R. Bolles and B. Roth, eds.), (Santa Cruz, CA), pp. 23–27, MIT Press, 1987.

[20] Megamation, "Speed and precision from novel assembly robot," *Assembly Automation*, vol. 9, pp. 85–87, May 1989.

[21] J. Payne, "RobotWorld—unrolled motors turn assembly on its head," *Industrial Robot*, vol. 20, no. 1, pp. 28–31, 1993.

[22] Z. J. Butler, A. A. Rizzi, and R. L. Hollis, "Integrated precision 3-DOF position sensor for planar linear motors," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, 1998.

[23] W. E. Hinds and M. Lewis, "Sensing and controlling the elements of a linear motor." U.S. Patent 5,880,541, March 9, 1999. Northern Magnetics, Inc.

[24] A. E. Brennemann, R. Hammer, R. L. Hollis, and J. Ish-Shalom, "Magnetic sensor for 2D linear stepper motor," *IBM Technical Disclosure Bulletin*, vol. 35, June 1992.

[25] K. Youcef-Toumi, D. Crawford, and F. Y. Wong, "Modelling and design of a sensor for two dimensional linear motors," tech. rep., Laboratory for Manufacturing and Productivity Report, Massachusetts Institute of Technology, Cambridge, Massachusetts, September 1994.

[26] J. Ish-Shalom, "Sawyer sensor for planar motion systems," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, (San Diego), pp. 2652–2658, May 1994.

[27] A. E. Brennemann and R. L. Hollis, "Magnetic and optical-fluorescence position sensing for planar linear motors," in *Proc. Int'l Conf. on Intelligent Robots and Systems*, vol. 3, pp. 101–107, August 1995.

[28] B. A. Sawyer, "Magnetic positioning device." U.S. Patent 3,457,482, July 22, 1969.

[29] B. A. Sawyer, "Control system for magnetic positioning device." U.S. Patent 4,009,428, February 22, 1977.

[30] B. A. Sawyer, "Linear motor." U.S. Patent 4,335,338, June 15, 1982. Xynetics, Inc.

[31] J. I. Nordquist and P. M. Smit, "A motion-control system for (linear) stepper motors," in *Proceedings of the Fourteenth Annual Symposium on Incremental Motion Control Systems and Devices*, pp. 215–231, 1985.

[32] J. I. Nordquist *et al.*, "Constant velocity systems using Sawyer linear motors," in *Proceedings of the Fifteenth Annual Symposium on Incremental Motion Control Systems and Devices*, 1986.

[33] E. R. Pelta, "Two-axis Sawyer motor for motion systems," *IEEE Control Systems magazine*, pp. 20–24, October 1987.

[34] E. R. Pelta, "Precise positioning without geartrains," *Machine Design*, pp. 79–83, April 1987.

[35] E. R. Pelta, "Method of moving head to correct for hysteresis." U.S. Patent 4,935,676, June 19 1990. General Signal Corporation.

[36] W. E. Hinds and B. Nocito, "15: The Sawyer linear motor," in *Theory and Application of Step Motors* (B. Kuo, ed.), pp. 327–340, St. Paul, West Publishing Co., 1974.

[37] L. M. McCarty, "Dual-axis linear motor precisely positions wafer probe," *Design news*, pp. 92–95, December 1, 1986.

[38] M. R. Genesereth, "DesignWorld," in *Proc. IEEE Int'l Symp. on Robotics and Automation*, pp. 2785–2788, April 1991.

[39] J. F. Canny and K. Y. Goldberg, "'RISC' for industrial robotics: recent results and open problems," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, (San Diego, CA), pp. 1951–1958, May 1994.

[40] E. J. Nicolson and et al., "Optical sensing for closed-loop control of linear stepper motors," in *Proc. Int'l Conf. on Advanced Mechatronics*, (Tokyo, Japan), August 1993.

[41] C. E. Lampson, "Incremental optical positioning sensor." U.S. Patent 5,818,039, October 6, 1998. Yaskawa, Inc.

[42] P. F. Lilenthal, J. P. Flemming, E. G. Harokopos, and G. C. V. Orden, "A flexible manufacturing workstation," *AT&T Technical Journal*, pp. 5–14, March/April 1988.

[43] J. Schneider and A. Kaufman, "Printed circuit robotics and transitioning technologies," *Printed Circuit Assembly*, vol. 1, no. 4, 1987.

[44] B. D. Hoffman and S. H. Pollack, "Robotic system." U.S. Patent 4,890,241, December 26, 1989. Megamation Incorporated.

[45] B. D. Hoffman and S. H. Pollack, "Method and apparatus for precision control of linear motors and the like employing a closed-loop technique." U.S. Patent 4,823,062, April 18, 1989. Megamation, Inc.

[46] B. D. Hoffman and S. H. Pollack, "Platen laminated in mutually perpendicular direction for use with linear motors and the like." U.S. Patent 4,835,424, May 30, 1989. Megamation Incorporated.

[47] S. H. Pollack, "Method and apparatus for monitoring and controlling linear motor robot apparatus and the like." U.S. Patent 5,081,593, August 16, 1989. Megamation Incorporated.

[48] T. D. Schmidt, "Umbilical release mechanism." U.S. Patent 5,136,196, June 4, 1991. Megamation Incorporated.

[49] T. D. Schmidt, "Method and apparatus for cooling hot spots in platen of linear motor system." U.S. Patent 5,138,206, June 4, 1992. Megamation Incorporated.

[50] L. Clark, "Fiberoptic encoder for linear motors and the like." U.S. Patent 5,324,934, June 28, 1994. Megamation Incorporated.

[51] T. D. Schmidt and P. Smit, "Linear motor surface coating system." U.S. Patent 5,340,400, August 23, 1994. Megamation Incorporated.

[52] B. D. Hoffman, S. H. Pollack, P. Smit, and J. Woolley, "Linear motor suspension system." U.S. Patent 5,352,946, October 4, 1994. Megamation Incorporated.

[53] F. Jacobs, "High resolution piggyback linear motor design for placement systems and the like." U.S. Patent 5,126,648, June 30, 1992. Megamation Incorporated.

[54] L. M. Clark, "Linear motor technology for SMD placement," in *5th International SAMPE Electronics conference*, (Los Angeles, CA), pp. 202–208, June 1991.

[55] B. D. Hoffman, "The use of 2-D linear motors in surface mount technology," in *Proc. 5th Int'l SAMPE Electronics Conference*, pp. 141–151, June 1991.

[56] Megamation, "Megamation WWW site." http://www.megamation.com, 2000.

[57] R. L. Hollis. Personal communication, July 2000.

[58] R. L. Hollis, J. Ish-Shalom, and E. J. Yarmchuk, "Position sensors for linear motors including plural symmetrical fluxes generated by a planar drive coil and received by planar sense coils being colinear along an axis of motion." U.S. Patent 5,434,504, July 18 1995.

[59] A. E. Brennemann, R. L. Hollis, J. Ish-Shalom, and W. V. Jecusco II, "An optical means of sensing position of a Sawyer motor on a magnetic grid surface," *IBM Technical Disclosure Bulletin*, vol. 37, pp. 375–378, May 1994.

[60] G. A. Fries, A. A. Rizzi, and R. L. Hollis, "Fluorescent dye based optical position sensing for planar linear motors," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, pp. 1614–1619, May 1999.

[61] A. E. Quaid, Y. Xu, and R. L. Hollis, "Force characterization and commutation of planar linear motors," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, pp. 1202–1207, April 1997.

[62] A. E. Quaid and R. L. Hollis, "3-DOF closed-loop control for planar linear motors," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, May 1998.

[63] A. E. Quaid and A. A. Rizzi, "Exploiting redundancy for autonomous calibration of a planar robot," in *Experimental Robots VI: the 6th International Symposium*, (Sydney, Australia), pp. 151–160, March 1999.

[64] R. L. Hollis and A. Quaid, "An architecture for agile assembly," in *American Society of Precision Engineering 10th Annual Meeting, Austin, Texas*, pp. 372–375, October 1995.

[65] A. E. Quaid and R. L. Hollis, "Cooperative 2-DOF robots for precision assembly," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, (Minneapolis, MN), April 1996.

[66] A. A. Rizzi, J. Gowdy, and R. L. Hollis, "Agile assembly architecture: an agent based approach to modular precision assembly systems," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, pp. 1511–1516, April 1997.

[67] P. F. Muir, A. A. Rizzi, and J. Gowdy, "Minifactory: a precision assembly system adaptable to the product life cycle," in *SPIE international symposium on intelligent systems and advanced manufacturing*, (Pittsburgh, PA), October 1997.

[68] R. L. Hollis and J. Gowdy, "Miniature factories for precision assembly," in *International workshop on microfactories*, (Tsukuba, Japan), pp. 9–14, December 1998.

[69] R. L. Hollis and J. Gowdy, "Miniature factories: tabletop assembly systems for mechatronic products," in *IARP micro robotics and systems conference*, (Beijing), October 1998.

[70] Z. J. Butler, A. A. Rizzi, and R. L. Hollis, "Contact sensor-based coverage of rectilinear environments," in *IEEE Int'l Symposium on Intelligent Control*, (Boston, MA), September, 1999.

[71] Z. J. Butler, A. A. Rizzi, and R. L. Hollis, "Complete distributed coverage of rectilinear environments," in *Workshop on the algorithmic foundations of robotics*, (Hanover, NH), March 2000.

[72] A. E. Quaid, "A miniature mobile parts feeder: Operating principles and simulation results," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, (Detroit, MI), pp. 2221–2226, May 1999.

[73] J. Ish-Shalom, "Sub-micron large motion multi-robot planar motion system," in *7th ISRR Conference*, 1995.

[74] J. Ish-Shalom, "Modeling of Sawyer planar sensor and motor dependence on planar yaw angle rotation," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, pp. 3499–3504, April 1997.

[75] J. Ish-Shalom and T. Wasfy, "A finite element model for real-time compensation of the thermal deformation of the platen of a planar step motor," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, pp. 1490–1495, May 1994.

[76] J. Ish-Shalom, "Composite magnetic structure for planar motors," *IEEE transactions on magnetics*, vol. 31, pp. 4077–4079, November 1995.

[77] H. Melkote, F. Khorrami, and J. Ish-Shalom, "Closed-loop control of a three degree-of-freedom ultra accurate linear stepper motor," in *Proc. 1997 IEEE Int'l Conference on control applications*, pp. 639–644, October 1997.

[78] H. Melkote and F. Khorrami, "Closed-loop control of a base xy stage with rotational degree-of-freedom for a high-speed ultra-accurate manufacturing system," in *Proc. of the 36th IEEE Conf. on Decision and Control*, pp. 2495–2500, Dec 1997.

[79] F. Khorrami, H. Melkote, and J. Ish-Shalom, "Advanced control system for high speed ultra accurate manufacturing system," in *Proc. American Control Conference*, pp. 164–165, June 1997.

[80] H. Melkote and F. Khorrami, "Closed-loop control of a base xy stage with rotational degree-of-freedom for a high-speed ultra-accurate manufacturing system," in *Proc. 1999 IEEE Int'l Conf. on Robotics and Automation*, pp. 1812–1817, May 1999.

[81] H. Melkote and F. Khorrami, "Nonlinear output feedback control for stepper motors: a robust adaptive approach," in *Proc. 1999 IEEE Int'l Conf. on Control Applications*, pp. 755–760, August 1999.

[82] H. Melkote and F. Khorrami, "Output feedback control of a three degree-of-freedom linear stepper motor with position measurements only," in *Proc. American Control Conference*, pp. 2531–5, June 1999.

[83] G. L. Miller, "Capacitively incremental position measurement and motion control." U.S. Patent 4,893,071, January 9 1990.

[84] G. L. Miller, "Capacitively commutated brushless DC servomotors." U.S. Patent 4,958,115, September 18 1990.

[85] Y. R. Bendaña, "Time-optimal motion control of two-dimensional linear stepper motors," Master's thesis, MIT, 1993.

[86] P. D. Gjeltema, "The design of a closed loop linear motor system," Master's thesis, MIT, June 1993.

[87] F. Y. Wong, "Inductive position/velocity sensor design and servo control of linear motors," Master's thesis, Massachusetts Institute of Technology, Oct. 1994.

[88] D. S. Crawford, "Sensor design and feedback motor control for two dimensional linear motors," Master's thesis, MIT, May 1995.

[89] S. J. Buckley, F. C. Park, and J. U. Korein, "Task planning for RobotWorld: a case study," in *Proc. IEEE Int'l Symp. on Intelligent Control*, pp. 348–353, September 1989.

[90] D. Parsons and J. Canny, "A motion planner for multiple mobile robots," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, pp. 8–13, May 1990.

[91] F. W. Sinden, "The tethered robot problem," *Int'l Journal of Robotics Research*, vol. 9, pp. 122–133, February 1990.

[92] G. Pardo-Castellote and H. Martins, "Real time motion scheduling for a SMALL workcell," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, (Sacramento, CA), pp. 810–817, April 1991.

[93] S. Hert and V. Lumelsky, "The ties that bind: motion planning for multiple tethered robots," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, (San Diego, CA), pp. 2734–2741, May 1994.

[94] F. Tahmasebi and L.-W. Tsai, "On the stiffness of a novel six-DOF parallel minimanipulator," in *Proceedings of the first world automation conference*, vol. 2, pp. 189–194, Aug. 1994.

[95] R. Ben-Horin, "A six-degrees-of-freedom parallel manipulator with three planarly actuated links," Master's thesis, Department of Mechanical Engineering, Technion – Israel Institute of Technology, Aug. 1994.

[96] F. Y. Wong, H. Schulze-Lauen, and K. Youcef-Toumi, "Modelling and digital servo control of a two-axis linear motor," in *Proc. American Control Conference*, pp. 3659–3663, June 1995.

[97] H. D. Chai, "6: permeance model and reluctance force between toothed structures," in *Theory and Application of Step Motors* (B. Kuo, ed.), pp. 141–153, St. Paul, West Publishing Co., 1974.

[98] D. J. Bennett and J. M. Hollerbach, "Autonomous calibration of single-loop closed kinematic chains formed by manipulators with passive endpoint constraints," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 5, pp. 597–606, 1991.

[99] R. M. Voyles and P. K. Khosla, "Collaborative calibration: Extending shape from motion calibration," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, (Albuquerque, NM), pp. 2795–2800, April 1997.

[100] A. A. Rizzi, "Hybrid control as a method for mobile robot programming," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, May 1998.

[101] A. E. Quaid and A. A. Rizzi, "Robust and efficient motion planning for a planar robot using hybrid control," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, (San Francisco, CA), pp. 4021–4026, April 2000.

[102] Z. J. Butler, *Distributed coverage of rectilinear environments.* PhD thesis, Carnegie Mellon University, Expected 2000.

[103] D. Reznik and J. Canny, "Analysis of part motion on a longitudinally vibrating plate," in *Proc. Int'l Conf. on Intelligent Robots and Systems*, 1997.

[104] D. Reznik and J. Canny, "The Coulomb pump: a novel parts feeding method using a horizontally-vibrating surface," in *Proc. Int'l Conf. on Intelligent Robots and Systems*, pp. 869–874, 1998.

[105] H. Choset and K. F. Böhringer, eds., *Distributed Manipulation.* Kluwer, 2000.

[106] Product of Wind River Systems, Alameda, CA.

[107] W.-C. J. Ma, "Precision optical coordination sensor for cooperative 2-DOF robots," Master's thesis, Carnegie Mellon University, Pittsburgh, PA, 1998.

[108] W.-C. Ma, A. A. Rizzi, and R. L. Hollis, "Optical coordination sensor for precision cooperating robots," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, (San Francisco, CA), pp. 1621–1626, April 2000.

[109] J. Ish-Shalom, "Experimental results of using a step motor as a spring," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, pp. 1841–1846, May 1989.

[110] P. F. Muir, "Converting a commercial electric direct-drive robot to operate from joint torque commands," Tech. Rep. SAND91-1548, UC-408, Sandia National Laboratories, 1991.

[111] H. D. Chai and A. C. Leenhouts, "Position error analysis for microstepping," in *Proceedings of the Fourteenth Annual Symposium on Incremental Motion Control Systems and Devices*, pp. 265–274, 1985.

[112] J. M. Hollerbach and D. M. Lokhurst, "Closed-loop kinematic calibration of the RSI 6-DOF hand controller," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, (Atlanta, GA), pp. 142–148, May 1993.

[113] K. Schroer, "Theory of kinematic modelling and numerical procedures for robot calibration," in *Robot Calibration* (R. Bernhardt and S. L. Albright, eds.), pp. 157–196, Chapman and Hall, 1993.

[114] J. M. Hollerbach and C. W. Wampler, "The calibration index and taxonomy for robot kinematic calibration methods," *International Journal of Robotics Research*, vol. 15, pp. 573–591, December 1996.

[115] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipies in C: The art of scientific computing*, ch. 15, pp. 656–706. Cambridge University Press, 2nd ed., 1992.

[116] M. Zefran and J. W. Burdick, "Design of switching controllers for systems with changing dynamics," in *Proceedings of the 37th IEEE Conference on Decision and Control*, (Tampa, FL), pp. 2113–2118, Dec. 1998.

[117] I. Kolmanovsky, E. G. Gilbert, N. H. McClamroch, and T. L. Maizenberg, "Toward less conservative designs of multimode controllers for systems with state and control constraints," in *Proceedings of the 37th IEEE Conference on Decision and Control*, (Tampa, FL), pp. 1647–1652, Dec. 1998.

[118] I. Kolmanovsky and E. G. Gilbert, "Multimode regulators for systems with state and control constraints and disturbance inputs," in *Control Using Logic-Based Switching*, (Block Island, RI), pp. 104–117, 1995.

[119] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek, "Sequential composition of dynamically dexterous robot behaviors," *International Journal of Robotics Research*, vol. 18, pp. 534–555, June 1999.

[120] J. P. LaSalle, "Some extensions of Liapunov's second method," *IRE Transactions on Circuit Theory*, pp. 520–527, December 1960.

[121] A. A. Rizzi and R. L. Hollis, "Opportunities for increased intelligence and autonomy in robotic systems for manufacturing," in *Robotics Research, the Eighth international symposium of robotics research* (Y. Shirai and S. Hirose, eds.), (Hayama, Japan), pp. 141–151, Springer-Verlag, 1998.

[122] M. L. Chen, S. Kume, A. A. Rizzi, and R. L. Hollis, "Visually guided coordination for distributed precision assembly," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, (San Francisco, CA), pp. 1651–1656, April 2000.

[123] G. Boothroyd, C. Poli, and L. E. Murch, *Vibratory Bowl Feeders*, ch. 3, pp. 27–49. Marcel Dekker, Inc., 1982.

[124] K.-F. Böhringer, V. Bhatt, and K. Y. Goldberg, "Sensorless manipulation using transverse vibrations of a plate," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, pp. 1989–1996, May 1995.

[125] N. C. Singer and W. P. Seering, "Utilizing dynamic stability to orient parts," *Journal of Applied Mechanics*, vol. 54, pp. 961–966, Dec. 1987.

[126] P. Swanson, R. Burridge, and D. Koditschek, "Global asymptotic stability of a passive juggler: A parts feeding strategy," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, (Nagoya, Japan), pp. 1983–1988, May 1995.

[127] J. Hollingum, "Sweeping it over the carpet," *Assembly Automation*, vol. 15, no. 3, pp. 29–30, 1995.

[128] A. Arban, "Adept parts feeder," *Assembly Automation*, vol. 15, no. 3, pp. 36–37, 1995.

[129] D. Gudmundsson and K. Goldberg, "Tuning robotic part feeder parameters to maximize throughput," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, pp. 2440–2445, April 1997.

[130] J. Krishnaswamy, M. J. Jakiela, and D. E. Whitney, "Mechanics of vibration-assisted entrapment with application to design," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, pp. 838–845, April 1996.

[131] A. E. Quaid, "A miniature mobile parts feeder: Operating principles and simulation results," Tech. Rep. CMU-RI-TR-98-26, The Robotics Institute, Carnegie Mellon University, 1998.

[132] R.-P. Berretty, K. Goldberg, M. H. Overmars, and A. F. van der Stappen, "Geometric trap design for automatic part feeders," in *Robotics Research, the Ninth International Symposium*, (Snowbird, UT), pp. 139–144, October 1999.

[133] S. K. Choi and J. Yuh, "Design of advanced underwater robotic vehicle and graphic workstation," in *IEEE Symp. on AUV Technology*, pp. 99–105, 1993.

[134] W.-C. Lam and T. Ura, "Non-linear controller with switched control law for tracking control of non-cruising AUV," in *IEEE Symp. on AUV Technology*, pp. 78–84, 1996.

[135] W. C. Dickson and R. H. Cannon, Jr., "Experimental results of two free-flying robots capturing and manipulating a free-flying object," in *Proc. Int'l Conf. on Intelligent Robots and Systems*, pp. 51–58, August 1995.

[136] W. C. Dickson, R. H. Cannon, Jr., and S. M. Rock, "A decentralized object impedance controller for object/robot-team systems: Theory and experiments," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, pp. 3589–3596, April 1997.

[137] B. R. Donald, L. Gariepy, and D. Rus, "Experiments in constrained prehensile manipulation: distributed manipulation with ropes," in *Distributed Manipulation* (H. Choset and K. F. Böhringer, eds.), pp. 29–47, Kluwer, 2000.

Appendix A

Controller output constraints

In this appendix, a method for constructing a convex force output envelope for a unipartite robot translating in $\mathbb{R}^n$ is presented. The robot is assumed to have m actuators, each capable of generating a force acting at a fixed position and direction relative to the robot body. Each actuator i is physically limited to output forces over the range $f_i^- \leq f_i \leq f_i^+,\ i \in \{1...m\}$ with $f_i^- < 0$ and $f_i^+ > 0$. Systems of this type include robots with electromagnetic forcers (planar robots), and robots with thrusters (space and underwater robots), provided the actuator limitations are reasonably modeled by the simple bounds given above. Although the force outputs of the actuators will generate torques on the robot body, it is assumed that the system trajectories keep the robot from rotating significantly. In this case, the mapping between the actuator forces and the output wrench is constant.[1] The constant angle assumption is not as limiting as it may first appear–often force saturation is only an issue during gross motions, in which case it is not unreasonable to maintain a fixed angle, varying it only at the start and end of the motion.

The motivation for this construction is to provide a mathematical description of the wrench output capabilities of the planar robot. This description can then used in design of controllers that will appropriately consider the output constraints. In particular, in Chapters 5 and 6, the output constraints are used to define the stable domain of the controllers in state space. This definition requires the controller force output to be a function of the state space variables. If there are controller

[1]Otherwise, the mapping will change with the orientation of the robot. If this is the case, one conservative approach is to consider the convex domains that result from each possible angle and use their intersection as the overall domain. This approach would be most effective when many linear actuators are distributed uniformly over the robot.

components that are functions of time, as with feed-forward terms for trajectory tracking, or components that have additional state, as with integral terms that seek to cancel disturbances, these components are assumed to be bounded and are considered separately, as detailed in Section A.2. First, a technique for computing the basic convex wrench output envelope is presented.

A.1 Convex wrench output envelope

Using static free-body analysis, it is easy to determine the linear mapping between the actuator forces and output wrench, of the form

$$w = J^T f, \tag{A.1}$$

where w is a wrench vector consisting of output forces and torques, J is the Jacobian, and f is a vector of actuator forces. To restate the problem, given bounds on the elements of f, what is the domain, $\mathcal{F}$, defined by $J^T f$? Note that on the boundary of $\mathcal{F}$ at least one of the actuator constraints must be active. However, the converse is not true–there may be f vectors that have active constraints that do not map to the boundary of $\mathcal{F}$. For example, there may be combinations of actuator outputs that result in internal forces and torques, but cancel each other to produce a zero net output wrench. This zero wrench will obviously be within the capabilities of the robot.

There is then a method to determine $\mathcal{F}$. Consider the set containing $3^m - 1$ vectors $f_{c,j}$ of length m with elements that are either zero or at the upper or lower limits. Each vector $f_{c,j}$ will be tested to determine whether it corresponds to a valid boundary in $\mathcal{F}$. Define a_j^- as the set of indices of elements that are set to their minimal values (from f^-), and a_j^+ as the set of indices of elements that are set to their maximal values (from f^+). Also define a square diagonal selection matrix S_j of size n with ones in the diagonal elements that correspond to indices in a_j^- or a_j^+, with zeros on all remaining elements.

Then, the condition for constraint combination j *not* to correspond to a surface of $\mathcal{F}$ can be expressed as

$$\text{rank}(S_j\ N) = \text{rank}(S_j\ [N\ f_{c,j}]), \tag{A.2}$$

which implies that the redundancy of the system (as encoded by N) can be used to find an alternate set of actuator forces that generate the same output wrench as $f_{c,j}$

without violating the constraint combination j. Otherwise, the wrench output cannot be generated without activating an actuator constraint and therefore constraint combination j is *valid* and will correspond to a boundary component of $\mathcal{F}$.

The boundary components of interest are $n-1$ dimensional hyperplanes in $\mathbb{R}^n$, but this technique will also produce intersections of these hyperplanes, edges, and points of $\mathcal{F}$ as constraints are added. Although they will not produce incorrect results, they are superfluous for the purposes of specifying the overall $\mathcal{F}$ convex domain and will lead to inefficiencies. Before continuing, the excess constraint combinations should be eliminated. A valid constraint combination j is *unnecessary* if there is another valid constraint combination j' such that

$$\begin{aligned} a_j^- \cap a_{j'}^- &= a_{j'}^-, \text{ and} \\ a_j^+ \cap a_{j'}^+ &= a_{j'}^+, \text{ and} \end{aligned} \tag{A.3}$$

Let the set $\mathcal{V} := \{\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_k\} \subset \{1 \ldots 3^m - 1\}$ contain the indices for the k necessary and valid constraint combinations.

After identifying the valid constraint combinations, the next task is to express them as constraint hyperplanes in $\mathbb{R}^n$. Note that a point $w_{0,j}$ on boundary component j of $\mathcal{F}$ is given by

$$w_{0,j} = J^T f_{c,j}, \tag{A.4}$$

but in general, this vector is not perpendicular to the boundary component. To find such a vector, it is necessary to eliminate any components of $w_{0,j}$ along the directions of freedom $d_{f,1}, d_{f,2}, \ldots d_{f,n-1}$ of the hyperplane, which are simply the columns of J^T corresponding to the $n-1$ unconstrained actuators for this constraint combination. This task is easier if orthogonal basis vectors $d'_{f,1}, d'_{f,2}, \ldots, d'_{f,n-1}$ for the hyperplane freedoms are first computed

$$d'_{f,1} = d_{f,1} \tag{A.5}$$

$$d'_{f,2} = \left(I - \frac{d'_{f,1} d'^T_{f,1}}{d'^T_{f,1} d'_{f,1}} \right) d_{f,2} \tag{A.6}$$

$$d'_{f,3} = \left(I - \frac{d'_{f,1} d'^T_{f,1}}{d'^T_{f,1} d'_{f,1}} - \frac{d'_{f,2} d'^T_{f,2}}{d'^T_{f,2} d'_{f,2}} \right) d_{f,3} \tag{A.7}$$

$$\cdots \tag{A.8}$$

$$d'_{f,k} = \left(I - \frac{d'_{f,1} d'^T_{f,1}}{d'^T_{f,1} d'_{f,1}} - \frac{d'_{f,2} d'^T_{f,2}}{d'^T_{f,2} d'_{f,2}} - \ldots - \frac{d'_{f,k-1} d'^T_{f,k-1}}{d'^T_{f,k-1} d'_{f,k-1}} \right) d_{f,k} \tag{A.9}$$

The normal to the hyperplane can then be computed as

$$w_{N,j} = \left(I - \frac{d'_1 d'^T_1}{d'^T_1 d'_1} - \frac{d'_2 d'^T_2}{d'^T_2 d'_2} - \ldots - \frac{d'_k d'^T_k}{d'^T_k d'_k} \right) w_{0,j}, \tag{A.10}$$

such that any vector w that lies in the hyperplane will satisfy the equation

$$1 = \frac{w^T_{N,j}}{w^T_{N,j} w_{N,j}} w := p^T_j w. \tag{A.11}$$

Together, these hyperplanes can be used to form a convex set in $\mathbb{R}^n$ that will include the origin. A norm-like function can then be defined

$$\Phi(w) = \max \left(p^T_{\mathcal{V}_1} w, p^T_{\mathcal{V}_2} w, \ldots, p^T_{\mathcal{V}_k} w \right), \tag{A.12}$$

where the indices of the valid necessary constraint combinations are encoded by the members of set $\mathcal{V}$. The domain $\mathcal{F}$ of achievable output wrenches can then be expressed as

$$\mathcal{F} := \{ w \in \mathbb{R}^n \mid \Phi(w) \leq 1 \}. \tag{A.13}$$

A.1.1 Planar robot example

For illustration, this technique is applied to the planar robot actuator configuration, with n=3 and m=4. For this case, the force mapping is given by

$$w = J^T f, \tag{A.14}$$

$$w = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ -d_a & d_a & -d_a & d_a \end{bmatrix} f, \tag{A.15}$$

where $w \in \mathbb{R}^3$, $f \in \mathbb{R}^4$, and d_a is as shown in Figure 1.8. The null-space of J^T is $N = [\ 1 \quad -1 \quad -1 \quad 1\ ]^T$. The force limits are assumed to be the same for all actuators, with $f_i^- = -f_{max}$, and $f_i^+ = f_{max}\ \forall i \in \{1 \ldots 4\}$.

Evaluation of condition (A.2) for all $3^m - 1 = 80$ possibilities yields 50 valid constraint combinations, listed in Table A.1. Elimination of unnecessary constraints based on condition (A.3) yields 12 valid and necessary constraints, denoted with an $*$ next to their indices. The above procedure is applied to these twelve $f_{c,j}$ vectors to compute a constraint surface vector $p_{\mathcal{V}_j}$ for each, as reported in Table A.2. The twelve constraint planes of the wrench output domain $\mathcal{F}$ will be a rhombic dodecahedron in $\mathbb{R}^3$, as shown in Figure 5.2.

A.2 Extension for time-varying controller terms

Assume a controller command

$$u_w = w_s(q, \dot{q}) + w_t(t, q'), \tag{A.16}$$

where w_s is a function consisting of system state-dependent terms and w_t is a function consisting of time-dependent terms (e.g. a feedforward acceleration for a trajectory) or terms that may have their own state (e.g. integral terms), contained in q'. Further assume that each element of w_t has known bounds that define the set $\mathcal{D}$

$$\{w_t \in I\!R^m | D^-_{t,i} < w_{t,i} < D^+_{t,i} \ \forall i = \{1, 2, 3\}\}. \tag{A.17}$$

The wrench output domain $\mathcal{F}$ derived in the previous section is used in Chapters 5 and 6 to define the provably stable domain of the controllers in state space. Only the state-dependent portion of the controller output w_s can be mapped to the state space. The other components of the controller output w_t must be treated differently, as they may assume any of the wrench vectors in set $\mathcal{D}$. It is necessary to find a smaller domain for w_s so that the overall controller output will not violate the actuator constraints. Implicitly define a domain $\mathcal{F}'$

$$\{w_s | w_s + w_t \in \mathcal{F} \ \forall w_t \in \mathcal{D}\} \tag{A.18}$$

This definition can also be expressed using a Minkowski sum

$$\mathcal{F}' \oplus \mathcal{D} = \mathcal{F}. \tag{A.19}$$

The original domain $\mathcal{F}$ must be reduced by $\mathcal{D}$, yielding the final domain $\mathcal{F}'$, which represents the set of w_s vectors that will not result in actuator saturation provided that $w_t \in \mathcal{D}$. Unlike the usual use of the Minkowski sum in robotics to grow obstacles based on the robot's footprint, here the domain must be "shrunk" based on the "footprint" of $\mathcal{D}$.

To find $\mathcal{F}'$, each face of $\mathcal{F}$ must be moved towards the origin based on the size of $\mathcal{D}$. From (A.12), each face of $\mathcal{F}$ is parameterized by $p_\mathcal{V} \in I\!R^m$ such that

$$p_\mathcal{V}^T w = 1. \tag{A.20}$$

The faces of $\mathcal{F}'$ will be defined similarly with

$$p_\mathcal{V}'^T w = 1. \tag{A.21}$$

To find a relation between the parameter vectors $p_{\mathcal{V}}$ and $p'_{\mathcal{V}}$, first note that from the definition of $\mathcal{D}$ it is easy to find the set of vertices of the boundary of $\mathcal{D}$, $\mathcal{D}_v = \{v_{\mathcal{D},1}, v_{\mathcal{D},1}, \ldots, v_{\mathcal{D},2^n}\}$. Then, the parameter vectors w'_N, j of $\mathcal{F}'$ can be computed as

$$w'_{N,j} = \min_{v \in \mathcal{D}_v} \left(1 - \frac{v^T w_{N,j}}{w_{N,j}^T w_{N,j}}\right) w_{N,j} \tag{A.22}$$

A vector w that lies on the boundary of $\mathcal{F}'$ will satisfy the equation

$$1 \quad = \quad \frac{w_{N,j}^{\prime T}}{w_{N,j}^{\prime T} w'_{N,j}} w := p_j^{\prime T} w. \tag{A.23}$$

As before, these hyperplanes can be used to form a convex set in $\mathbb{R}^n$ that will include the origin. A new norm-like function can then be defined

$$\Phi'(w) \quad = \quad \max\left(p_{\mathcal{V}_1}^{\prime T} w, p_{\mathcal{V}_2}^{\prime T} w, \ldots, p_{\mathcal{V}_k}^{\prime T} w\right), \tag{A.24}$$

where the indices of the valid necessary constraint combinations are encoded by the members of set $\mathcal{V}$. The domain $\mathcal{F}'$ of achievable output wrenches can then be expressed as

$$\mathcal{F}' \quad := \quad \{w \in \mathbb{R}^n \mid \Phi'(w) \leq 1\}. \tag{A.25}$$

For the planar motor case, the p' vectors computed using the above procedure are given in Table A.3.

j	$f^T_{c,j}$				a_j^-				a_j^+			
$*4$	f_1^+	f_2^+	0	0					1	2		
$*8$	f_1^-	f_2^-	0	0	1	2						
$*10$	f_1^+	0	f_3^+	0					1		3	
13	f_1^+	f_2^+	f_3^+	0					1	2	3	
$*15$	0	f_2^-	f_3^+	0		2					3	
16	f_1^+	f_2^-	f_3^+	0		2			1		3	
17	f_1^-	f_2^-	f_3^+	0	1	2					3	
$*20$	f_1^-	0	f_3^-	0	1		3					
$*21$	0	f_2^+	f_3^-	0			3			2		
22	f_1^+	f_2^+	f_3^-	0			3		1	2		
23	f_1^-	f_2^+	f_3^-	0	1		3			2		
26	f_1^-	f_2^-	f_3^-	0	1	2	3					
$*29$	f_1^-	0	0	f_4^+	1							4
$*30$	0	f_2^+	0	f_4^+						2		4
31	f_1^+	f_2^+	0	f_4^+					1	2		4
32	f_1^-	f_2^+	0	f_4^+	1					2		4
35	f_1^-	f_2^-	0	f_4^+	1	2						4
$*36$	0	0	f_3^+	f_4^+							3	4
37	f_1^+	0	f_3^+	f_4^+					1		3	4
38	f_1^-	0	f_3^+	f_4^+	1						3	4
39	0	f_2^+	f_3^+	f_4^+						2	3	4
40	f_1^+	f_2^+	f_3^+	f_4^+					1	2	3	4
41	f_1^-	f_2^+	f_3^+	f_4^+	1					2	3	4
42	0	f_2^-	f_3^+	f_4^+		2					3	4
43	f_1^+	f_2^-	f_3^+	f_4^+		2			1		3	4
44	f_1^-	f_2^-	f_3^+	f_4^+	1	2					3	4
47	f_1^-	0	f_3^-	f_4^+	1		3					4
48	0	f_2^+	f_3^-	f_4^+			3			2		4
49	f_1^+	f_2^+	f_3^-	f_4^+			3		1	2		4
50	f_1^-	f_2^+	f_3^-	f_4^+	1		3			2		4
53	f_1^-	f_2^-	f_3^-	f_4^+	1	2	3					4
$*55$	f_1^+	0	0	f_4^-				4	1			
58	f_1^+	f_2^+	0	f_4^-				4	1	2		
$*60$	0	f_2^-	0	f_4^-		2		4				
61	f_1^+	f_2^-	0	f_4^-		2		4	1			
62	f_1^-	f_2^-	0	f_4^-	1	2		4				
64	f_1^+	0	f_3^+	f_4^-				4	1		3	
67	f_1^+	f_2^+	f_3^+	f_4^-				4	1	2	3	
69	0	f_2^-	f_3^+	f_4^-		2		4			3	
70	f_1^+	f_2^-	f_3^+	f_4^-		2		4	1		3	
71	f_1^-	f_2^-	f_3^+	f_4^-	1	2		4			3	
$*72$	0	0	f_3^-	f_4^-			3	4				
73	f_1^+	0	f_3^-	f_4^-			3	4	1			
74	f_1^-	0	f_3^-	f_4^-	1		3	4				
75	0	f_2^+	f_3^-	f_4^-			3	4		2		
76	f_1^+	f_2^+	f_3^-	f_4^-			3	4	1	2		
77	f_1^-	f_2^+	f_3^-	f_4^-	1		3	4		2		
78	0	f_2^-	f_3^-	f_4^-		2	3	4				
79	f_1^+	f_2^-	f_3^-	f_4^-		2	3	4	1			
80	f_1^-	f_2^-	f_3^-	f_4^-	1	2	3	4				

Table A.1: Valid wrench constraints for planar robot. Valid and necessary constraints are marked by *.

i	$j = \mathcal{V}_i$	$p^T_{\mathcal{V}_i}$		
1	4	$\frac{1}{f_1^+ + f_2^+}$	0	0
2	8	$\frac{1}{f_1^- + f_2^-}$	0	0
3	10	$\frac{1}{2f_1^+ + 2f_3^+}$	$\frac{1}{2f_1^+ + 2f_3^+}$	$-\frac{1}{2d_a(f_1^+ + f_3^+)}$
4	15	$\frac{1}{2f_2^- - 2f_3^+}$	$-\frac{1}{2f_2^- - 2f_3^+}$	$\frac{1}{2d_a(f_2^- - f_3^+)}$
5	20	$\frac{1}{2f_1^- + 2f_3^-}$	$\frac{1}{2f_1^- + 2f_3^-}$	$-\frac{1}{2d_a(f_1^- + f_3^-)}$
6	21	$\frac{1}{2f_2^+ - 2f_3^-}$	$-\frac{1}{2f_2^+ - 2f_3^-}$	$\frac{1}{2d_a(f_2^+ - f_3^-)}$
7	29	$\frac{1}{2f_1^- - 2f_4^+}$	$-\frac{1}{2f_1^- - 2f_4^+}$	$-\frac{1}{2d_a(f_1^- - f_4^+)}$
8	30	$\frac{1}{2f_2^+ + 2f_4^+}$	$\frac{1}{2f_2^+ + 2f_4^+}$	$\frac{1}{2d_a(f_2^+ + f_4^+)}$
9	36	0	$\frac{1}{f_3^+ + f_4^+}$	0
10	55	$\frac{1}{2f_1^+ - 2f_4^-}$	$-\frac{1}{2f_1^+ - 2f_4^-}$	$-\frac{1}{2d_a(f_1^+ - f_4^-)}$
11	60	$\frac{1}{2f_2^- + 2f_4^-}$	$\frac{1}{2f_2^- + 2f_4^-}$	$\frac{1}{2d_a(f_2^- + f_4^-)}$
12	72	0	$\frac{1}{f_3^- + f_4^-}$	0

Table A.2: Result of wrench constraint computation for planar robot.

i	$p_{\mathcal{V}_i}^{\prime T}$		
1	$\frac{1}{\left(f_1^+ + f_2^+ + D_1^+\right)}$	0	0
2	$\frac{1}{\left(f_1^- + f_2^- - D_1^-\right)}$	0	0
3	$\frac{1}{2f_1^+ + 2f_3^+ - D_1^+ - D_2^+ + D_3^- / d_a}$	$\frac{1}{2f_1^+ + 2f_3^+ - D_1^+ - D_2^+ + D_3^- / d_a}$	$\frac{-1}{d_a(2f_1^+ + 2f_3^+ - D_1^+ - D_2^+ + D_3^- / d_a)}$
4	$\frac{1}{2f_2^- - 2f_3^+ - D_1^- + D_2^+ - D_3^- / d_a}$	$\frac{-1}{2f_2^- - 2f_3^+ - D_1^- + D_2^+ - D_3^- / d_a}$	$\frac{1}{d_a(2f_2^- - 2f_3^+ - D_1^- + D_2^+ - D_3^- / d_a)}$
5	$\frac{1}{2f_1^- + 2f_3^- - D_1^- - D_2^- + D_3^+ / d_a}$	$\frac{1}{2f_1^- + 2f_3^- - D_1^- - D_2^- + D_3^+ / d_a}$	$\frac{-1}{d_a(2f_1^- + 2f_3^- - D_1^- - D_2^- + D_3^+ / d_a)}$
6	$\frac{1}{2f_2^+ - 2f_3^- - D_1^+ + D_2^- - D_3^+ / d_a}$	$\frac{-1}{2f_2^+ - 2f_3^- - D_1^+ + D_2^- - D_3^+ / d_a}$	$\frac{1}{d_a(2f_2^+ - 2f_3^- - D_1^+ + D_2^- - D_3^+ / d_a)}$
7	$\frac{1}{2f_1^- - 2f_4^+ - D_1^- + D_2^+ + D_3^+ / d_a}$	$\frac{-1}{2f_1^- - 2f_4^+ - D_1^- + D_2^+ + D_3^+ / d_a}$	$\frac{-1}{d_a(2f_1^- - 2f_4^+ - D_1^- + D_2^+ + D_3^+ / d_a)}$
8	$\frac{1}{2f_2^+ + 2f_4^+ - D_1^+ - D_2^+ - D_3^+ / d_a}$	$\frac{1}{2f_2^+ + 2f_4^+ - D_1^+ - D_2^+ - D_3^+ / d_a}$	$\frac{1}{d_a(2f_2^+ + 2f_4^+ - D_1^+ - D_2^+ - D_3^+ / d_a)}$
9	0	$\frac{1}{\left(f_3^+ + f_4^+ - D_2^+\right)}$	0
10	$\frac{1}{2f_1^+ - 2f_4^- - D_1^+ + D_2^- + D_3^- / d_a}$	$\frac{-1}{2f_1^+ - 2f_4^- - D_1^+ + D_2^- + D_3^- / d_a}$	$\frac{-1}{d_a(2f_1^+ - 2f_4^- - D_1^+ + D_2^- + D_3^- / d_a)}$
11	$\frac{1}{2f_2^- + 2f_4^- - D_1^- - D_2^- - D_3^- / d_a}$	$\frac{1}{2f_2^- + 2f_4^- - D_1^- - D_2^- - D_3^- / d_a}$	$\frac{1}{d_a(2f_2^- + 2f_4^- - D_1^- - D_2^- - D_3^- / d_a)}$
12	0	$\frac{1}{\left(f_3^- + f_4^- - D_2^-\right)}$	0

Table A.3: Wrench constraint reduced for secondary terms.

Appendix B

Stability domain for PD controller

Assume a Lyapunov candidate function

$$\eta = \frac{1}{2} e^T K_a M^{-1} K_p e + \frac{1}{2} \dot{e}^T K_a \dot{e}, \tag{B.1}$$

with K_p and K_a positive definite (and for simplicity, diagonal) matrices, and M given by

$$M = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I_z \end{bmatrix}, \tag{B.2}$$

where m is the motor mass and I_z the rotational inertia. K_a is a diagonal matrix with diagonal elements $[1, 1, k_a]$, where k_a is used for adjusting the scaling between the rotational and translational axes. Note that this Lyapunov function is defined with respect to error coordinates, given by $e = q - q^*$, where q is the position vector of the robot and q^* is the desired position vector. The desired position may be fixed (for regulation) or may vary (for trajectory tracking), in which case we assume that $\dot{q}^*(t)$ is continuous and $\ddot{q}^*(t)$ is bounded.

The controller and system dynamics are given by

$$u_f = M\ddot{q} + w_e = -K_p e - K_d \dot{e} + M\ddot{q}^* + w_{ff}, \tag{B.3}$$

where K_d is also a diagonal positive definite matrix, w is a constant disturbance wrench, and w_{ff} is some form of integral term designed to cancel this disturbance. The controller force command is given by $u_f = [f_x \; f_y \; \tau]^T$.

Stability of this controller can be shown by taking the derivative of (B.1)

$$\dot{\eta} \;\; = \;\; \dot{e}^T K_a M^{-1} K_p e + \dot{e}^T K_a \ddot{e} \tag{B.4}$$

$$\dot{\eta} = \dot{e}^T K_a M^{-1} K_p e + \dot{e}^T K_a (M^{-1}(-K_p e - K_d \dot{e}) + w_{ff} \quad \text{(B.5)}$$

$$\dot{\eta} = -\dot{e}^T K_a M^{-1} K_d \dot{e} + \dot{e}^T K_a M^{-1}(w_{ff} - w_e). \quad \text{(B.6)}$$

The first term is negative semi-definite in the state. With the additional assumption that $\lim_{t\to\infty}(w_{ff} - w_e) = 0$, the second term will not affect the stability properties of the system[1]. Therefore, we can conclude that $\lim_{t\to\infty} \dot{e} = 0$. Finally, by recourse to LaSalle's Invariance Principle [120], $\lim_{t\to\infty} e^T K_a M^{-1} K_p e = 0$ and e converges to zero.

However, the outputs of controller (B.3) are subject to saturation constraints (these are from the individual actuator constraints mapped thru the force kinematic equation). The above stability result will not be valid if actuator saturation affects the effective controller output. However, given a particular controller, it is possible to determine a set, $\mathcal{U}$, of state-space position/velocity pairs where saturation will not occur. This set will not necessarily be *invariant*. For example, there may be system trajectories that leave the set. The above Lyapunov function (B.1) defines a family of invariant sets, parameterized by their size η. Any of these sets that are completely within $\mathcal{U}$ define a stable region in state space for the system and controller described by (B.3). The remainder of this chapter is concerned with finding the largest such invariant set.

For simplicity, rewrite (B.1) as:

$$\eta = \frac{1}{2} E^T K E \text{ where,} \quad \text{(B.7)}$$

$$E = \begin{bmatrix} e \\ \dot{e} \end{bmatrix}, \text{ and} \quad \text{(B.8)}$$

$$K = \begin{bmatrix} K_a M^{-1} K_p & 0 \\ 0 & K_a \end{bmatrix}. \quad \text{(B.9)}$$

From (A.24), the set $\mathcal{F}'$ of achievable output wrenches can then be expressed as

$$\mathcal{F}' := \{w \in I\!R^n \mid \Phi'(w) \leq 1\}, \quad \text{(B.10)}$$

where

$$\Phi'(w) = \max\left(p_1'^T w, p_2'^T w, \ldots, p_1 2'^T w\right), \quad \text{(B.11)}$$

[1]For a constant disturbance this assumption is valid; for a position- or time-varying disturbance, stability cannot be assured without a more detailed analysis than is presented here.

where the $p' \in I\!R^3$ vectors each encode a plane in $I\!R^3$, such that the condition

$$p_i'^T w \leq 1 \tag{B.12}$$

must be met for all $j \in 1...12$ for wrench w to be within the capabilities of the actuators. The state-dependent terms, $w_s := -K_p e - K_d \dot{e}$, of controller (B.3) can be used to provide a mapping from wrench space to state space. The other terms, $w_t := M\ddot{q}^* + w_{ff}$, are assumed to be bounded, and their combined bounds $D_i^- < M\ddot{q}_i^* + w_{ff} < D_i^+$ must be used in the computation of the p_i' vectors, as described in Appendix A. To compute the mapping, first note that

$$w_s = KE, \tag{B.13}$$

so that from (B.12) the relation

$$p_i' KE \leq 1 \quad \forall i \in \{1 \ldots 12\} \tag{B.14}$$

must be satisfied if KE is to not saturate the actuators for all bounded values of $w_t \in \mathcal{D}$.

Relation (B.14) defines a set, $\mathcal{S}'$, in the state-space of the robot that must completely contain the invariant controller domain. The largest such domain will be the largest hyperellipsoid defined by (B.7) that lies within $\mathcal{S}'$.

First, define $L_i := p'^T K_i$ and B_i such that the actuator constraints can be expressed as

$$B_i(E) := L_i E \leq 1. \tag{B.15}$$

If the value of η is minimized over the B_i, the minimum value will give the size of the hyperellipsoid when it just touches the hyperplane. So, find the minimum value, η_i for

$$A(E) = \frac{1}{2} E^T K E, \tag{B.16}$$

subject to each of the constraints B_i. A LaGrange multiplier approach is used, giving the simultaneous equations

$$\frac{\partial A(E)}{\partial E} + \lambda_i \frac{\partial B_i(E)}{\partial E} = 0 \tag{B.17}$$

$$B_i(E) = 1. \tag{B.18}$$

Substituting and taking the partial derivatives,

$$0 = KE + \lambda_i L_i^T \tag{B.19}$$

$$E = -\lambda_i K^{-1} L_i^T \tag{B.20}$$

Using this expression for E in the constraint equation and solving for λ_i yields

$$-L_i(\lambda_i K^{-1} L_i^T) = 1 \tag{B.21}$$

$$\frac{-1}{L_i K^{-1} L_i^T} = \lambda_i. \tag{B.22}$$

Next, use this λ_i to find E

$$0 = KE + \left(\frac{-1}{L_i K^{-1} L_i^T}\right) L_i^T \tag{B.23}$$

$$E = K^{-1}\left(\frac{1}{L_i K^{-1} L_i^T}\right) L_i^T. \tag{B.24}$$

Finally, η_i can be computed as

$$\eta_i = \frac{1}{2} E^T K E \tag{B.25}$$

$$= \frac{1}{2} L_i \left(\frac{1}{L_i K^{-1} L_i^T}\right) K^{-1} K K^{-1} \frac{1}{2} \left(\frac{1}{L_i K^{-1} L_i^T}\right) L_i^T \tag{B.26}$$

$$= \frac{1}{2} \frac{(L_i K^{-1} L_i^T)}{(L_i K^{-1} L_i^T)^2} \tag{B.27}$$

$$= \frac{1}{2} \frac{1}{L_i K^{-1} L_i^T} \tag{B.28}$$

One design question remaining is the choice of the k_a scaling factor between the translation and rotation variables. For the planar motor application, k_a can be chosen to force the maximum angle to be within the small angle range of the device. Note that the η_is depend on k_a (the K matrix includes k_a), so that a different k_a must be computed for each case. The one associated with the smallest η_i will be used for the final controller design. Assuming that the desired angle is fixed at zero, the maximum angle constraint is used by substituting $E = [0\ 0\ q_{\theta,max}\ 0\ 0\ 0]$ into (B.7)

$$\begin{aligned} \eta &= \tfrac{1}{2} q_{\theta,max} K_{3,3} \theta_{max} \\ &= \tfrac{1}{2} q_{\theta,max}^2 k_a k_{p3}/m. \end{aligned} \tag{B.29}$$

This equation can then be substituted into (B.28) to give a scalar relation that can be solved for k_a (using a symbolic math program) to give

$$k_{ai} = \frac{-k_{p1}k_{p2}\left(q_{\theta,max}^2 L_{i,3}^2 I_z + q_{\theta,max}^2 k_{p3} L_{i,6}^2 - I_z\right)}{q_{\theta,max}^2 k_{p3}\left(L_{i,1}^2 m k_{p2} + L_{i,2}^2 m k_{p1} + k_{p1}k_{p2}L_{i,4}^2 + k_{p1}k_{p1}L_{i,5}^2\right)} \tag{B.30}$$

Note that the denominator is always positive, while the term in parenthesis in the numerator may be positive or negative, making the computed value of k_a negative or

positive, respectively. A negative k_a value indicates an invalid design (*i.e.* the chosen controller gains are too high or $q_{\theta,max}$ is too big for the given physical parameters).

Appendix C

Hybrid controller derivations

Proposition 3 *The time derivative of* $\eta_\omega = \frac{1}{2}\omega^T\omega$ *(6.17), with* $\omega = (I - \frac{\hat{v}\hat{v}^T}{\hat{v}^T\hat{v}})\dot{q}$, *under the control law (6.11) can be expressed as* $\dot{\eta_\omega} = -k\sigma\omega^T\omega$.

Begin by computing $\dot{\omega}$:

$$\begin{aligned}\dot{\omega} &= \tfrac{d}{dt}\left(\left(I - \tfrac{\hat{v}\hat{v}^T}{\hat{v}^T\hat{v}}\right)\dot{q}\right) \\ &= \left(I - \tfrac{\hat{v}\hat{v}^T}{\hat{v}^T\hat{v}}\right)\ddot{q} - \tfrac{d}{dt}\left(\tfrac{\hat{v}\hat{v}^T}{\hat{v}^T\hat{v}}\right)\dot{q} \\ &= \left(I - \tfrac{\hat{v}\hat{v}^T}{\hat{v}^T\hat{v}}\right)(-k\sigma(\dot{q} - \hat{v}) + \zeta D\hat{v}\dot{q}) - \tfrac{d}{dt}\left(\tfrac{\hat{v}\hat{v}^T}{\hat{v}^T\hat{v}}\right)\dot{q}\end{aligned} \tag{C.1}$$

then by applying Proposition 4 (below),

$$\begin{aligned}&= -k\sigma\left(I - \tfrac{\hat{v}\hat{v}^T}{\hat{v}^T\hat{v}}\right)(\dot{q} - \hat{v}) + \zeta D\hat{v}\dot{q} - \zeta\tfrac{\hat{v}\hat{v}^T}{\hat{v}^T\hat{v}}D\hat{v}\dot{q} - \left(\tfrac{D\hat{v}\dot{q}\hat{v}^T}{\hat{v}^T\hat{v}} + \tfrac{\hat{v}\dot{q}^T D\hat{v}}{\hat{v}^T\hat{v}} - \tfrac{2(\hat{v}^T D\hat{v}\dot{q})(\hat{v}\hat{v}^T)}{(\hat{v}^T\hat{v})^2}\right)\dot{q} \\ &= -k\sigma\left(I - \tfrac{\hat{v}\hat{v}^T}{\hat{v}^T\hat{v}}\right)(\dot{q} - \hat{v}) + \zeta D\hat{v}\dot{q} - \zeta\tfrac{\hat{v}\hat{v}^T D\hat{v}\dot{q}}{\hat{v}^T\hat{v}} - \tfrac{D\hat{v}\dot{q}\hat{v}^T\dot{q}}{\hat{v}^T\hat{v}} - \tfrac{\hat{v}\dot{q}^T D\hat{v}\dot{q}}{\hat{v}^T\hat{v}} + \tfrac{2(\hat{v}^T D\hat{v}\dot{q})(\hat{v}\hat{v}^T)\dot{q}}{(\hat{v}^T\hat{v})^2},\end{aligned} \tag{C.2}$$

then apply the definition $\zeta := \frac{\hat{v}^T\dot{q}}{\hat{v}^T\hat{v}}$ and continue to simplify

$$\begin{aligned}\dot{\omega} &= -k\sigma\left(I - \tfrac{\hat{v}\hat{v}^T}{\hat{v}^T\hat{v}}\right)(\dot{q} - \hat{v}) + \zeta D\hat{v}\dot{q} - \zeta\tfrac{\hat{v}(\hat{v}^T D\hat{v}\dot{q})}{\hat{v}^T\hat{v}} - \zeta D\hat{v}\dot{q} - \tfrac{\hat{v}\dot{q}^T D\hat{v}\dot{q}}{\hat{v}^T\hat{v}} + \tfrac{2(\hat{v}^T D\hat{v}\dot{q})\hat{v}}{\hat{v}^T\hat{v}} \\ &= -k\sigma\left(I - \tfrac{\hat{v}\hat{v}^T}{\hat{v}^T\hat{v}}\right)(\dot{q} - \hat{v}) - \zeta\tfrac{(\hat{v}\dot{q}^T D\hat{v})\hat{v}}{\hat{v}^T\hat{v}} - \tfrac{(\hat{v}\dot{q}^T D\hat{v})\dot{q}}{\hat{v}^T\hat{v}} + \tfrac{2(\hat{v}\hat{v}^T D\hat{v})\dot{q}}{\hat{v}^T\hat{v}} \\ &= -k\sigma\left(I - \tfrac{\hat{v}\hat{v}^T}{\hat{v}^T\hat{v}}\right)(\dot{q} - \hat{v}) + \tfrac{(\hat{v}\dot{q}^T D\hat{v})}{\hat{v}^T\hat{v}}(\dot{q} - \zeta\hat{v}) \\ &= -k\sigma\left(I - \tfrac{\hat{v}\hat{v}^T}{\hat{v}^T\hat{v}}\right)(\dot{q} - \hat{v}) + \tfrac{\hat{v}(\dot{q}^T D\hat{v}\omega)}{\hat{v}^T\hat{v}}.\end{aligned} \tag{C.3}$$

Next, use this result to compute

$$\begin{aligned}\tfrac{d}{dt}\tfrac{1}{2}\omega^T\omega &= \omega^T\dot{\omega} \\ &= \left(\left(I - \tfrac{\hat{v}\hat{v}^T}{\hat{v}^T\hat{v}}\right)\dot{q}\right)^T\left(-k\sigma\left(I - \tfrac{\hat{v}\hat{v}^T}{\hat{v}^T\hat{v}}\right)(\dot{q} - \hat{v}) + \tfrac{\hat{v}(\dot{q}^T D\hat{v}\omega)}{\hat{v}^T\hat{v}}\right).\end{aligned} \tag{C.4}$$

Then, noting that $\left(I - \frac{\hat{v}\hat{v}^T}{\hat{v}^T\hat{v}}\right)\hat{v} = 0$, it can be shown that

$$\begin{aligned}\tfrac{d}{dt}\tfrac{1}{2}\omega^T\omega &= -k\sigma\left(\left(I - \tfrac{\hat{v}\hat{v}^T}{\hat{v}^T\hat{v}}\right)\dot{q}\right)^T\left(I - \tfrac{\hat{v}\hat{v}^T}{\hat{v}^T\hat{v}}\right)\dot{q} \\ &= \omega^T\omega.\end{aligned} \tag{C.5}$$

Proposition 4 *The time derivative of* $\frac{d}{dt}\frac{\hat{v}\hat{v}^T}{\hat{v}^T\hat{v}}$ *is* $\frac{D\hat{v}\dot{x}\hat{v}^T}{\hat{v}^T\hat{v}} + \frac{\hat{v}\dot{x}^T D\hat{v}}{\hat{v}^T\hat{v}} - \frac{2(\hat{v}^T D\hat{v}\dot{x})(\hat{v}\hat{v}^T)}{(\hat{v}^T\hat{v})^2}$.

Begin by using the quotient rule, then simplify

$$\begin{aligned}
\tfrac{d}{dt}\tfrac{\hat{v}\hat{v}^T}{\hat{v}^T\hat{v}} &= \tfrac{\frac{d}{dt}(\hat{v}\hat{v}^T)(\hat{v}^T\hat{v}) - \frac{d}{dt}(\hat{v}^T\hat{v})(\hat{v}\hat{v}^T)}{(\hat{v}^T\hat{v})^2} \\
&= \tfrac{(D\hat{v}\dot{q}\hat{v}^T + \hat{v}(D\hat{v}\dot{q})^T)(\hat{v}^T\hat{v}) - ((D\hat{v}\dot{q})^T\hat{v} + \hat{v}^T D\hat{v}\dot{q})(\hat{v}\hat{v}^T)}{(\hat{v}^T\hat{v})^2} \\
&= \tfrac{(D\hat{v}\dot{q}\hat{v}^T + \hat{v}\dot{q}^T D\hat{v})(\hat{v}^T\hat{v}) - (2\hat{v}^T D\hat{v}\dot{q})(\hat{v}\hat{v}^T)}{(\hat{v}^T\hat{v})^2} \\
&= \tfrac{D\hat{v}\dot{q}\hat{v}^T}{\hat{v}^T\hat{v}} + \tfrac{\hat{v}\dot{q}^T D\hat{v}}{\hat{v}^T\hat{v}} - \tfrac{2(\hat{v}^T D\hat{v}\dot{q})(\hat{v}\hat{v}^T)}{(\hat{v}^T\hat{v})^2}.
\end{aligned} \tag{C.6}$$