

COMPLIANT BEHAVIORS FOR SUPERVISED AND AUTONOMOUS ROBOTIC OPERATIONS

BY

JOSEPH D. CRESSMAN

Submitted in Partial Fulfillment of the Requirements for the Degree of
Master of Science

Electrical, Computer and Systems Engineering Department

CASE WESTERN RESERVE UNIVERSITY

May, 2023

COMPLIANT BEHAVIORS FOR SUPERVISED AND AUTONOMOUS ROBOTIC OPERATIONS

Case Western Reserve University
Case School of Graduate Studies

We hereby approve the thesis¹ of

Joseph D. Cressman

for the degree of

Master of Science

Wyatt Newman

Committee Chair, Advisor
Electrical, Computer and Systems Engineering

January 17, 2023

M. Cenk Çavuşoğlu

Committee Member
Electrical, Computer and Systems Engineering

January 17, 2023

Zonghe Chua

Committee Member
Electrical, Computer and Systems Engineering

January 17, 2023

¹We certify that written approval has been obtained for any proprietary material contained therein.

Contents

1	Introduction	2
2	Methods	6
2.1	Overview	6
2.1.1	Admittance control and Virtual Attractor	6
2.2	Behaviors	9
2.2.1	Pose Target Wrench Limiting	10
2.2.2	Reset Wrench Equilibrium	10
2.2.3	Dither	11
3	Tasks	12
3.1	Berthing	13
3.2	Plug insertion	18
3.3	Opening and closing a latched door	21
3.4	Wide, shallow lid installation	22
3.5	Socket wrench insertion	25
4	Capture	28
4.1	Simulation	30
4.1.1	Fixed Vehicle	30
4.1.2	Free vehicle	37
4.1.3	Deliberate distance overestimation	39
4.1.4	Teleporting Attractor	40
4.1.5	Positive Feedback Attractor	42
4.1.6	Fixed Displacement Attractor	44
4.2	Hardware Experiments	45
4.2.1	Wheelie Satellite	45

4.3	Air bearings	48
4.3.1	Construction	48
4.3.2	Tracking	50
4.3.3	Experiments	53
5	Discussion and Future Work	56

List of Figures

1	The layers of the control architecture.	7
2	A 2D schematic of the virtual attractor.	7
3	A system diagram of the admittance controller	9
4	An overhead photo of the experimental setup for berthing.	14
5	A step-by-step illustration of the berthing procedure.	16
6	A flowchart of the berthing state machine.	17
7	A plot of the attractor position and orientation during three automated berthing operations.	18
8	A plot of the measured forces and torque during three automated berthing trials.	19
9	The steps involved in plug insertion	20
10	A screenshot taken during the plug insertion process.	20
11	Attractor placement for door opening	22
12	A screenshot taken during the door manipulation	23
13	Exaggerated geometry of the saucepan and lid.	24
14	The steps for saucepan state lid placement.	25
15	The initial contact forces during socket shank insertion	26
16	An illustration of the steps involved in shank insertion.	26
17	Hybrid capture with velocity control until contact and compliance thereafter. Yellow is the satellite position, and blue is the robot end- effector.	29
18	A schematic diagram of the modified accommodation controller. . . .	30
19	A schematic representation of the physical model used for initial sim- ulation.	31
20	Simulated 1-D capture under strict trajectory control.	35
21	Flowchart of the simulation loop.	36

22	Simulated 1-D capture with a bumpless transition into compliance. . .	36
23	The simulated system with a free-floating robot base.	37
24	Simulated 1-D capture with a bumpless transition into compliance but with a free robot base	39
25	Simulated 1-D capture with a bumpless transition into compliance but with a free robot base, where the distance to the robot is deliberately underestimated.	41
26	Simulated 1-D capture with a bumpless transition into compliance but with a free robot base, where the attractor was teleported forward upon contact.	42
27	Simulated 1-D capture with a bumpless transition into compliance but with a free robot base. The attractor motion is equal to the robot motion plus a constant displacement.	43
28	Simulated 1-D capture with a bumpless transition into compliance but with a free robot base. The attractor position is equal to the robot motion plus a constant displacement and the attractor (feedforward) velocity was taken from the planned trajectory.	44
29	The top left quadrant is a view from the robot's onboard POV camera, while the bottom left quadrant is a view from an external camera. The top right shows a plot of the robot's position as a function of time, where the cyan curve is the relevant axis. The lower plot shows the measured forces as a function of time (The magenta curve is the relevant axis).	46

30	The top left quadrant is a view from the robot's onboard POV camera, while the bottom left quadrant is a view from an external camera. The top right shows a plot of the robot's position as a function of time, where the cyan curve is the relevant axis. The lower plot shows the measured forces as a function of time (The magenta curve is the relevant axis).	47
31	A cross sectional diagram of the air bearing.	49
32	A flowchart representation of the EKF	51
33	A screenshot of the console during capture testing.	54

Compliant Behaviors for Supervised and Autonomous Robotic Operations

Abstract

by

JOSEPH D. CRESSMAN

The demands of traditional industrial robotics differ significantly from those of remote, supervised operations in hostile environments, such as operations in space. While industry requires robots that can perform repetitive tasks with precision and speed, the space environment needs robots to cope with uncertainties, dynamics, and communication delays or interruptions, similar to human astronauts. These demands make a well-suited application for compliant robotics and behavior-based programming. Pose Target Wrench Limiting (PTWL) is a compliant behavior paradigm developed specifically to meet these demands. PTWL controls a robot by moving a virtual attractor to a target pose. The attractor applies virtual forces, based on stiffness and damping presets, to an underlying admittance controller. Guided by virtual forces, the robot will follow the attractor until safety conditions are violated or success criteria are met. PTWL was tested on a variety of quasi-static and dynamic tasks that may be useful for future space operations. Results demonstrate that PTWL is a powerful tool. It makes teleoperation easy and safe for a wide range of quasi-static tasks. It also facilitates the creation of semi-autonomous state machines that can reliably complete complex tasks with minimal human intervention.

1 Introduction

Space operations have long been a driver of robotic technology. Maintenance tasks often require complex manipulation and real-time adaptability at which humans excel. Unfortunately, these tasks also expose astronauts to dangerous conditions and waste their valuable time. Ideally, robots should perform dangerous tasks under the supervision of ground based operators. In practice, transmission latency and interruptions make this difficult. A robust and adaptable control scheme is necessary to overcome these challenges.

In terrestrial applications, safety, efficiency and competence can often be synergistically achieved by pairing a human operator with a robot. Competence can generally be increased by closely coupling the remote task to the human operator's actions and senses. For example, in First Person View (FPV) drone racing, an operator wears goggles that stream live video from the drone directly to the eyes. This helps to immerse the operator; making them feel like they are actually piloting from a cockpit on the drone. Combined with the analog joysticks on the remote controller, FPV allows the operator to react quickly to obstacles and to naturally adjust to variable wind conditions ([1]).

Unfortunately, for many situations immersive telepresence is impractical or even counterproductive. If latency between performing an action and observing the result is on the order of 1/10th of a second, the operator may experience nausea [2]. Significant time delays also have the potential to cause instability [3]. When an operator performs an action and does not immediately see results, they may exaggerate their input to compensate for perceived unresponsiveness. Experienced operators can usually learn to adapt to substantial delays, but this comes at a cost of speed and efficiency. With large delays, operators are forced to break up a task into a set of small motions separated by pauses. Task completion time increases linearly with latency

[4], and high latency makes it difficult or impossible to carry out more complicated manipulation tasks. For these reasons, unmanned robotic satellite servicing has not yet been accomplished, despite decades of research in the area [5, 6].

To address these problems, [7] introduced supervisory control. Instead of relying on the reactions and motor skills of a human operator for real-time control, the remote robot operates with relative autonomy at the lowest level and responds to higher-level directives from the human “supervisor”. For robustness, the remote computer should be able to deal safely with any contingency that may arise without human intervention. This may mean that the remote computer only handles relatively short movements. For efficiency and speed, however, the human operator should interfere as infrequently as possible. For a given task and latency, a relative sweet spot exists that balances the level of human control with autonomy. With a more robust and independent autonomous subsystem, less human intervention is required and the task can be completed more quickly.

Even in terrestrial applications with little to no transmission latency, supervisory control can be useful. Immersive telepresence systems are expensive and complex, and supervisory control greatly facilitates manipulation tasks that would otherwise require complicated and precise commands. For example, researchers in the 2015 DARPA robotics challenge found supervisory control to be essential for manipulation tasks. For known manipulation tasks, such as turning a valve ([8]) and cutting through a wall ([9]), it is more reliable and faster for an operator to initiate predefined skills rather than direct fine motor movements.

In order to achieve robustness, the robot should be able to respond to its circumstances in real time. One of the most important events is contact. If the robot makes contact with something in its environment and fails to adjust its motion in response, it can very easily damage itself or its environment. One of the simplest methods by which a robot can respond to contact is the “guarded move.” Under a guarded move,

a robot carries out a motion until contact forces exceed some threshold, at which point motion is terminated. Many industrial robots have this kind of functionality built into their controllers at a very low level, and will shut down if actuator currents get too high. While the guarded move is very safe, on its own it does very little to continue the task after contact.

A more effective approach is to dynamically adjust the motion in response to forces. This is superficially similar to the way humans use tactility to accomplish manipulation tasks ([10], [11], [12]). When robots do this, it generally falls under the umbrella of compliant motion control. Admittance control is a form of compliant motion control in which the robot is driven to behave with a defined mechanical admittance (the reciprocal of mechanical impedance). Admittance control in the current context will be implemented as twist commands (a 6-DOF end-effector velocity) in response to sensed wrenches (6-DOF force/torque measurements at the end effector). The system considered has a restriction that underlying servo controls are frozen, and the only option for achieving compliant behavior is to command joint velocities (derived from desired end-effector twist) as some function of sensed wrenches. Admittance of the robot is thus a response of velocity (twist) due to imposition of a wrench, consistent with the definition of a mechanical admittance.

Ideally, an admittance controller should maximize the motion response to an applied force while also maintaining stability. This is especially useful for human-robot interaction because it allows the robot to be easily guided by applied forces [13]. The controller can be designed to passively interact with the environment, which guarantees stability [14]. This is useful for space applications [15], since instability could result in irrecoverable damage to equipment in orbit. Combining compliant motion control with more discrete behaviors such as guarded moves can be more effective than either method on its own.

Use of compliant-motion control and soft attractors offers opportunity for safe and

gentle interactions. Given such an underlying system, the next layer of abstraction to consider is, how should the compliance parameters (stiffnesses and dampings) and soft attractor trajectories be generated to perform useful tasks? To help address this challenge, three parameterized “behaviors” are presented, and these behaviors are shown to be effective in performing a wide variety of interaction tasks. These behaviors can be invoked incrementally under supervisory control. Additionally, they constitute effective building blocks for constructing state machines that achieve higher levels of autonomy.

Behaviors have been employed for assembly tasks in the past to great effect. Behaviors built on top of compliant motion controllers have been especially successful ([16]). In this research, a variety of compliant behaviors were distilled into a much smaller yet still powerful set of behaviors.

2 Methods

2.1 Overview

The control architecture for our robot is comprised of several layers of increasing abstraction. At the lowest layer, the robot joint velocities are controlled by a high-speed servo loop. The joint velocities are specified by an admittance controller at the next layer. The admittance controller receives force input from a sensor in series with the robot’s tool flange and virtual forces computed from the virtual attractor. The pose of this virtual attractor is controlled by various behavior primitives that move the attractor in a manner defined by a handful of parameters. Once a force threshold is crossed or a desired pose is achieved, the behavior terminates and returns an exit condition. At the highest layer, a human operator or an automated state machine selects behavior primitives and appropriate parameters. In the case of a state machine, the subsequent state depends on the exit condition returned by the behavior. Behavior primitives are executed sequentially until the task is completed. This architecture is summarized in Fig. 1.

2.1.1 Admittance control and Virtual Attractor

The goal of the admittance controller is to “admit” environmental forces via its virtual dynamics. In our implementation, this is achieved through the use of a “Virtual Attractor.” The attractor consists of a desired pose together with a matrix of virtual stiffness. The attractor is attached to the robot via “virtual springs” at the “port of interaction,” which is most conveniently the frame of the force-torque sensor, as illustrated in Fig. 2. Together, these are used to generate virtual forces that compete with the forces measured by the force-torque sensor.

While there are many admittance-control implementation variations, the method

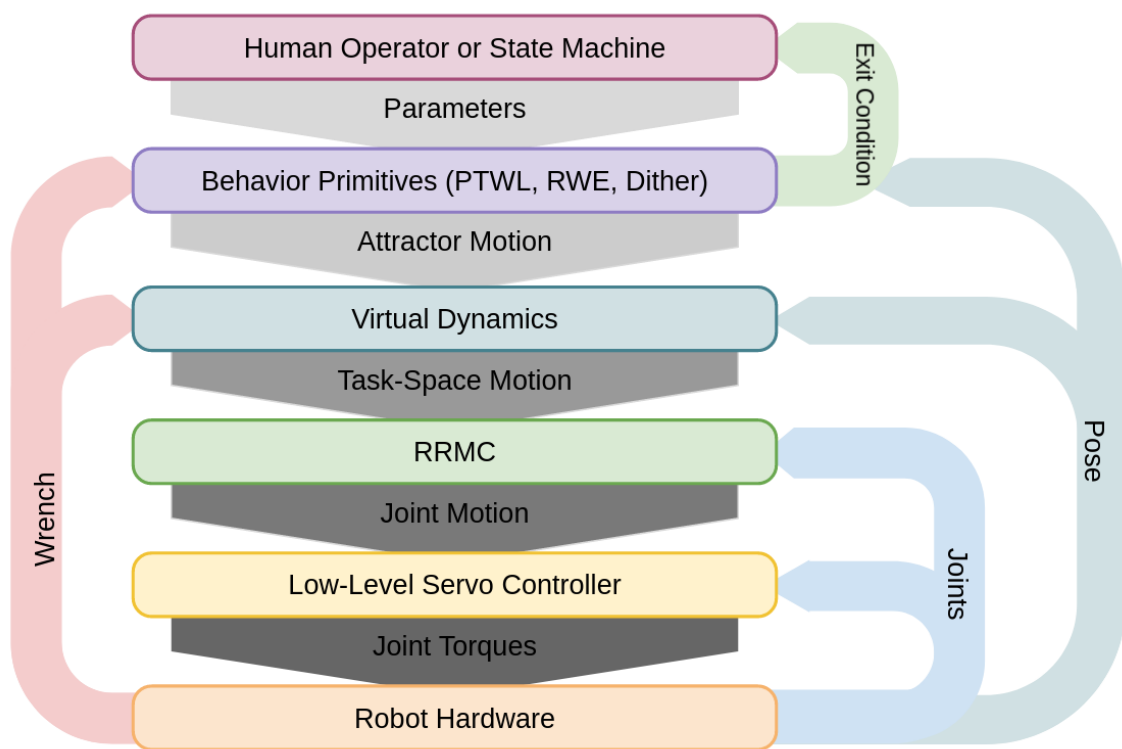


Figure 1: The layers of the control architecture.

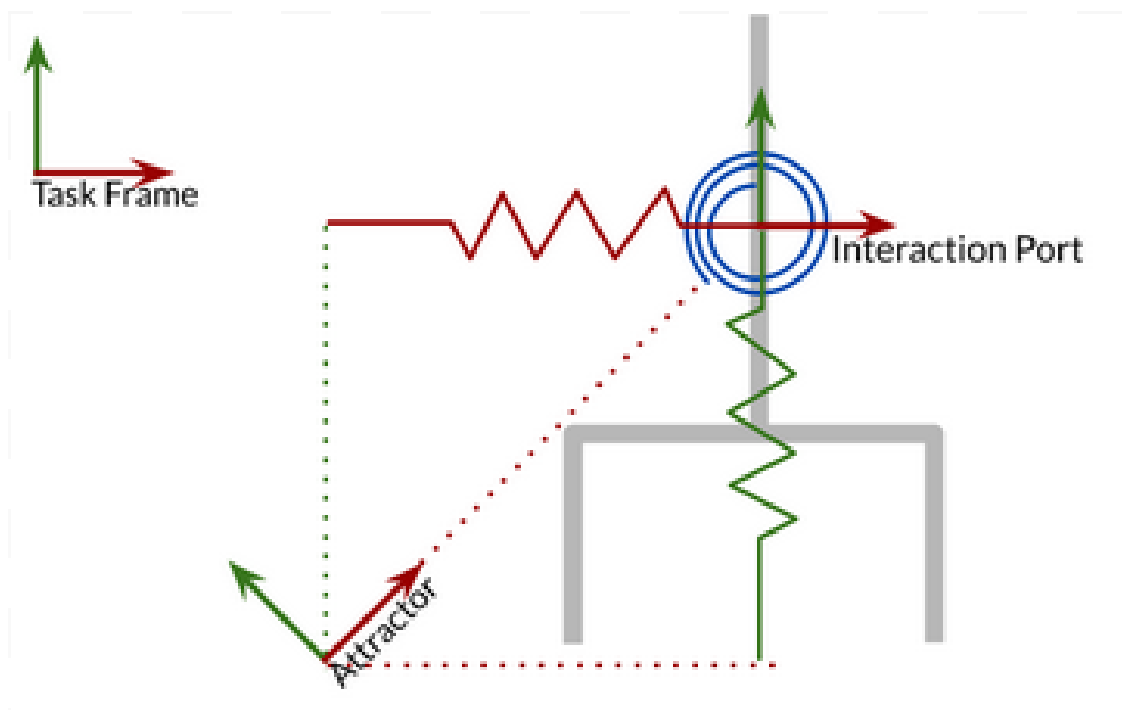


Figure 2: A 2D schematic of the virtual attractor.

used for the experiments reported below as follows. It is heavily related to the controller described by [17] and passivity analysis described by [18].

$$\mathbf{0} = \mathbf{w}_{net} = \mathbf{w}_{ft} + \tilde{\mathbf{w}}_{ft,K} + \tilde{\mathbf{w}}_{ft,B} \quad (1)$$

$$\tilde{\mathbf{w}}_{ft,B} = -\mathbf{B}_{des}\mathbf{t}_{ft} \quad (2)$$

$$\mathbf{0} = \mathbf{w}_{ft} + \tilde{\mathbf{w}}_{ft,K} - \mathbf{B}_{des}\mathbf{t}_{ft} \quad (3)$$

$$\mathbf{B}_{des}\mathbf{t}_{ft} = \mathbf{w}_{ft} + \tilde{\mathbf{w}}_{ft,K} \quad (4)$$

$$\mathbf{t}_{ft,des} = \mathbf{B}_{des}^{-1}[\mathbf{w}_{ft} + \tilde{\mathbf{w}}_{ft,K}] \quad (5)$$

In the above, $\mathbf{w}_{ft}$ is the force/torque wrench exerted by the environment on the interaction port of the robot. This is assumed to be identical to the force/torque that is measured at this point. The terms $\tilde{\mathbf{w}}_{ft,K}$ and $\tilde{\mathbf{w}}_{ft,B}$ are virtual wrenches, computed based on the assigned (desired) virtual stiffness and virtual damping at the interaction port. The virtual wrench due to the virtual spring scales with the stretch between the virtual attractor pose (defined per trajectories generated for specific tasks) and the actual pose of the interaction port (computed in real time via forward kinematics). The virtual wrench due to damping scales with the twist (6-DoF velocity) of the interaction port, $\mathbf{t}_{ft}$, and this quantity is also computed in real time based on the measured joint velocities and the robot's Jacobian. The compliance controller (as implemented) attempts to achieve the twist vector $\mathbf{t}_{ft,des}$ that is consistent with the defined physical and virtual interaction wrenches. This is accomplished by continuously recomputing $\mathbf{t}_{ft,des}$ and using a Resolved Rate Motion Controller (which uses the Jacobian inverse) to convert this task-space velocity into joint motions that are sent to the robots internal servo. This subsystem is illustrated schematically in Fig. 3.

This control scheme is useful when the robot is moving unobstructed through space, and when it is performing tasks. In free-space, the robot will just follow the

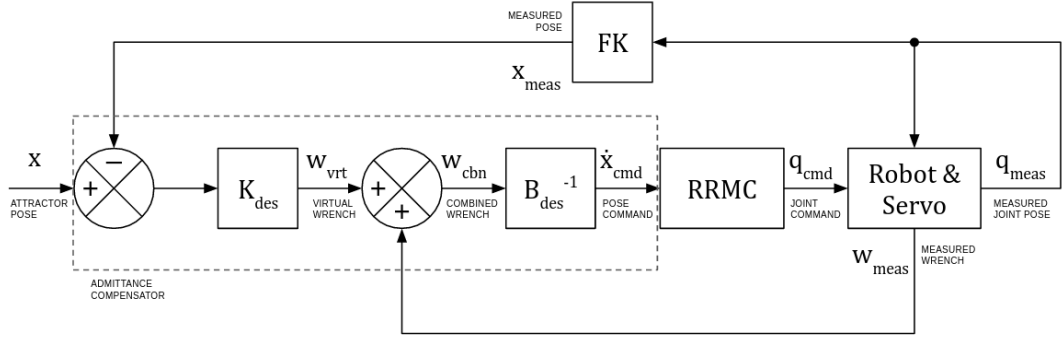


Figure 3: A system diagram of the admittance controller

virtual attractor as quickly as $\mathbf{B}_{des}$ allows. This eliminates the need for careful second-order trajectory planning, as the damping will ensure that the robot never moves too quickly. In contact, the robot will still try to follow the position commands, but it will never allow the contact forces to get too high. When necessary, higher forces can be easily achieved by simply moving the attractor farther in the desired direction.

2.2 Behaviors

Regardless of the specific compliant-motion controllers implementation, the intent is to provide a foundation for performing interactive tasks in which contact wrenches must be exerted, regulated and interpreted. To exploit an underlying compliant-motion controller, a subsequent layer of abstraction may be defined, described here as compliant “behaviors.”

Behaviors are actions that a robot can perform without operator intervention. Simple behaviors can be made reliable and robust, and these behaviors can be invoked directly, under supervisory control, or chained together in state machines to create higher levels of autonomy.

Three specific behaviors were found to be remarkably capable and flexible. It will be shown that these three behaviors can be exploited to perform a wide variety of

manipulation tasks while exhibiting remarkable robustness. These three behaviors are optimized and generalized versions of those described by [19], and are capable of performing the same functionality with fewer steps. This simplifies both teleoperation and state-machine design.

2.2.1 Pose Target Wrench Limiting

PTWL is a simple scheme for generating attractor motion. PTWL is defined by a handful of parameters: 1) a desired displacement of the attractor, 2) the duration of the attractor move, 3) a force/torque threshold, 4) a pose tolerance, 5) the stiffness matrix, 6) the damping matrix, and 7) a watchdog timer. Once PTWL begins, the attractor is moved at a constant velocity toward the desired pose in the specified amount of time. This will continue until the desired pose is reached within tolerance, or the force/torque threshold is exceeded, or the watchdog timer runs out. Once any of the exit conditions are met, the attractor is immediately frozen and remains stationary until another command is given.

In general, all of these parameters are held constant for a given task, except for the desired displacement. This allows the operator to focus on plotting desired motions of the manipulator to complete the task. PTWL offers intuitive control over both position and force. This allows it to be applied to a wide range of manipulation tasks that involve fine control over forces and motions.

2.2.2 Reset Wrench Equilibrium

The behavior “RWE”, for “Reset Wrench Equilibrium”, moves the attractor to be coincident with the port of interaction. As a result, the virtual wrenches (both stiffness and damping) are definitionally set to zero, and only the sensed (physical) wrench contributes to the desired twist. The desired twist, via the compliance controller, converges to zero as the contact wrench is driven to zero. This relieves contact forces

without substantially changing the robot pose. RWE is useful between PTWL moves, especially if the move terminated on a force/torque threshold. If PTWL were used by an operator or state machine without RWE, it could get stuck every time a force threshold were crossed. RWE allows the operator to easily relieve contact efforts without substantially moving the robot end-effector.

2.2.3 Dither

Over the course of several experiments, some sequences of attractor movements tend to appear multiple times in various contexts. These sequences offer enough utility to warrant their own separate behavior. One example is “dithering”, in which the attractor oscillates while simultaneously moving in a specified direction, usually in order to free itself from a stuck position. Dithering was found to be especially useful in the tight-tolerance sleeve-on-peg task, described below, where any misalignment could cause the sleeve to become lodged in place. In order to dislodge, the sleeve needs to be moved in a specific direction that depends on the exact way in which it is lodged. RWE is often insufficient for this because it will only move the attractor until forces are reduced, which does not necessarily solve the problem. Dithering, however, wiggles the attractor in all directions until the correct one is found by trial and error.

Several dither modes were created to deal with various circumstances in which they were found useful. These are divided into torsional and translational dither modes. In torsional mode, the attractor rotates about a specified axis while keeping its origin on the axis. In translational mode, the attractor orientation is fixed while its origin is dragged in circles about the specified axis. These two modes are divided further into push and pull modes, where the attractor is either moved forward or backward along the specified axis.

3 Tasks

One potential application of space robotics is to extend the life of existing satellites by capturing and refueling them. This operation requires several distinct phases. First, during the capture phase the robot must grapple the satellite. Grappling is dominated by dynamic interactions, and also requires machine vision or some other form of tracking [20]. Admittance control has been proposed and tested as an appropriate control scheme for robotic satellite capture [21]. Capture with our admittance controller is explored in its own section .

After the satellite is captured, it must be berthed. This involves manipulating the satellite into position such that it can be held in place by fixed posts on the servicing vehicle. Once successfully berthed, the servicing phase can begin. This may involve several servicing tasks, such as cutting, refueling, and tool exchange.

For each of these tasks, a terrestrial analog was created in the lab environment to roughly simulate the geometry and forces involved. This is an effective testing method, despite the inherent differences between the laboratory and space environments and is in many ways superior to purely virtual simulation ([22]). The tasks were carried out with the terrestrial analog to determine appropriate parameters and control strategies.

Other space robotics applications involve a variety of as-of-yet unforeseen operations. It is anticipated that PTWL and the compliant behaviors developed here are sufficiently versatile to address almost any quasi-static manipulation task. To demonstrate this, several tasks that may be useful for servicing, assembly or manufacturing were tested.

3.1 Berthing

As discussed above, berthing involves manipulating the satellite into a specific position so that it can be held in place for servicing. For terrestrial experiments, the 6-degree-of-freedom satellite is modeled using 3-DoF planar air bearings. The bearings are fixed to an elongated sled and weights are distributed such that the moment of inertia is large and the center of gravity is significantly offset from the port of interaction, as would likely be the case for a robot externally gripping a large satellite bus. In the 3-DoF emulation, the payload is to be manipulated to align with two posts. The robot grips a handle on the payload using a pneumatic three-jaw chuck, which completely constrains payload. These components are labeled in Fig. 4. Additionally, the virtual stiffnesses are set to near zero in all degrees of freedom constrained by the plane (z , ϕ_x and ϕ_y). This mitigates issues that could arise from slight misalignment of the robot base from the payload (emulated satellite) plane, such as lifting or tilting.

Experimentally, the robot was controlled under incremental supervisory commands to achieve berthing, thus demonstrating that berthing is completely achievable through the previously developed compliant behaviors, PTWL and RWE. This process is illustrated in Fig. 5 (The red-green axes represent the virtual attractor’s location). With the satellite roughly aligned with the berthing posts (1), the virtual attractor was moved toward and past the posts until the satellite contacted and pressed against both (2). Since the original alignment was imperfect, this step usually left the satellite canted, with one clamp touching one post and the other post just touching the satellite’s body (2). Looking at the camera view, one could see which post was in contact, and in which direction to move the satellite to correct the alignment. To make this correction, RWE was first exerted to eliminate contact forces. Subsequently, the attractor was displaced to induce sliding sideways in the appropriate direction while still maintaining forward pressure (3). Finally, once the

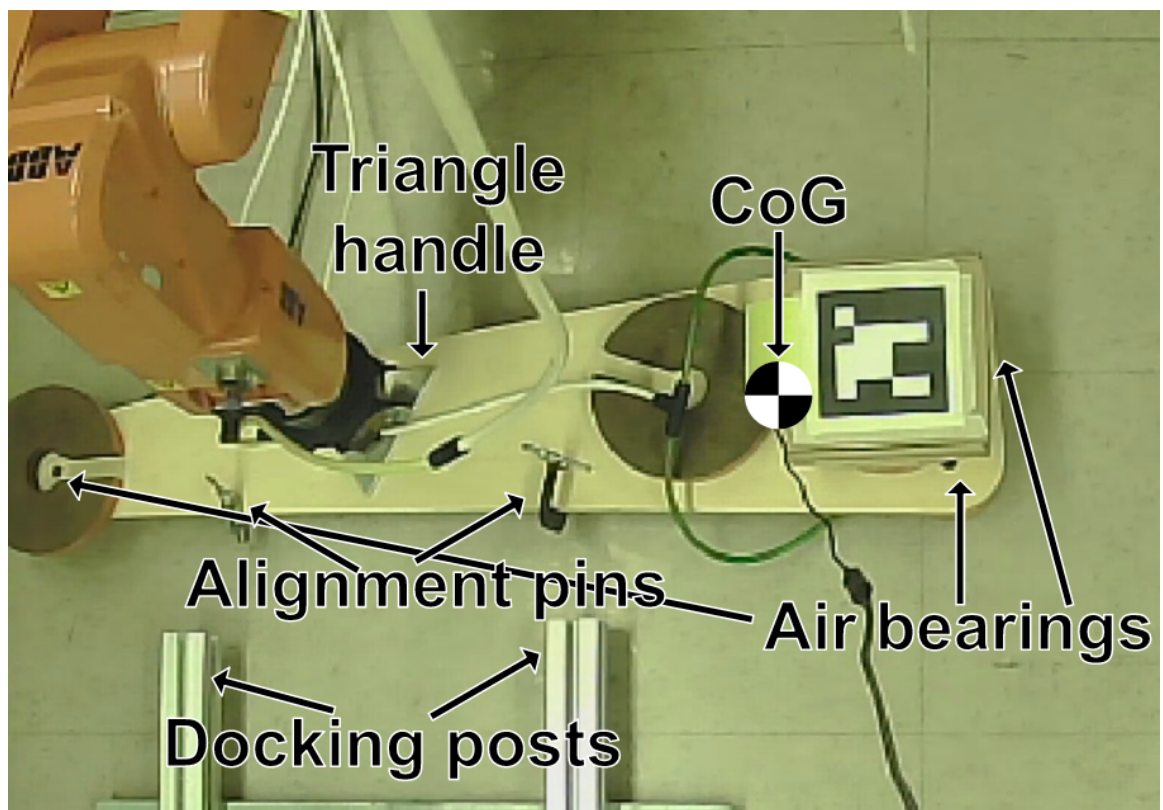


Figure 4: An overhead photo of the experimental setup for berthing.

second post made contact with the second clamp, the attractor was displaced forward (in the positive y direction), drawing the satellite into its berthing pose, satisfying multiple physical constraints.

After the task was proven possible under supervisory control, a state machine was designed to carry out the task autonomously. The process used by the state machine is very similar to that used by the human operator, but modified slightly for a sole reliance on force feedback with no supplementation by visual cues. Specifically, the system needs to determine whether the payload is aligned with the posts and, if not, in which direction it needs to move to correct the alignment.

This determination can be made using torque measurements at step 1.5 in Fig. 5. If the satellite is misaligned when first contact occurs, then the robot will measure a substantial torque. If this torque is positive, then it needs to move in the positive-x direction to correct the alignment. Conversely, if the torque is negative (as in the figure), then the payload will need to move in the negative-x direction (to the right in the figure).

After measuring this torque, the robot can continue to press forward until both posts are pressed against the satellite (Step 2). Next, the robot slides the satellite in the direction that was determined by the torque measurement in step 1.5. At the same time, it continues to apply forward pressure, as well as a slight torque in order to maintain contact with both posts. Once the ring contacts the second post (Step 3), the satellite is horizontally aligned and the berthing can be completed by simply pressing forward. Fig. 6 is a flowchart summarizing this algorithm. In the flowchart, τ_z is the torque about the z axis, F_x is the force along the x axis, F_{thresh} is the contact force threshold parameter, (15N for these experiments) and τ_{thresh} is the contact torque threshold. $PTWL(x, y, z, \phi_x, \phi_y, \phi_z)$ displaces the virtual attractor from the port of interaction by the specified dimensions. RWE moves the attractor back onto the port of interaction to eliminate contact forces. This state machine

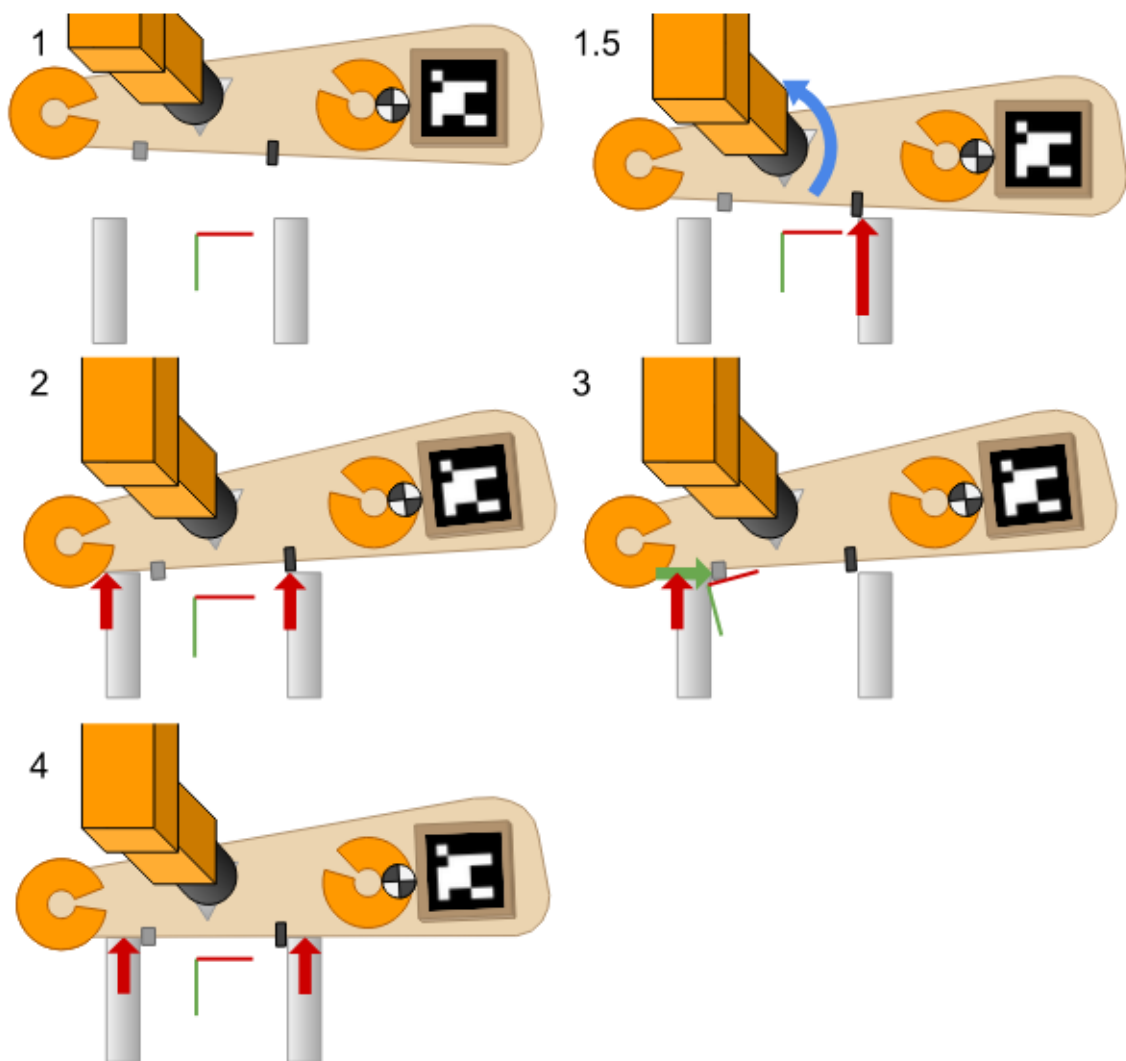


Figure 5: A step-by-step illustration of the berthing procedure.

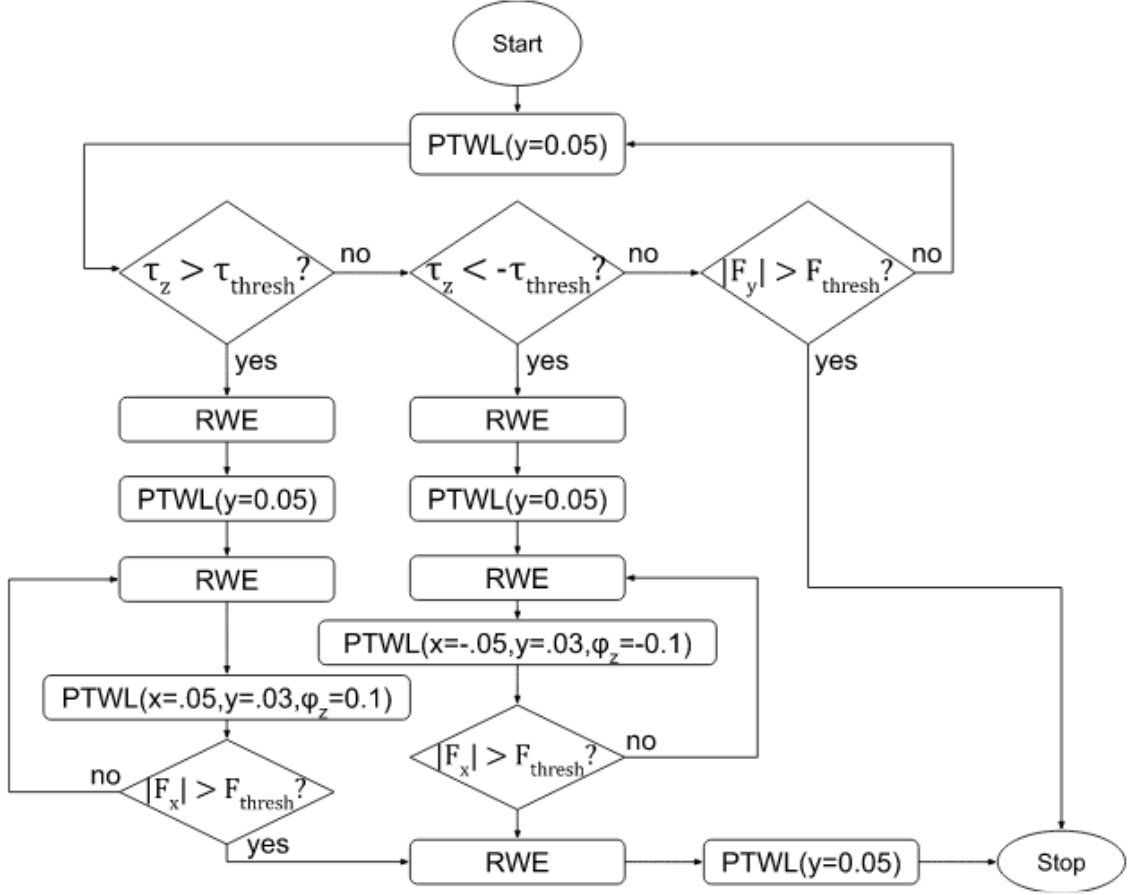


Figure 6: A flowchart of the berthing state machine.

was able to successfully and consistently berth the satellite even in the presence of significant perturbations.

A video of an operator manually berthing the satellite using compliant behaviors can be seen at [23] and a video of the same task conducted by a state machine constructed using compliant behaviors can be seen at [24].

To test the robustness of the berthing state machine, it was run repeatedly under initial conditions that varied by a 3cm maximum. This number is reasonable because it comfortably encompasses a human operator’s ability to “eyeball” the alignment before executing the state machine. The state machine was executed ten times in an automated loop. It was also automatically unberthed each time by a simple PTWL command. It would have been better to run this for more loops, perhaps 100 times



Figure 7: A plot of the attractor position and orientation during three automated berthing operations.

as [25] did with the tool stowage and retrieval state machine, but unfortunately this test was limited here by the air supply for the bearings. Fig. 7 shows the attractor pose over three loops. Fig. 8 shows the planar wrench for the same three tests.

In these plots, the distinct phases of the operation are clearly visible in the attractor motion profiles. Also notice that for each trial, the relative motion of the attractor is identical at the beginning, but diverges depending on the measured contact forces as the state machine adapts to the measurements.

3.2 Plug insertion

This task involved inserting a plug into a standard US electrical outlet under supervisory control. This example is representative of a common need for performing connections, and it also is conveniently familiar as a manual operation in terms of alignments and efforts required. It is also relatively easy for a robot under position control if the exact location of the outlet is known. Without this information, however, it is very easy for the robot to miss the outlet and generate large contact forces. These contact forces can be substantially reduced using supervisory control with un-

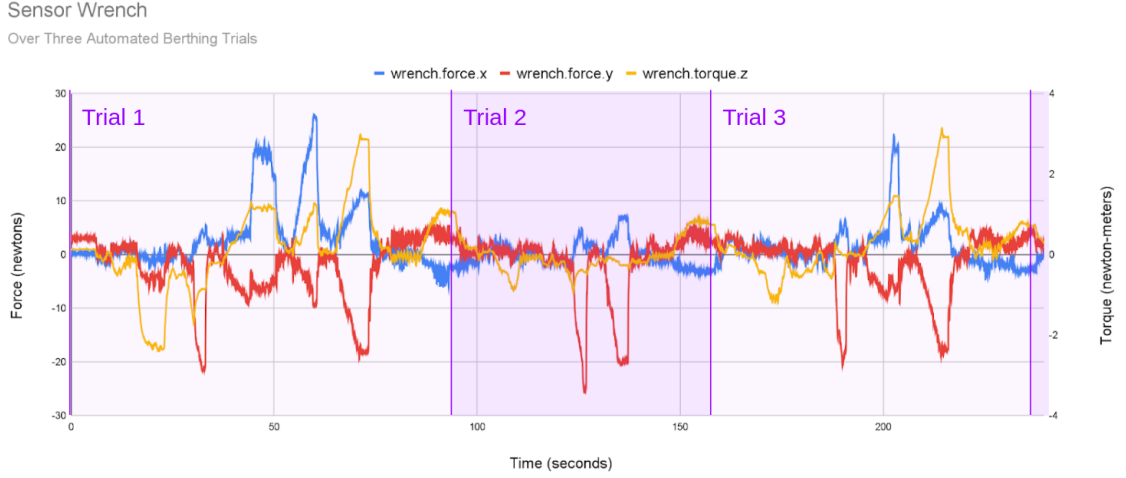


Figure 8: A plot of the measured forces and torque during three automated berthing trials.

derlying compliant-motion control. First, the operator roughly aligns the plug with the outlet, and moves toward the surface. Once contact has been made, the operator can clearly see the direction of the offset toward the hole. The plug is moved in roughly this direction while maintaining downward force until the hole is reached, at which point the plug can be inserted.

It was found that the behaviors presented here were sufficient to perform this task under supervisory control using soft attractors. First, the attractor is placed somewhere below the outlet surface until contact is detected. Next the attractor is moved to only slightly below the surface (to maintain moderate contact force) and translated in the direction of the outlet until the force in the y-direction crosses the contact threshold. Then the attractor can be placed deeply into the socket, tugging the plug into engagement.

Fig. 9 illustrates the steps for plug insertion. The upper axes represent the port of interaction and lower axes are the attractor pose. This process is forgiving of both spatial inaccuracies and temporal imprecision, since the different phases of the task are separated by discrete contact events. Fig. 10 shows a screen capture taken during the plug insertion process.

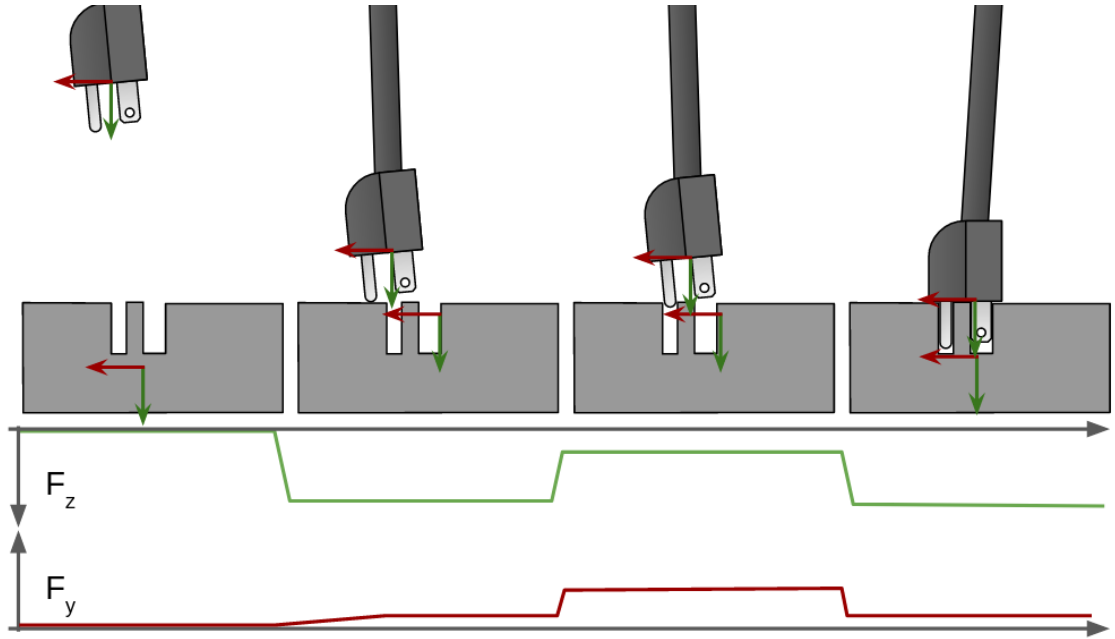


Figure 9: The steps involved in plug insertion

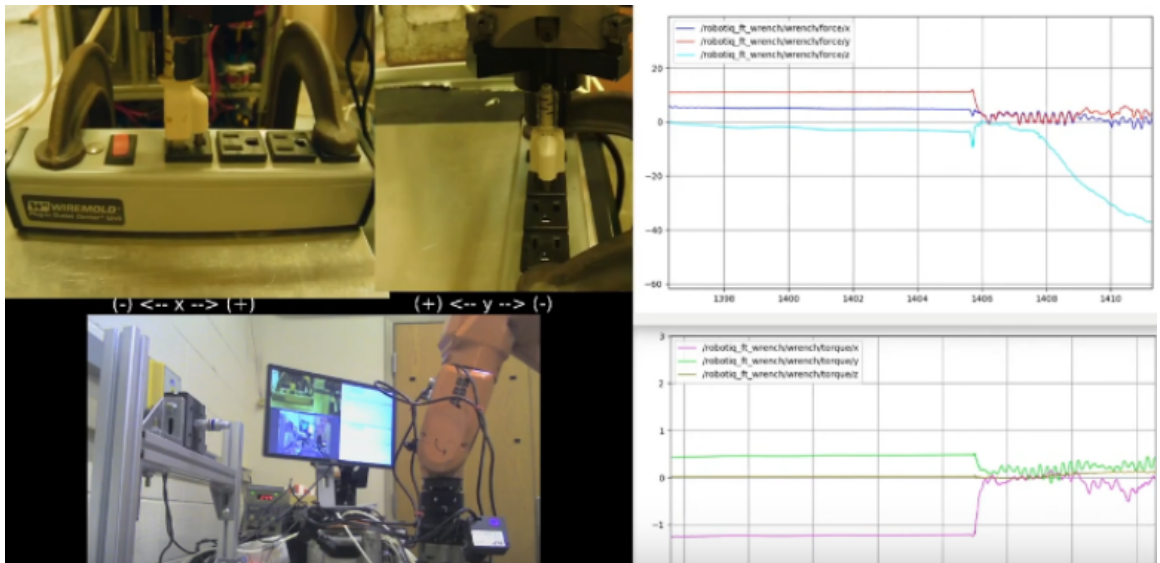


Figure 10: A screenshot taken during the plug insertion process.

3.3 Opening and closing a latched door

For this task, a small door analog was fabricated, consisting of a hinge, frame, door, and latch. To open the door, the robot must grab the door handle, turn the handle to unlatch the door, then translate and rotate through an arc centered around the door hinge axis. Closing the door is similar, but in reverse. This task is representative of kinematically-constrained manipulation tasks.

Just like the plug, this task is possible under strict position control, as long as the geometry of the hinge and latch are known precisely. This task would additionally require the generation of precise nonlinear trajectories, specifically gripper motion about a precise 6-DoF circular arc in space. Any inaccuracies would cause the forces/torques of interaction to become excessive. Under compliance, the task is dramatically simplified and intuitive. The robot operator must only know the approximate orientation of the door handle and the door hinge. The handle can be turned by imposing a rotational displacement of the soft attractor. Subsequently, the door can be opened by setting all virtual translational stiffnesses to near zero, then imposing an attractor rotation roughly parallel to the door hinge.

Fig. 11 illustrates this strategy. The left axes in each step are the attractor and the one on the robot is the port of interaction. Since translational stiffness was nearly zero, the robot accommodated with whatever translation was necessary to minimize the interaction efforts. Consequently, the robot conformed to the hinge’s kinematic constraint while opening the door.

A snapshot of the robot performing the door task is shown in Fig. 12.

A video of this operation can be seen at [26].

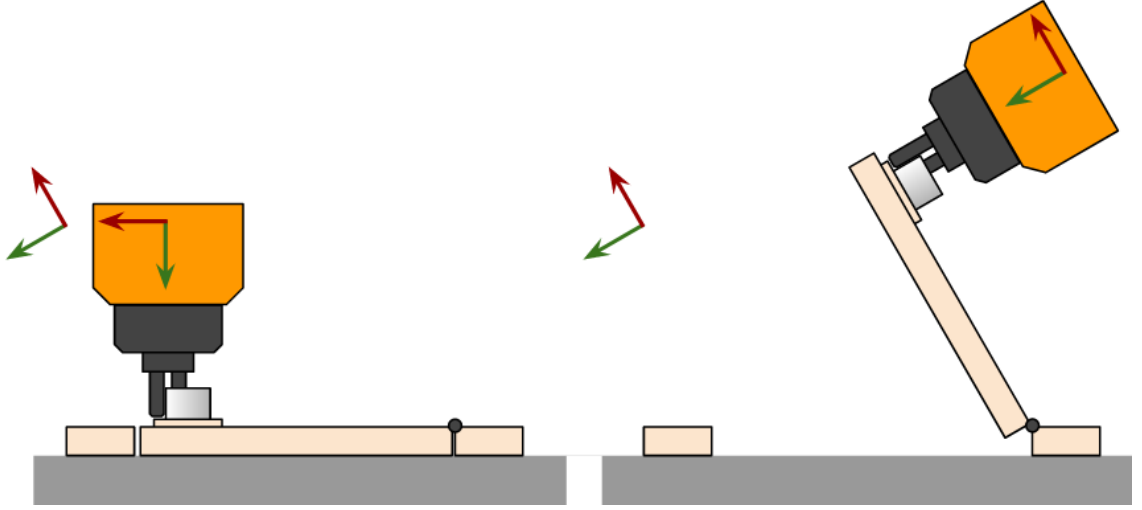


Figure 11: Attractor placement for door opening

3.4 Wide, shallow lid installation

This task involved fitting a wide disk into a hole, emulated by placing a lid on a saucepan. Though similar in principle to the sleeve-on-peg task, it is fundamentally different. In the sleeve-on-peg task, the biggest challenge is achieving the correct vertical orientation to prevent jams. In the saucepan task, the horizontal position is more important.

In Fig. 13, the outer radius of the smaller lid r_1 and the inner radius of the larger saucepan lip r_2 are exaggerated to show the results of slight misalignment. In this experiment, the lid should fit inside of the lip on the saucepan. The outer radius of the lid is r_1 and the inner radius of the saucepan lip is r_2 . If the offset d between the center of the lid and saucepan is less than $r_2 - r_1$, then the lid can be inserted straight down without any adjustments. Otherwise, the lid will contact the lip at two points when it is pressed downward. If d is less than $d_{max} = \sqrt{(r_2 - r_1)^2}$, then a torque will be generated about the axis going through both points, causing the lid to tilt toward the center. This tilt can be measured and used to calculate the direction of the offset. If d is greater than d_{max} , then no torque would be generated—only vertical

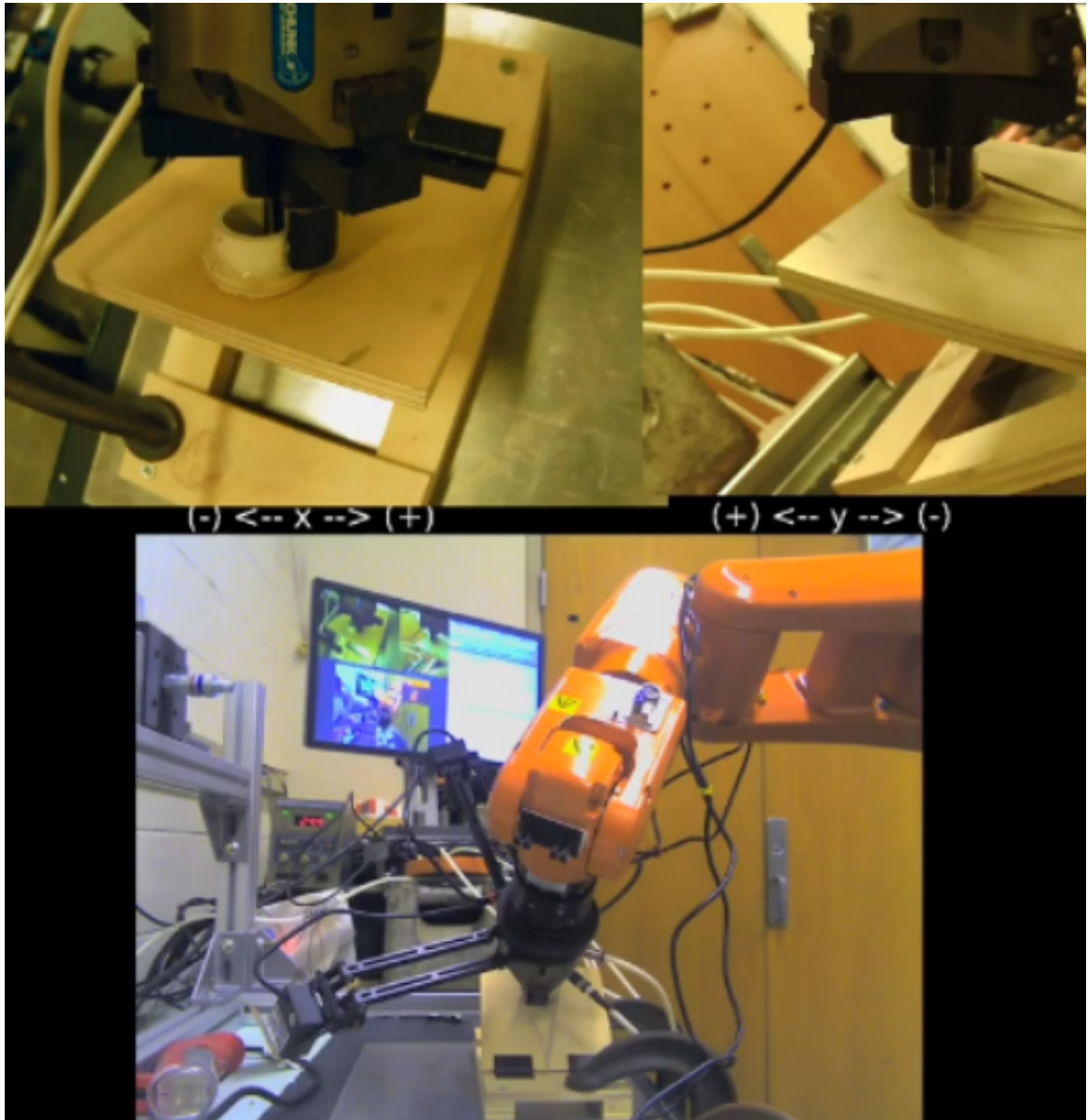


Figure 12: A screenshot taken during the door manipulation

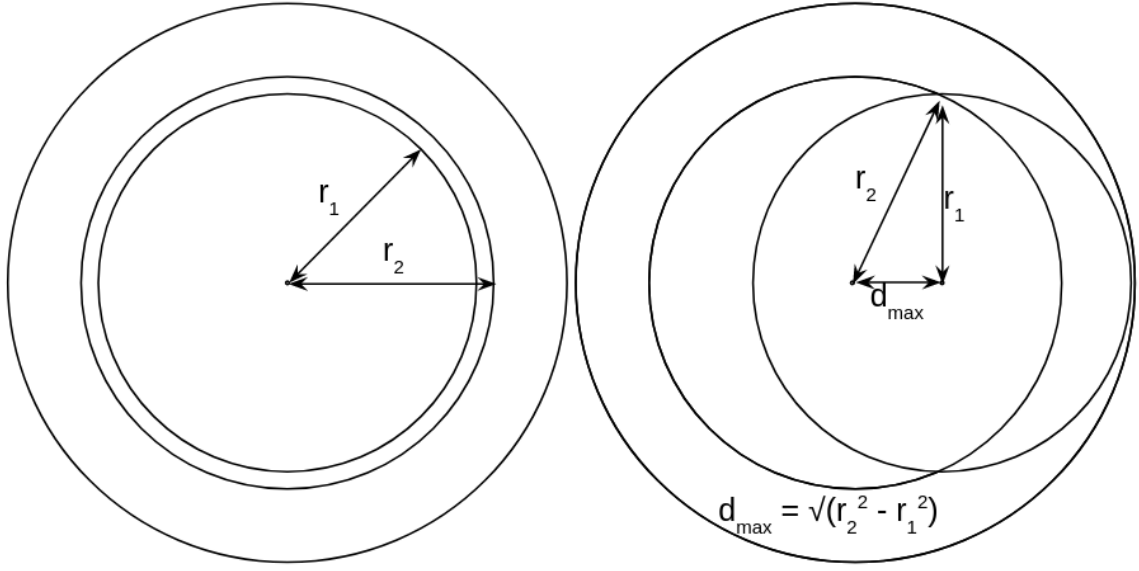


Figure 13: Exaggerated geometry of the saucepan and lid.

contact forces, since the center of the lid would be on the outside of the axis through the contact points, and would be pressed flat against the lip of the pan.

A state machine was created to place the lid on the pan after it was aligned within d_{max} . First, the attractor was moved down past the lip of the pan, causing the lid to contact the lip and tilt slightly. The attractor was then tilted and slid in the direction of the center of the pan until horizontal contact was detected with the opposing lip. After this contact, the attractor was reset to horizontal and pressed downward until the lid was horizontal and firmly in place. Fig. 14 illustrates the steps in the lid installation process. Again, the simple behaviors presented herein were sufficient to perform this task robustly, whether under supervisory control or autonomously via a state machine.

A video of this operation can be seen at [27]

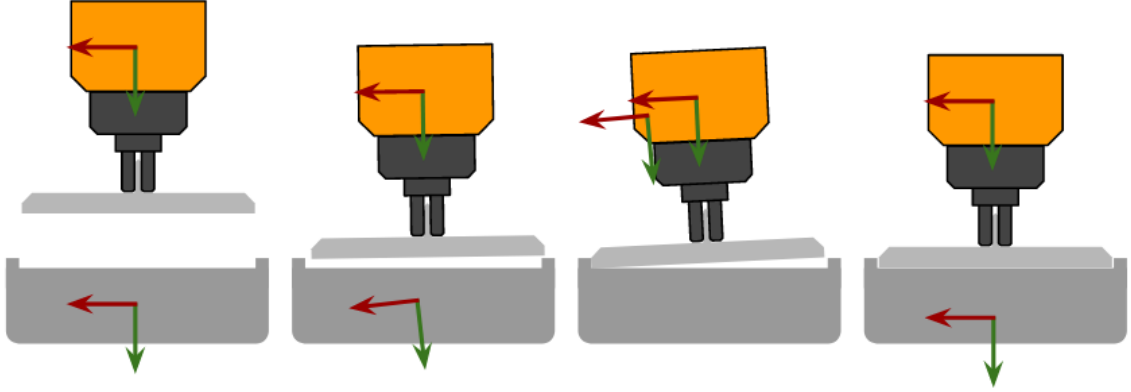


Figure 14: The steps for saucepan state lid placement.

3.5 Socket wrench insertion

The final task presented here used a consumer socket wrench tool-change system consisting of a shank and socket. The shank consists of a square-keyed shaft with a spring-loaded ball detent, and the socket is a mating square hole. Without the detent, the task is equivalent to a square-peg in square-hole insertion task. When inserting a peg into a hole with tight tolerances, a slight misalignment can generate contact forces that tend to amplify the original misalignment until the peg is jammed. In [25], it was demonstrated that a sleeve-on-peg task could be performed without any remote center of compliance by applying dithers (small horizontal perturbations) whenever a jam occurs in order to relieve friction.

Even under perfect alignment, when the shank is inserted, the ball will contact the edge of the socket. The vertical contact force dominates this interaction. Since it is offset from the axis of the applied force, a moment is induced, tilting the whole shank (clockwise in the figure) until it is jammed. Since the location of the detent on the shank is known beforehand, this jamming is predictable and can be counteracted in the same way every time, making indiscriminate dithering unnecessary.

By tilting the shank (counterclockwise in the figure), the ball detent can be depressed. Once it is depressed, the only vertical force resisting insertion comes from

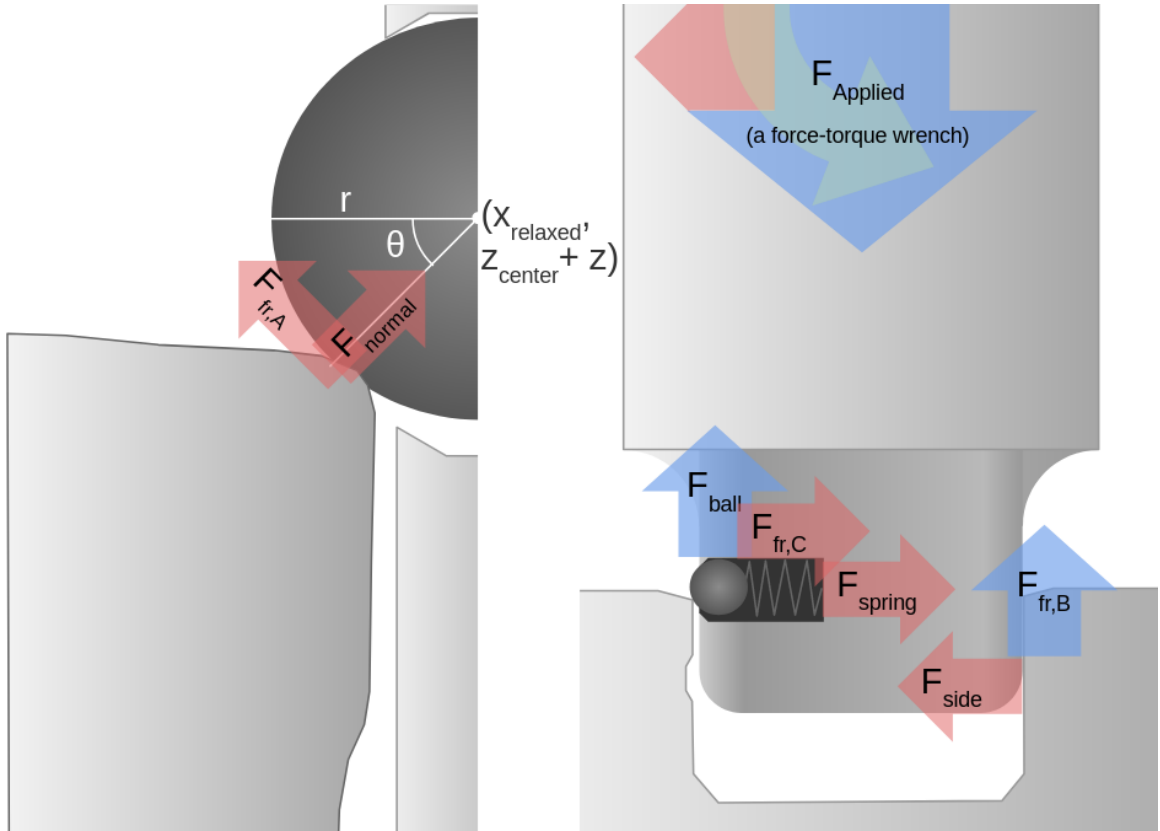


Figure 15: The initial contact forces during socket shank insertion

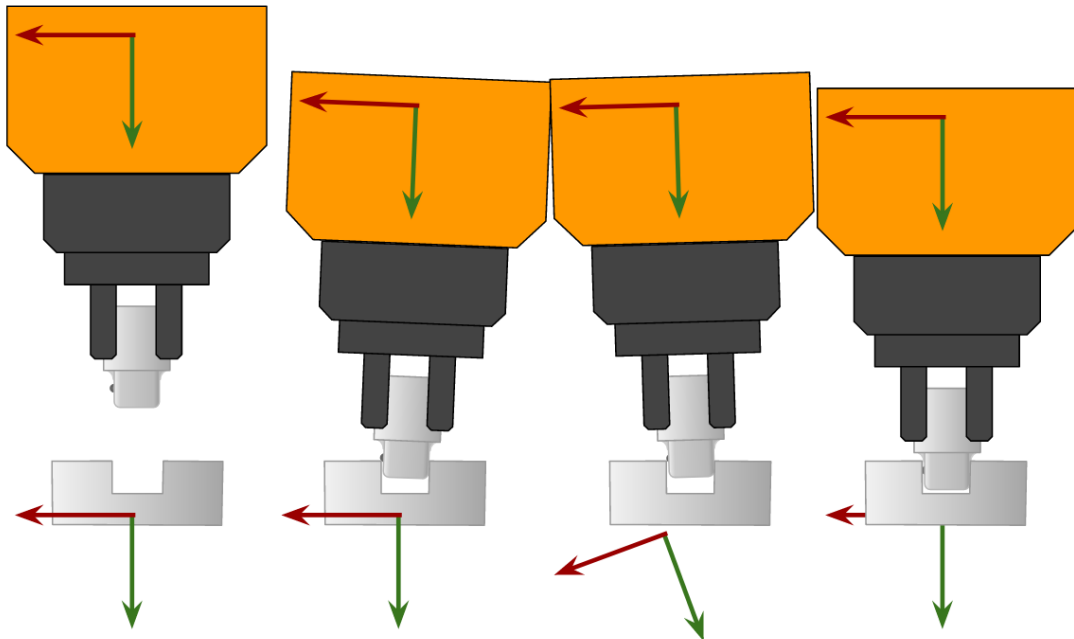


Figure 16: An illustration of the steps involved in shank insertion.

friction, and downward insertion becomes trivial.

Again, it was found that this operation could be performed using only the simple behaviors presented herein. A video of this operation can be seen at [28]

4 Capture

To service a satellite in orbit, a servicing vehicle will first need to capture it. This involves first approaching the satellite and matching its velocity as closely as possible. Next, the robotic arm will intercept the satellite and grasp it. If the velocities and inertia of the satellite are known precisely, then the slowdown maneuver can be done with careful motion planning to keep forces below acceptable thresholds. A major concern is that upon contact, the satellite will bounce away.

Compliant berthing and docking are not new ideas in this industry. The NASA Docking System used for the International Space Station has a “Soft Capture System” which utilizes both mechanical compliance and active control to ensure reliable docking [29]. The dynamics of robotic satellite capture have been extensively studied [30] and several authors have used compliant impedance/admittance control scheme [21].

Regardless of whether compliance is used after contact with the satellite, the interception of the satellite would be performed with trajectory control to ensure that the robot matches the satellite velocity as precisely as possible. It is then desired to transition “bumplessly” into compliance.

In a bumpless transition, there are no immediate changes in velocity when compliance mode is activated. The control law in Eqn. 5 can be rearranged to solve for the attractor position necessary to maintain force.

$${}^{task}\mathbf{o}_{attr} = {}^{task}\mathbf{o}_{ft} + (\mathbf{B}_{des}\mathbf{t}_{ft,des} - \mathbf{w}_{bf}) \quad (6)$$

This essentially means placing the virtual attractor some distance ahead of the interaction port in its path so that it can counteract the virtual damping forces. It might also be desirable to remove the w_{ft} (measured wrench) term to allow the

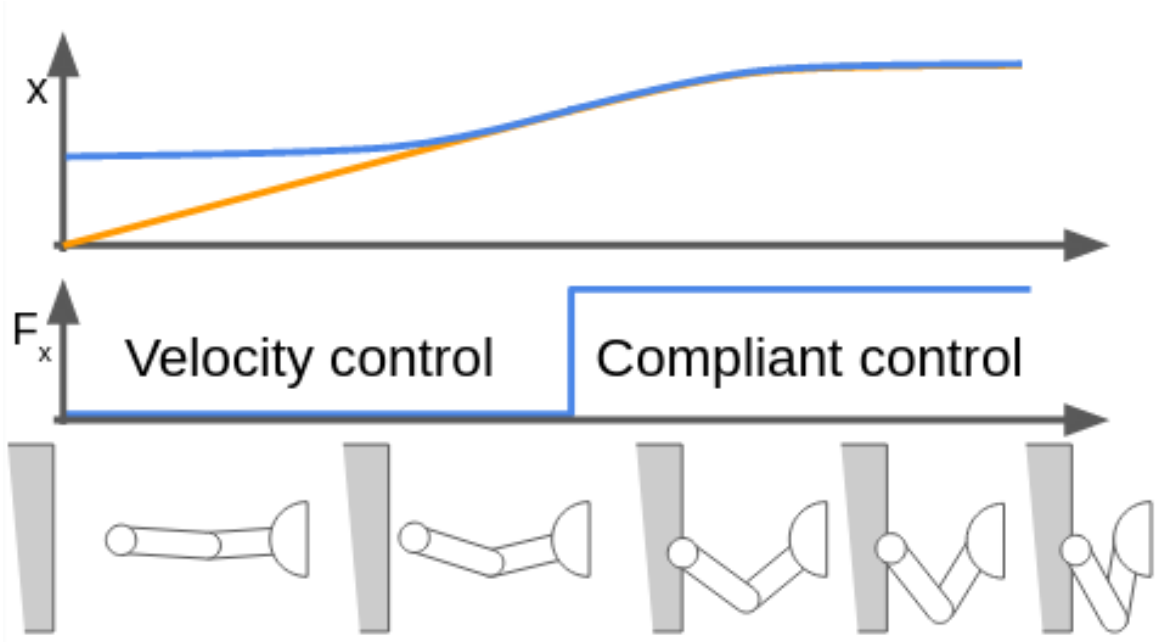


Figure 17: Hybrid capture with velocity control until contact and compliance thereafter. Yellow is the satellite position, and blue is the robot end-effector.

controller to react immediately to contact forces.

Alternatively, a new control law could apply $\mathbf{B}_{des}$ to the relative velocity of the robot with respect to the attractor, rather than the absolute velocity of the robot with respect to the earth.

$$\mathbf{t}_{ft,des} = \mathbf{B}_{des}^{-1} (\sim \mathbf{w}_{ft} + \mathbf{w}_{ft}) + \mathbf{t}_{ft,attr} \quad (7)$$

With this law, to transition bumplessly into compliance, the attractor needs only to be placed exactly on the port of interaction with the same velocity. Since our controller implementation was already velocity-based, this modification showed up as a velocity feed-forward term, as seen in Fig. 18.

To implement this, modified version of the accommodation controller code was created that accepts as control parameters the attractor pose, twist, as well as a third enumerated variable that specifies which control law to use.

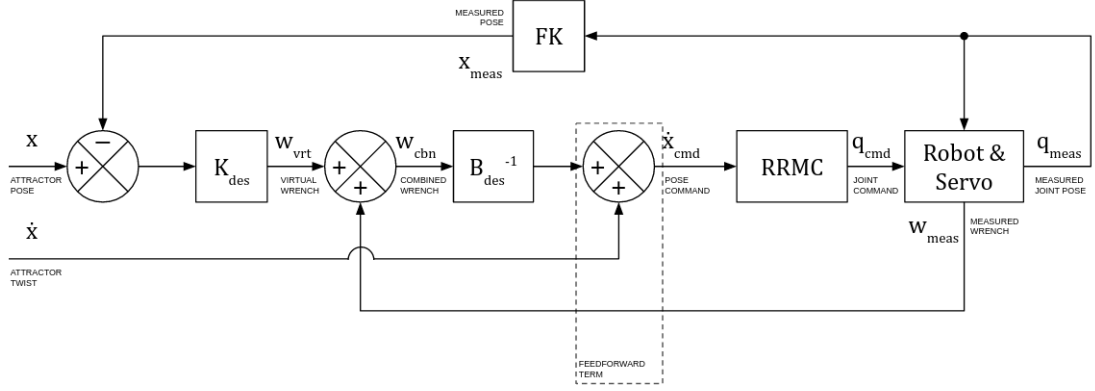


Figure 18: A schematic diagram of the modified accommodation controller.

4.1 Simulation

For validation, the controller was tested in a variety of purely virtual and hardware-in-the-loop simulations. First, in a purely virtual, one-dimensional simulation, various controller schemes were tested under various conditions to see which produced the most desirable results. The primary evaluation criteria was the contact time, defined as the total time period during which the satellite is in contact with the robot manipulator. Maximizing this time ensures that the robot’s gripper will have enough time to close and grasp the satellite. Also of interest are the maximum contact forces, defined as the peak measured force during capture. This should be minimized to avoid stressing the hardware and to reduce the propensity to bounce.

4.1.1 Fixed Vehicle

The first set of simulations considered the servicing vehicle as fixed to an inertial frame. This is a simplifying assumption and is still accurate when the vehicle is sufficiently more massive than the satellite being captured. The robot was represented as a 1-dimensional actuator with a low-level servo loop and with a virtual force sensor that reports the contact forces exerted between the robot end-effector and the satellite. The satellite itself was represented as a mass in 1-dimensional motion, with

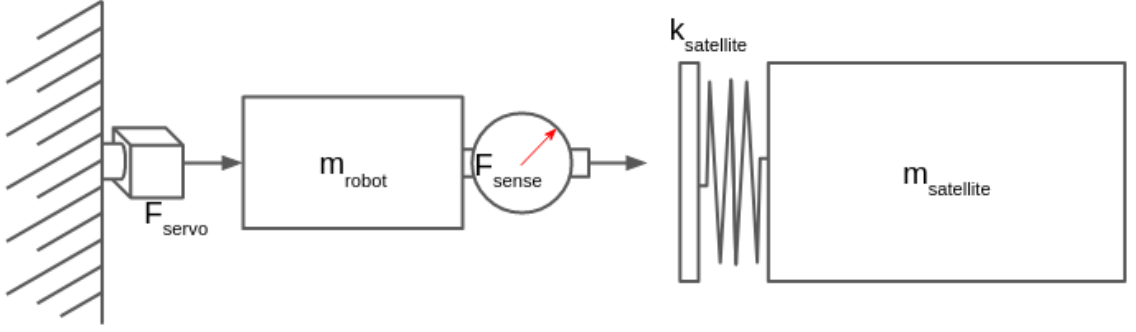


Figure 19: A schematic representation of the physical model used for initial simulation.

no rotational-inertial properties. Contact forces were generated using a reductive, purely linear-elastic model. Under this model, contact force is proportional to the displacement between the satellite and the robot when they overlap, and zero when the satellite is to the right of the robot. The simulation was carried out using Euler's method.

Under these simplifications, the entire state of the system can be represented as using 4 state variables and one input. Let the state be represented as

$$\mathbf{X} = \begin{bmatrix} x_s(t) \\ x_r(t) \\ \dot{x}_s(t) \\ \dot{x}_r(t) \end{bmatrix} \quad (8)$$

Then the state evolution follows

$$\dot{\mathbf{X}} = \begin{bmatrix} \dot{x}_s(t) \\ \dot{x}_r(t) \\ \ddot{x}_s(t) \\ \ddot{x}_r(t) \end{bmatrix} = \begin{bmatrix} \dot{x}_s(t) \\ \dot{x}_r(t) \\ \frac{F_{contact}}{m_{satellite}} \\ \frac{F_{servo} - F_{contact}}{m_{robot}} \end{bmatrix} \quad (9)$$

The servo force was computed using a PD servo controller, that takes as input the desired robot pose and twist, which, in this case, is just a position and velocity.

$$F_{servo} = K_P(x_{des} - x_{robot}) + K_D(\dot{x}_{des} - \dot{x}_{robot}) \quad (10)$$

The constants K_P and K_D were selected such that the servo system was critically damped with a natural frequency of 1Hz.

The contact force was defined piecewise based on the relative positions of the robot and the satellite.

$$F_{contact} = \begin{cases} 0 & \text{if } x_{robot} < x_{satellite} \\ k_{satellite}(x_{satellite} - x_{robot}) & \text{if } x_{robot} \geq x_{satellite} \end{cases} \quad (11)$$

Under these definitions, the entire system is piecewise-linear, and can be represented using two distinct linear systems.

$$\dot{\mathbf{X}} = \begin{cases} \mathbf{A}_f \mathbf{X} + \mathbf{B} \mathbf{U} & \text{if } x_{robot} < x_{satellite} \\ \mathbf{A}_c \mathbf{X} + \mathbf{B} \mathbf{U} & \text{if } x_{robot} \geq x_{satellite} \end{cases} \quad (12)$$

In this representation, the state matrix, A , is the only thing that changes between the free (A_f) and contact (A_c) case.

$$\mathbf{A}_f = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{K_P}{m_{robot}} & 0 & \frac{K_D}{m_{robot}} \end{bmatrix}, \mathbf{A}_c = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{-k_{satellite}}{m_{satellite}} & \frac{k_{satellite}}{m_{satellite}} & 0 & 0 \\ \frac{k_{satellite}}{m_{robot}} & \frac{K_P - k_{satellite}}{m_{robot}} & 0 & \frac{K_D}{m_{robot}} \end{bmatrix} \quad (13)$$

The input matrix, B is constant. The input vector, U is just the servo input velocity and position.

$$\mathbf{U} = \begin{bmatrix} x_{des} \\ v_{des} \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{K_P}{m_{robot}} & \frac{K_D}{m_{robot}} \end{bmatrix}, \quad (14)$$

At the start of the simulation, the satellite is coming toward the robot from the right at a constant velocity.

Three control schemes were used for simulated capture. The first was direct motion control. In this method, the force sensor was disregarded entirely and the “attractor” motion was sent directly to the motion controller. A motion planner takes the initial position and velocity of the satellite and computes a trajectory for the robot such that their velocities match upon contact. The satellite’s initial velocity is v_{s0} and the robot starts at $x_{r0} = 0$. For optimal workspace usage, x_{r0} should also be at the workspace limit closest to the satellite’s initial position. To eliminate contact forces, the robot should be moving at the same velocity as the satellite when they come into contact. This condition can be rewritten as:

$$\begin{cases} x_r(t_c) = x_s(t_c) \\ v_r(t_c) = v_s(t_c) \end{cases} \quad (15)$$

Where t_c is the time at which contact occurs. If a maximum acceleration is assigned to the empty-handed robot a_r then the optimal trajectory is triangular, given by:

$$\begin{cases} v_r(t) = \int_0^t a_r dt = a_r t \\ x_r(t) = \int_0^t a_r d^2t = \frac{1}{2} a_r t^2 \end{cases} \Bigg|_{t \leq t_c} \quad (16)$$

The constants of integration are all zero by definition. The contact time can be derived using Eqn. 15.

$$t_c = \frac{a_r}{v_s} \quad (17)$$

The initial position of the satellite can be found using Eqn. 15:

$$x_{s0} = -\frac{v_{s0}^2}{2a_r} \quad (18)$$

While the position of the satellite cannot be controlled, the start time $t = 0$ can be defined such that $x_{s0} = x_s(0)$.

After contact, the robot follows a triangular stopping trajectory to maintain the contact force.

$$\left\{ \begin{array}{l} v_r(t) = -a_c(t - t_c) + v_{s0} \\ x_r(t) = \frac{1}{2}a_r(t - t_c)^2 + v_{s0}(t - t_c) + x_c \end{array} \right| t \geq t_c \quad (19)$$

This is similar to the approach trajectory, but with a smaller acceleration, a_s . This acceleration can be determined by the desired stopping force given by

$$a_s = F_s/m_s \quad (20)$$

Where m_s is the mass of the satellite.

With a perfect control system and perfect knowledge of the satellite's inertia and motion, direct trajectory control would work perfectly like in Fig. 17 and the satellite would never bounce. In reality, however, even slight uncertainties in position or velocity can make drastic differences in contact forces. Fig. 20 shows the result of a simulation where strict trajectory control was used.

In the position plot, the satellite and robot appear to coincide for the most part. This is expected since the trajectory is being controlled here. The force plot, however, is more revealing. It shows two distinct contact periods; the satellite makes initial contact and bounces twice before being ejected forever.

In the second simulation, strict trajectory control was used only while the robot was moving unimpeded through free space. Upon making contact, the robot switched

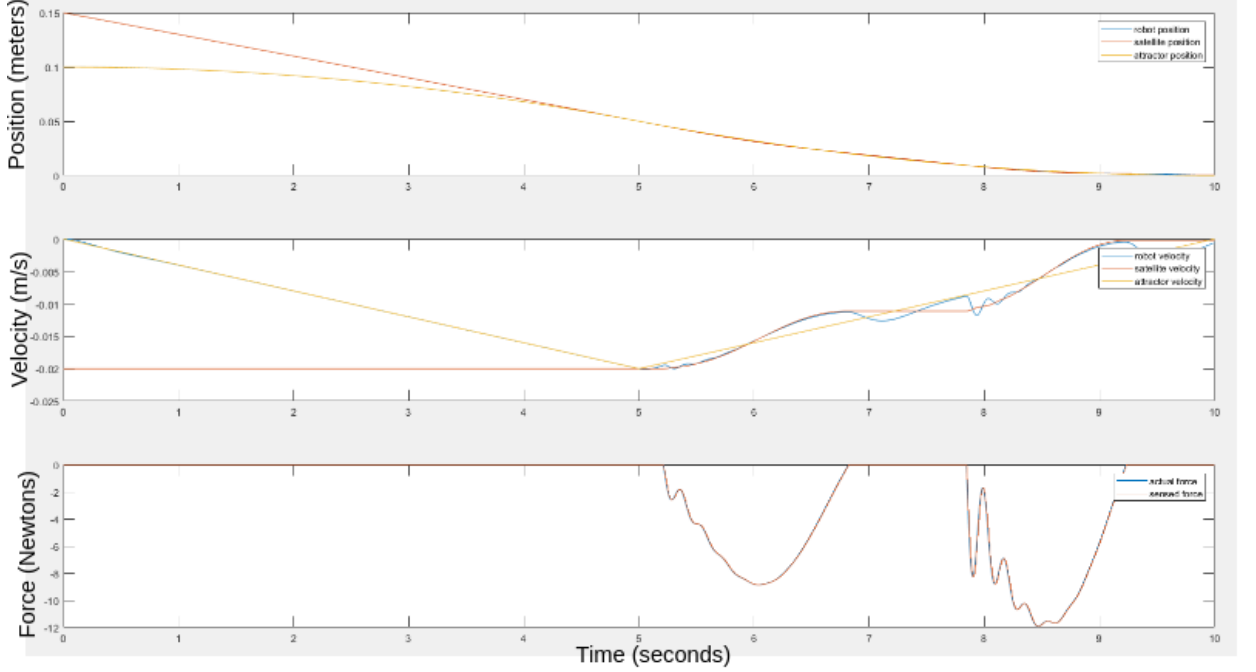


Figure 20: Simulated 1-D capture under strict trajectory control.

bumplessly into compliant control. After the transition, the attractor followed the same triangular stopping trajectory from the previous simulation. A well-designed accommodation controller should be able to do a better job of regulating the contact forces than the robot’s base servo loop.

One of the limiting factors for admittance controller performance is the update rate of the force sensor, or more directly the latency. This is especially important for capture because if the accommodation controller cannot react to the contact event before the satellite bounces away, then it does not serve any purpose. In order to accurately simulate the control law, the accommodation controller’s control law was implemented in an inner loop that only updated at 100Hz. This inner loop also contains the Boolean condition that detects when contact occurs.

Under this control scheme, the robot immediately performed capture much better than it did under strict trajectory control, as expected. With the same initial conditions as the previous simulation, the satellite did not even bounce once. It made initial contact, but after some oscillations the force eventually settled to F_s .

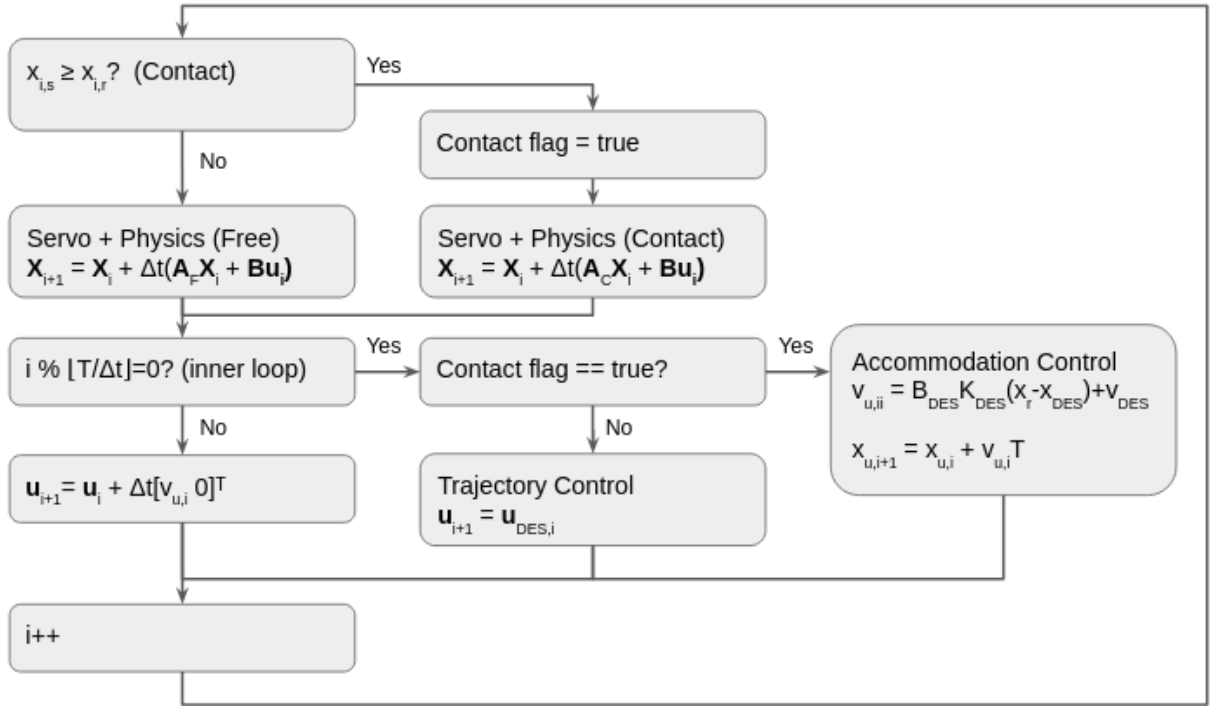


Figure 21: Flowchart of the simulation loop.

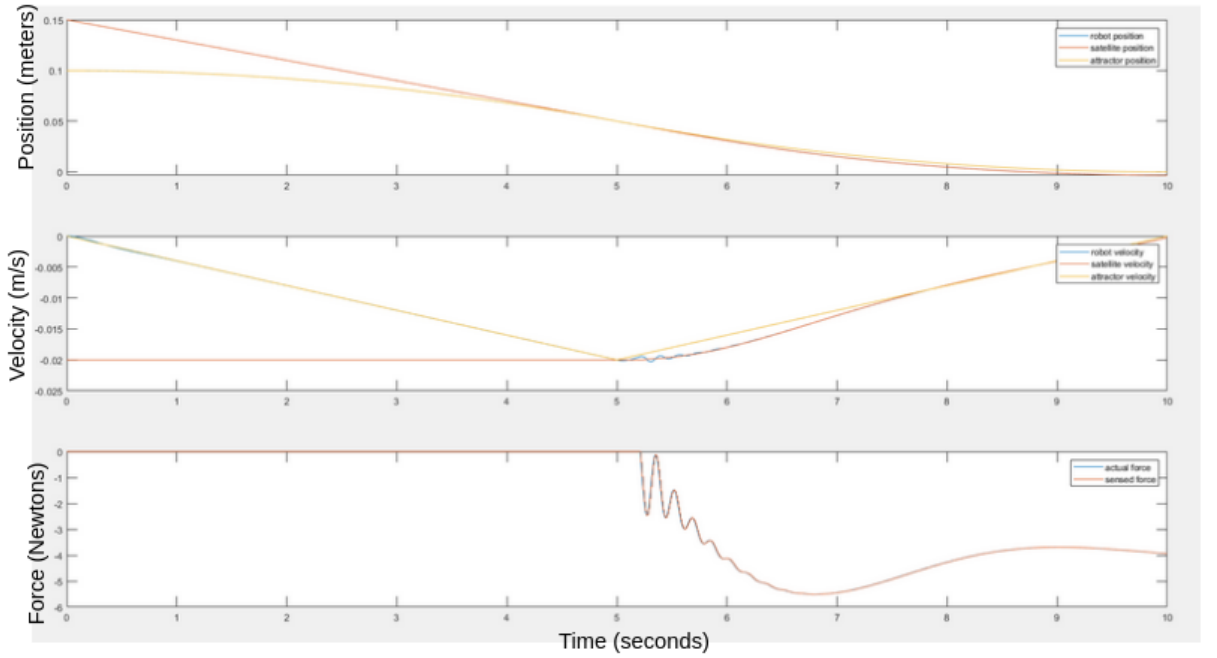


Figure 22: Simulated 1-D capture with a bumpless transition into compliance.

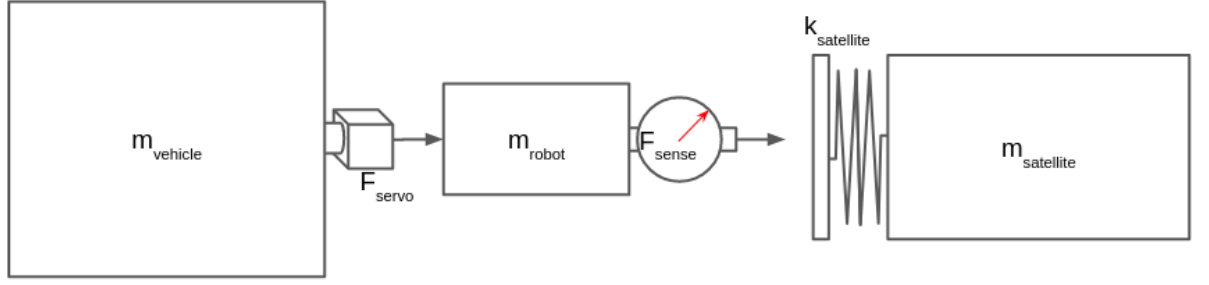


Figure 23: The simulated system with a free-floating robot base.

4.1.2 Free vehicle

The next step in producing higher-fidelity simulations was to add a free-floating robot base, since the actual servicing vehicle would itself be a free-floating mass. A schematic representation of this system is displayed in Fig. 23

This new configuration resulted the addition of two more state variables to the system, x_v and $\dot{x}_v$ for the vehicle's position and velocity, respectively.

$$\mathbf{X} = \begin{bmatrix} x_s(t) \\ x_r(t) \\ x_v(t) \\ \dot{x}_s(t) \\ \dot{x}_r(t) \\ \dot{x}_v(t) \end{bmatrix} \quad (21)$$

The only change in the state equation is that now the servo force acts equally and oppositely on the robot and the vehicle. Also, the reference for the servo loop is no longer the absolute position of the robot arm, but rather the relative position of the robot with respect to the vehicle, as shown in Eqn. 22. This affects the state matrices and the input matrix. The new state matrices are given in Eqn. 23. Note that these

new $\mathbf{A}$ matrices are now 5x5 and include terms for the motion of the vehicle. These new matrices are still applied according to 12.

$$F_{servo} = K_P(x_{des} - x_{robot} + x_{vehicle}) + K_D(\dot{x}_{des} - \dot{x}_{robot} + \dot{x}_{vehicle}) \quad (22)$$

$$\mathbf{A}_f = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{K_P}{m_r} & \frac{-K_P}{m_r} & 0 & \frac{K_D}{m_r} & \frac{-K_D}{m_r} \\ 0 & \frac{-K_P}{m_v} & \frac{K_P}{m_v} & 0 & \frac{-K_D}{m_v} & \frac{K_D}{m_v} \end{bmatrix}, \mathbf{A}_c = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{-k_s}{m_s} & \frac{k_s}{m_s} & 0 & 0 & 0 & 0 \\ \frac{k_s}{m_r} & \frac{K_P - k_s}{m_r} & \frac{-K_P}{m_r} & 0 & \frac{K_D}{m_r} & \frac{-K_D}{m_r} \\ 0 & \frac{-K_P}{m_v} & \frac{K_P}{m_v} & 0 & \frac{-K_D}{m_v} & \frac{K_D}{m_v} \end{bmatrix} \quad (23)$$

As mentioned before, the mass of the vehicle is generally larger than that of the satellite. For the “fixed vehicle” simulations to be accurate, the vehicle must also be much *much* heavier than the robot arm, especially when it is under strict trajectory control. This is because the servo force is pushing on both the robot and the vehicle. If the robot’s mass is even barely noticeable compared to the vehicle mass, then it will perturb the position of the vehicle before contact occurs. This will cause a misalignment between the predetermined trajectory and the actual trajectory of the satellite. As was shown in the previous simulations, even the slightest misalignment can drastically increase contact forces and the likelihood of bouncing.

In the following simulation, all the parameters and initial conditions were retained from the previous simulation with a bumpless transition into compliance. The only difference was the addition of the mobile robot base.

Despite the use of compliance, the initial velocity error at contact was too great

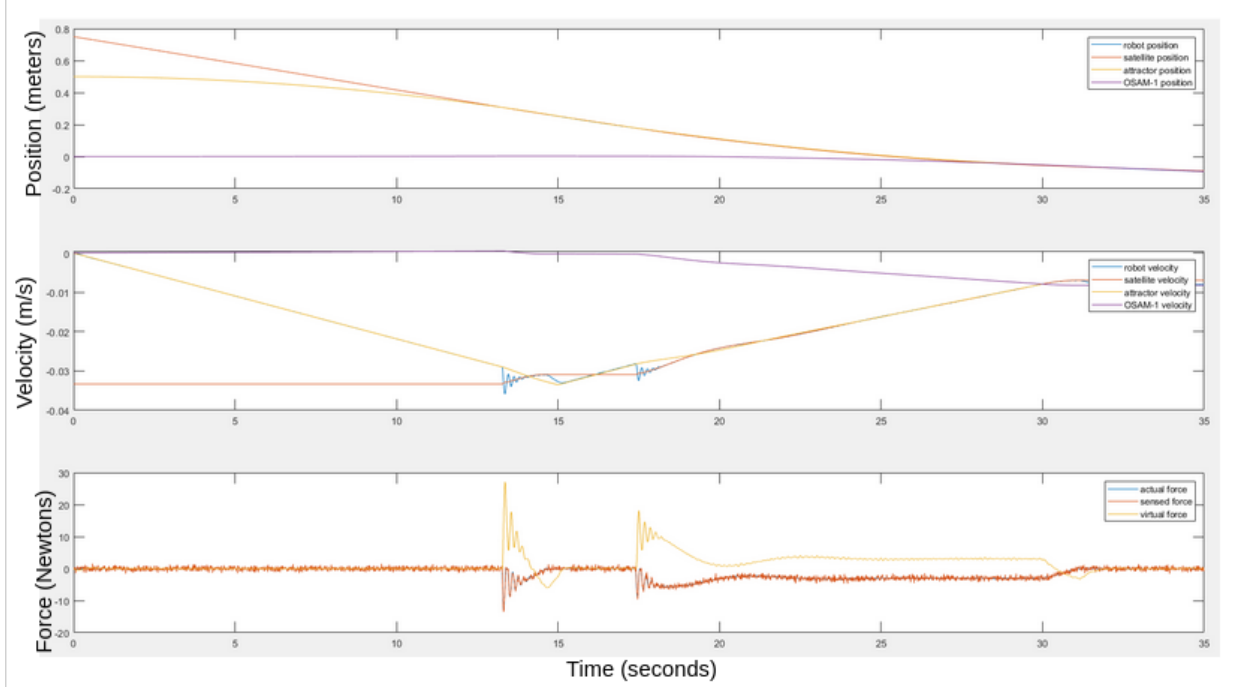


Figure 24: Simulated 1-D capture with a bumpless transition into compliance but with a free robot base

to virtually accommodate. There was an initial bounce, where contact forces reached thirty Newtons; much greater than even the contact forces measured in Fig. 20. Despite the initial bounce, the robot eventually recaptured the satellite, and successfully sustained contact for around twelve seconds before it reached its workspace limits.

Unfortunately, even with virtual compliance, it was impossible to eliminate large contact forces that result from a substantial velocity mismatch at contact. Part of the reason is that the internal compliant control loop could not react within the time frame in which the initial contact event occurred. Even with an ideal, zero-latency natural admittance controller, it would be impossible to passively achieve the necessary behavior, because the inertia of the robot itself would get in the way.

4.1.3 Deliberate distance overestimation

Any control decision to reduce contact forces must be executed before the contact event occurs. On the surface, it seems that the interception trajectory was already op-

timally generated to eliminate contact forces. This trajectory was generated without any consideration for uncertainty.

In the model, the measurement uncertainty is symmetric; the velocity and position are just as likely to be underestimated as overestimated and the trajectory is planned based on the expected value. The effect of the measurement error, however, is asymmetric. If the initial distance to the satellite is underestimated then the robot will intercept it early, while it is still speeding up to match the satellite’s velocity. This will cause the satellite to bounce off prematurely, and possibly never come into contact with the satellite a second time.

If the initial distance is underestimated, however, the satellite will come into contact while the robot is already executing the stopping trajectory. Once contact occurs, the robot will already be slowing down, and the actuators will dissipate the satellite’s kinetic energy. This outcome is clearly more desirable than the underestimation case. So, deliberately starting the capture maneuver late should increase the likelihood of this outcome. The results of such a maneuver are illustrated in Fig. 25.

Due to the deliberate velocity mismatch, the contact forces were actually higher in this example than they were in the previous bumpless attempt. Despite this, the contact force was much more homogeneous, consisting of a single continuous contact and no bounces.

Several other “tricks” were tested for viability in capture operations with varying degrees of effectiveness, though none were better than the “distance overestimation” technique discussed above. Nonetheless, in the spirit of reporting negative results, here are the results of the other methods tried.

4.1.4 Teleporting Attractor

The first strategy to maintain contact was to immediately teleport the attractor inside of the satellite upon contact. The nominal trajectory was maintained until contact

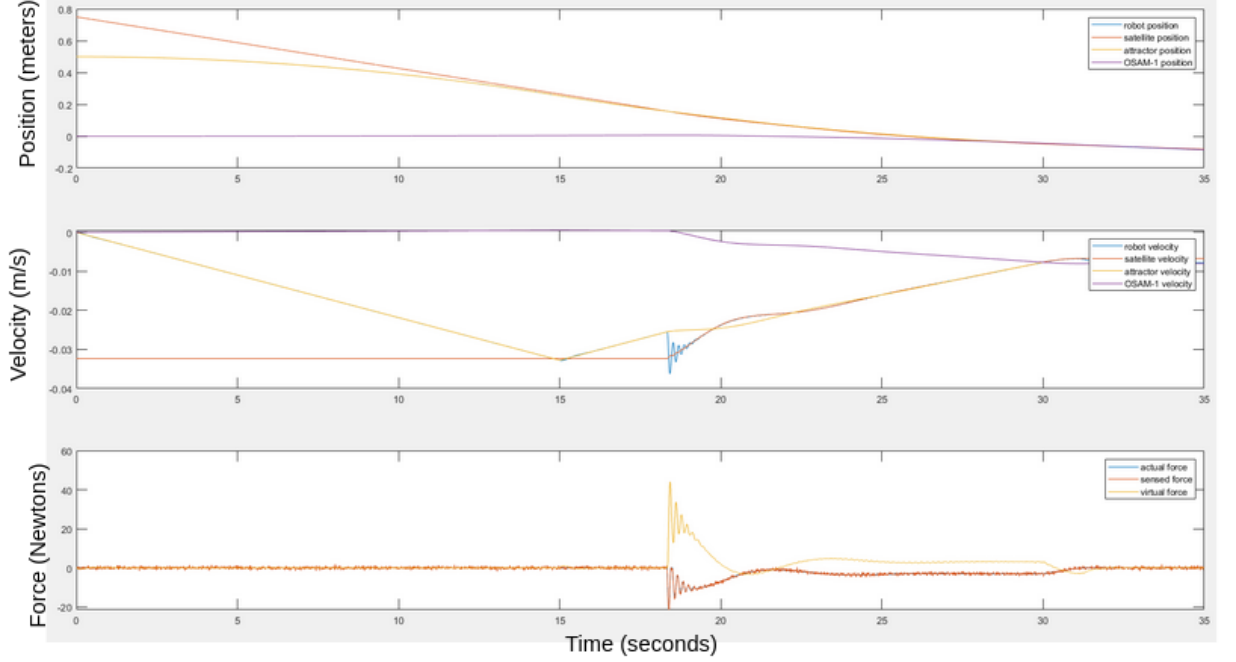


Figure 25: Simulated 1-D capture with a bumpless transition into compliance but with a free robot base, where the distance to the robot is deliberately underestimated.

occurs, and after contact the position of the attractor was the nominal position plus an offset, as shown in Eqn. 24.

$$x_r(t) = \frac{1}{2}a_r(t - t_c)^2 + v_{s0}(t - t_c) + x_c + x_{offset} \text{ and } t \geq t_c \quad (24)$$

A sudden position change like this causes a singularity in the trajectory. To address this, the singularity in the velocity profile was ignored, leaving it identical to velocity of the nominal trajectory. Immediately upon contact, the displaced attractor should result in a net virtual pushing into the satellite. The idea behind this strategy is that the virtual force will help the robot to follow the satellite as it bounces away. Fig. 26 shows the results of using this method with the same initial conditions as the previous simulation.

This strategy had a negative effect on the capture behavior overall. The peak contact forces increased dramatically, and the satellite bounced away after only a few seconds, never to make contact again.

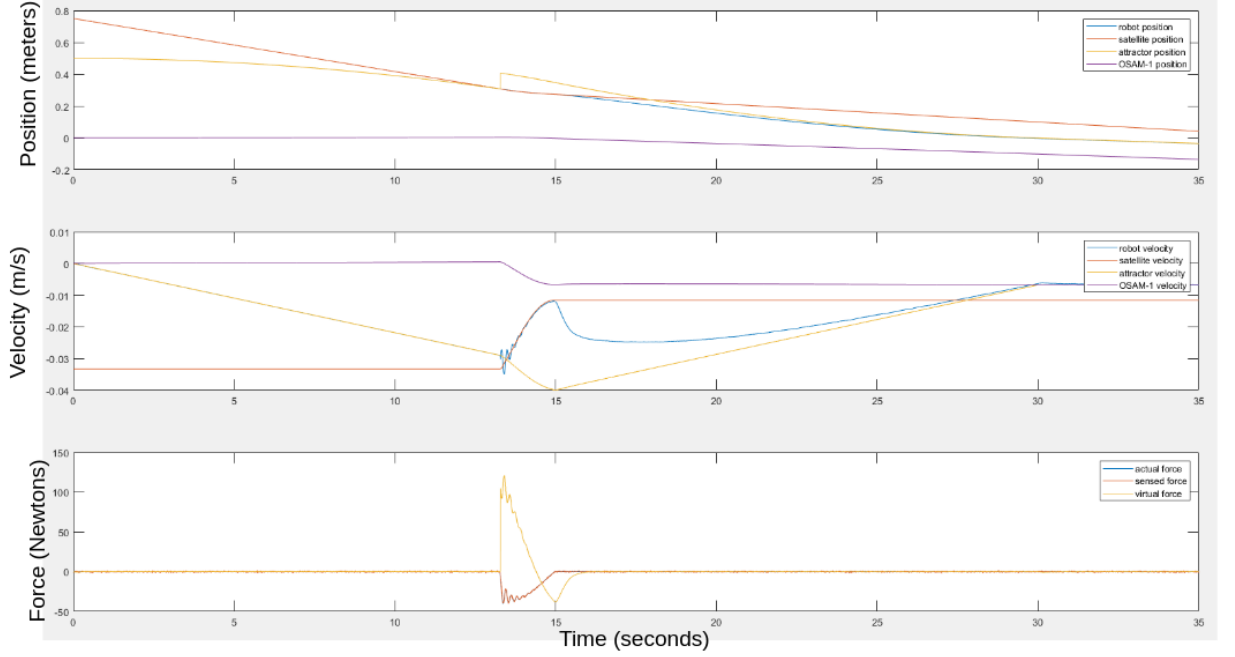


Figure 26: Simulated 1-D capture with a bumpless transition into compliance but with a free robot base, where the attractor was teleported forward upon contact.

This may be due to the fact that once the attractor was only initially inside of the satellite, so the imparted force depleted rapidly as the robot returned to the attractor. Once the attractor and robot once again coincided, the virtual force was zero and the robot was no longer pushing into the satellite. Also, the nominal, triangular trajectory was designed to impart a constant stopping force on the satellite. If that force is not going to be constant then this trajectory is useless, if not counterproductive.

4.1.5 Positive Feedback Attractor

One possibility is to use the robot's position itself as a reference, instead of the pre-planned trajectory. The attractor motion can be defined as always a fixed displacement from the robot. In principle, this should be equivalent to direct force control; the force can be precisely controlled by varying the distance between the attractor and the robot.

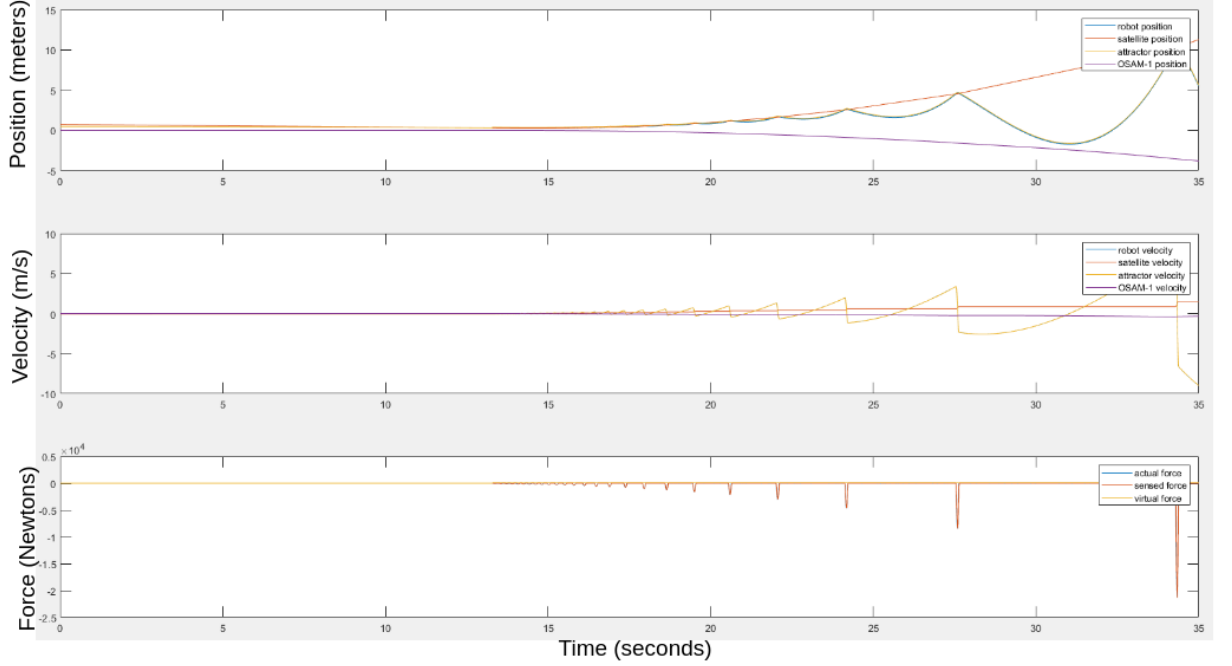


Figure 27: Simulated 1-D capture with a bumpless transition into compliance but with a free robot base. The attractor motion is equal to the robot motion plus a constant displacement.

$$\ddot{x} = F/m = \frac{k}{m}(x - x_{attr}) + \frac{d}{m}(\dot{x} - \dot{x}_{attr}) = \frac{k}{m}(x_{offset}) \quad (25)$$

In reality, however, there are several problems with this approach. The first is that the robot's motion already depends on the attractor motion, so making the attractor motion depend on the robot motion results in another, higher-level control loop. Moreover, this loop exists in positive feedback with itself.

In the simulation shown in Fig. 27, the triangular trajectory was completely discarded upon contact, and the attractor was assigned the velocity and motion of the actual robot plus a fixed displacement. As anticipated, it was unstable and blew up almost immediately due to a positive feedback relationship between the nested laws governing the robot and the attractor.

Part of the reason for the blowup was the coupling between the force control and the physical dynamics. For example, when the satellite bounced after initial contact

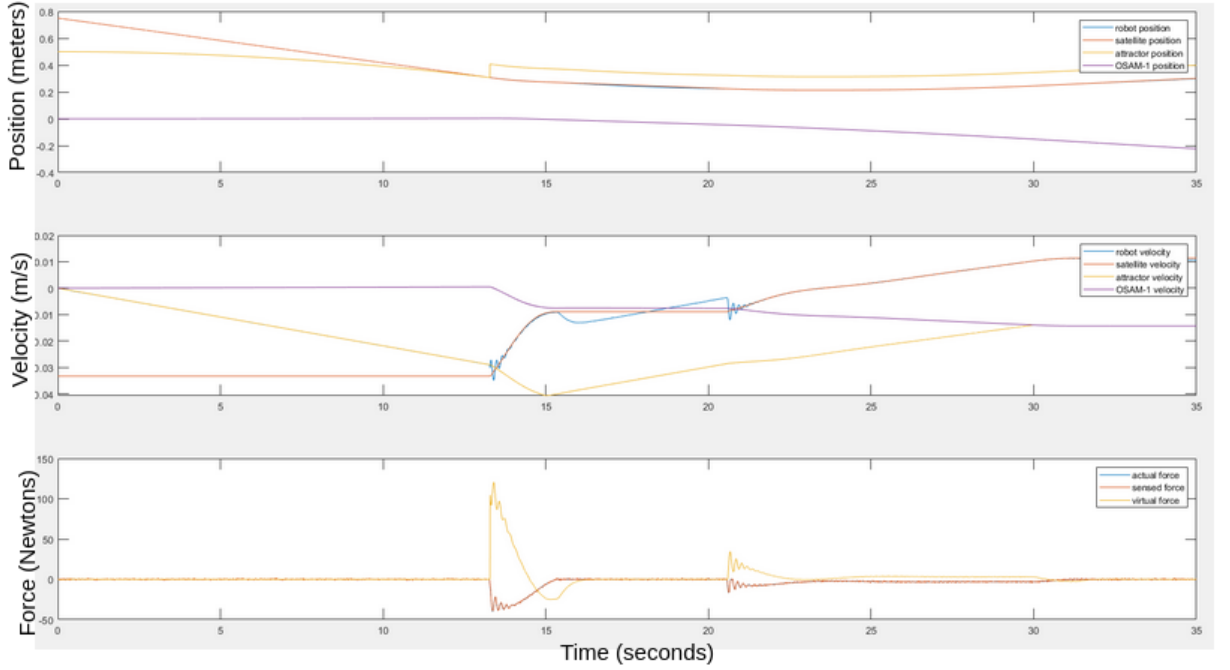


Figure 28: Simulated 1-D capture with a bumpless transition into compliance but with a free robot base. The attractor position is equal to the robot motion plus a constant displacement and the attractor (feedforward) velocity was taken from the planned trajectory.

was made, the robot rapidly accelerated because it was no longer pushing on the massive satellite. Eventually, it caught up to the satellite and made contact again, but this time the velocity difference upon contact was even greater, leading to an even bigger bounce than the previous one. This trend continued indefinitely in the simulation. A real robot, however, would reach the end of its workspace and the satellite would irretrievably bounce away.

4.1.6 Fixed Displacement Attractor

One way to mediate the runaway bouncing effect is to reintroduce damping. In the previous simulation, the damping was effectively negated by the coupling of the attractor velocity to the robot velocity. One option is to return to the original velocity profile from the preplanned stopping trajectory.

Here the triangular velocity profile was kept, but the attractor position was set

to the robot position plus a fixed displacement, in an attempt to maintain constant force. This differs from the previous experiment in that the velocity feedforward was completely decoupled from the actual velocity of the attractor. It was also much more stable than the previous simulation.

4.2 Hardware Experiments

The natural next step was to validate the simulations with physical experiments. The biggest challenge in designing a hardware experiment was to create an analog for the satellite that had substantial inertia but minimal friction.

4.2.1 Wheelie Satellite

One of the most common ways to reduce friction is to use wheels. For initial experiments a small, four-wheeled cart was built using “scooter” wheels. The wheels themselves used ball bearings to minimize friction at the interface between the wheels and the cart. The cart was built without any kind of steering mechanism, so its motion was one-dimensional and linear, as it was in the previously conducted simulations.

As an initial experiment, the cart was set at rest on a flat table surface with a 50lb weight resting on it. The robot started at a fixed distance from the cart, and was controlled to crash into it with a velocity difference of a few centimeters per second at contact. This experiment was done under two different control schemes. The first was pure trajectory control, with no compliance whatsoever. A snapshot from this procedure is shown in Fig. 29 below:

In this experiment, the velocity difference at contact was 2.4 centimeters per second, and the peak contact force was 12 Newtons. Based on the force plot, the satellite *appears* to “bounce” twice before stable contact was achieved. The fidelity of this interpretation is discussed later.

In the second experiment, the robot trajectory was controlled directly until contact

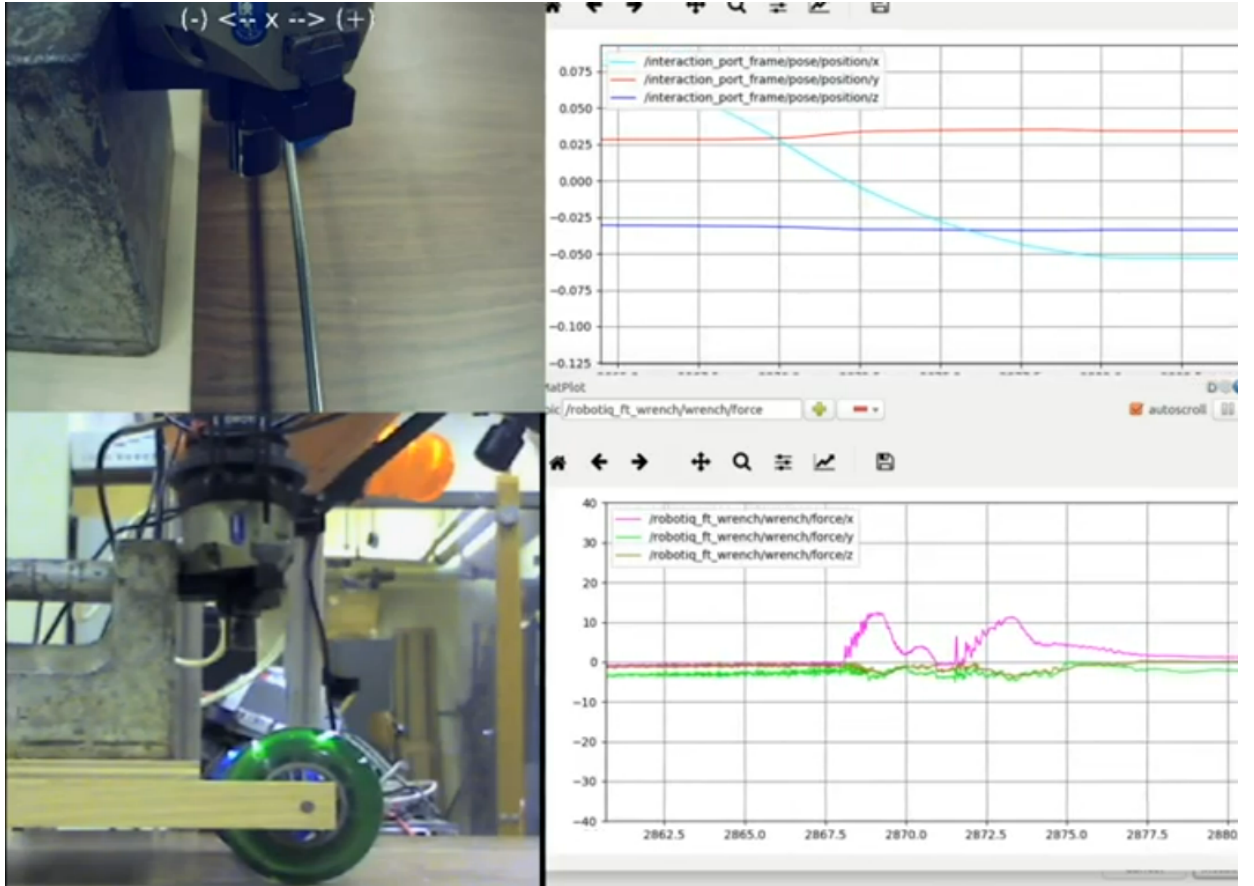


Figure 29: The top left quadrant is a view from the robot's onboard POV camera, while the bottom left quadrant is a view from an external camera. The top right shows a plot of the robot's position as a function of time, where the cyan curve is the relevant axis. The lower plot shows the measured forces as a function of time (The magenta curve is the relevant axis).

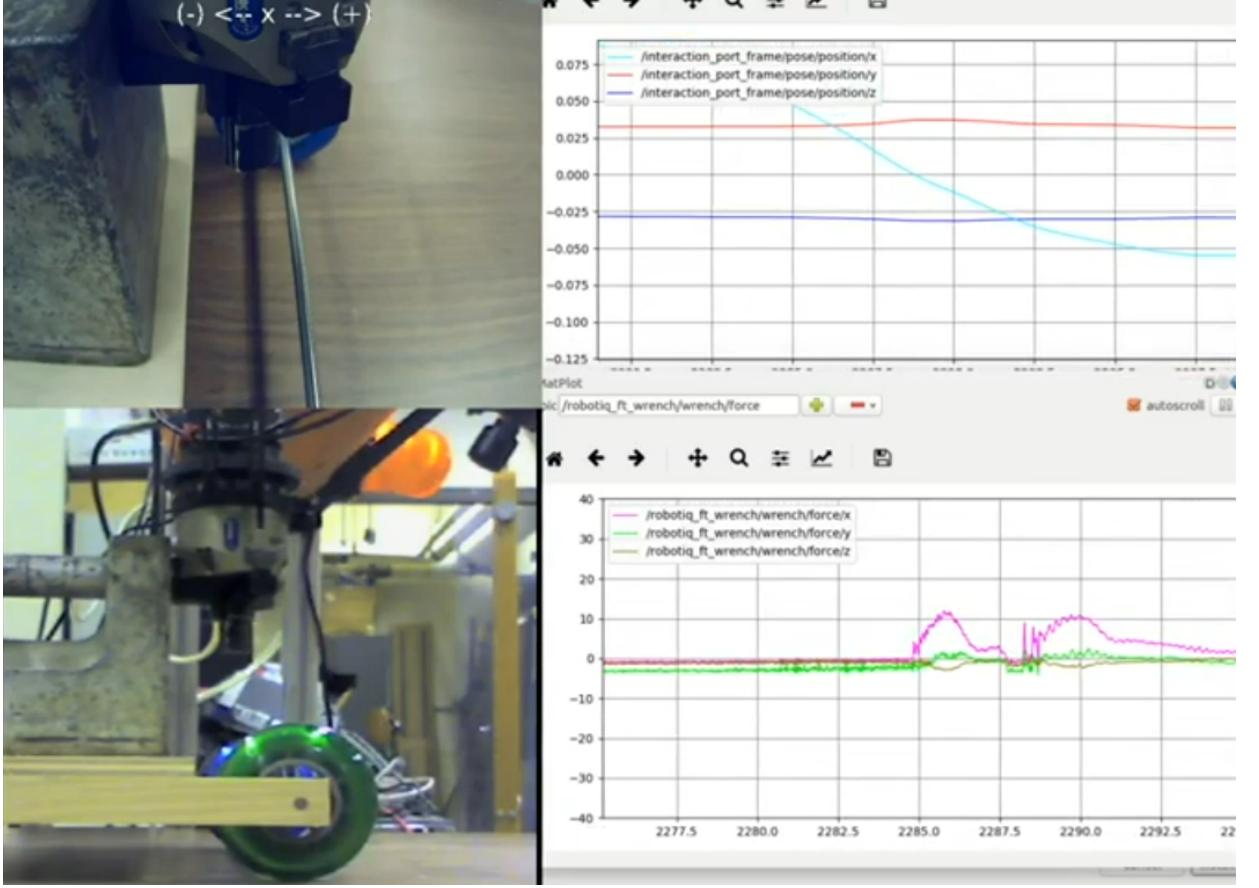


Figure 30: The top left quadrant is a view from the robot’s onboard POV camera, while the bottom left quadrant is a view from an external camera. The top right shows a plot of the robot’s position as a function of time, where the cyan curve is the relevant axis. The lower plot shows the measured forces as a function of time (The magenta curve is the relevant axis).

was made, and after contact, the control was bumplessly switched into compliance.

In this trial, the velocity difference at contact was 2.3 centimeters per second and the peak force measured was 11 Newtons. Once again, the force plot shows two distinct peaks before a steady decline. While the contact force was slightly less in the bumpless trial, it was not a significant enough difference to indicate a clear advantage to this control scheme, especially when paired with the slight difference in contact velocity.

While the plots look similar to those from the simulation, the video [31] reveals that the cart was not behaving like a pure inertia, but exhibited jerky, nonuniform

motion even when it was out of contact with the robot. This was mostly due to the nonuniformity of the scooter wheels used. The wheels were designed for use on concrete and asphalt pavement, so perfect uniformity likely was not a design consideration. Moreover, the rubber used to make the wheels, while excellent for traction, deforms significantly under large loads, resulting in undesirable rolling resistance.

These issues could have been mitigated by using more rigid wheels and flat rails instead of a table surface. Unfortunately, rolling resistance is generally proportional to the normal load. This means that any wheeled system will have a tradeoff between inertia and mobility.

4.3 Air bearings

One way to remove the weight dependence of friction is to place a lubricant between all sliding interfaces. Lubricant friction depends on speed, separation, and viscosity. The interface separation itself can depend on speed and viscosity, and the speed is determined by how fast the satellite needs to move in the tests. Therefore the viscosity is the only real design choice for lubrication.

Air itself is one of the least viscous lubricants available. Air bearings are used commonly for terrestrial satellite testing, including docking [32] and reaction control systems [33]. Unfortunately, commercial air bearing systems are prohibitively expensive, even for low load models. As it turns out, however, air bearings are surprisingly easy to construct using relatively cheap materials and processes. Three air bearings were constructed as part of a senior engineering design project with Arlo Petitjean and Tyler Young. The design was inspired by the DiY designs found in [34, 35].

4.3.1 Construction

The air bearings consisted of a 3D-printed plastic shell attached to a graphite puck in a configuration shown in Fig. 31.

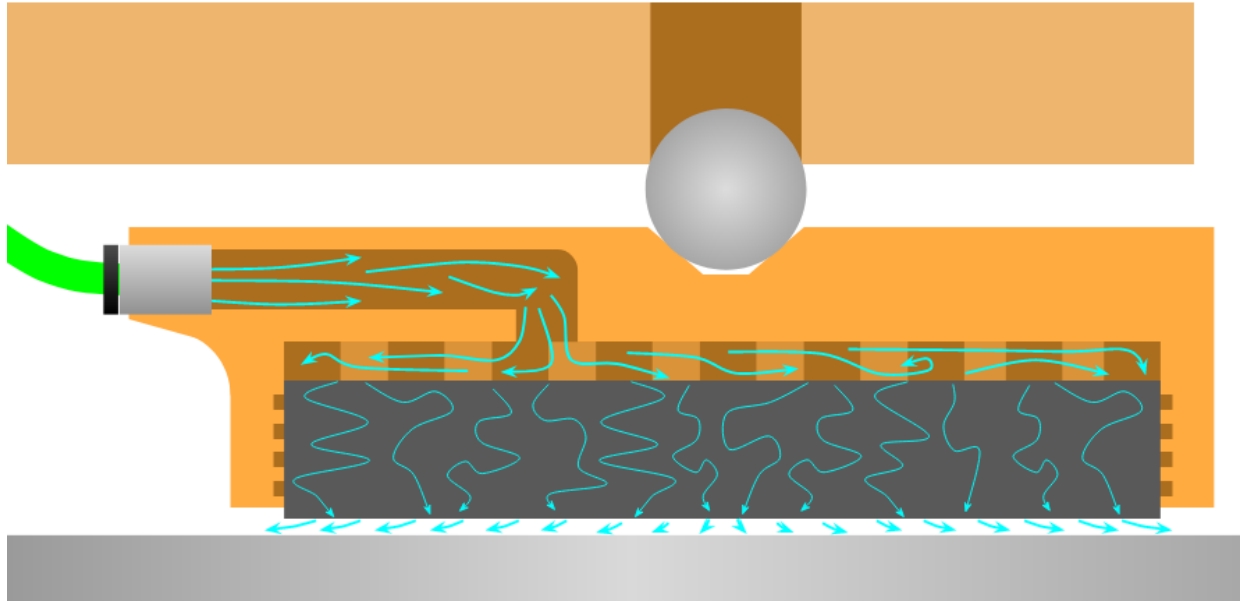


Figure 31: A cross sectional diagram of the air bearing.

The purpose of the 3D printed shell was to apply air pressure to the top of the graphite puck. It consisted of a hollow cylinder with an air inlet on one side. The inlet was tapped to hold a pneumatic push-to-connect fitting and had a pipe that led to the hollowed out portion. The interior face of the shell was covered with evenly spaced offsets. These offsets helped to keep the graphite puck in place. On the interior rim of the puck were several grooves. The grooves' function was to provide a stronger interface between the shell and the epoxy that held the puck in place. The top of the shell had a pyramidal indent designed to interface with a ball “joint.” This allowed the bearing to tilt freely to maintain alignment with the gliding surface.

The puck itself was just a stout cylinder of graphite. It was openly porous, but not isotropically so. This was evident in a bubble test, where the completed bearing is placed under water and pressurized. Bubbles form on the surface of the puck, and their size indicates the local porosity [36]. Isotropic graphite is more desirable and is commercially available (used for EDM forms) but it is significantly more expensive. The anisotropic graphite proved sufficient for our bearing to support the necessary

loads. The graphite was held in place and sealed to the shell using epoxy.

Once constructed, the air bearings could be pressurized. Air from the inlet is funneled over the surface of the graphite. The air works its way through the porous material until it reaches the other side. There it forms a thin pressurized film between the puck and the flat surface it rests on, yielding an almost frictionless interface. Three identical air bearings were placed at the corners of a triangular sled using makeshift ball joints.

4.3.2 Tracking

In order to perform capture experiments, it was necessary to track the position of the satellite. Instead of using an expensive and complicated motion tracking system, a single-camera, visual-inertial tracking system was designed. An overhead webcam was used to track the position of an ArUco [37] marker on the sled. The webcam had a fairly low framerate and shutter speed, so its tracking data, while highly accurate, was intermittent and unreliable.

An inertial measurement unit and microcontroller were placed immediately below the fiducial. The IMU reported the acceleration and angular velocity of the sled to the microcontroller, which sent the telemetry to a host computer over USB. The host computer used an Extended Kalman Filter to fuse the low-frequency, zero-drift camera data with the high-frequency, high-drift IMU data.

In the EKF, the state variable $\mathbf{x}$ consists of the two-dimensional position x and y , one-dimensional orientation ϕ of the sled on the plane, and its 2D translational velocity, v_x and v_y . To initialize the filter, an initial state estimation was created by averaging several pose measurements from the camera tracking system while the sled was at rest. Also with sled at rest, several seconds of IMU readings were averaged to generate a baseline zero for the sensor.

Since the sled itself was unpowered, the motion model consists only of the inertial

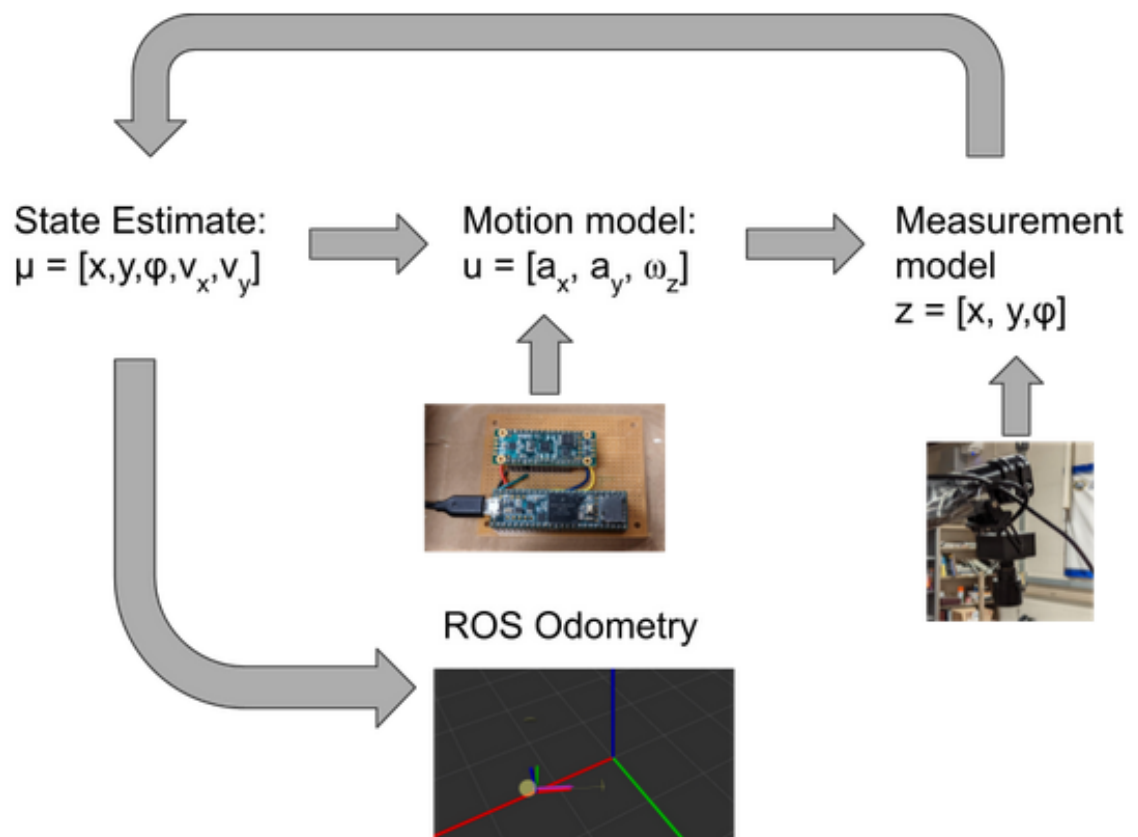


Figure 32: A flowchart representation of the EKF

measurements , a_x , a_y , ω_z . The motion model is defined by the nonlinear map in Eqn. 26.

$$\mathbf{f}(\boldsymbol{\mu}_n, \mathbf{u}_n) = \hat{\boldsymbol{\mu}}_{n+1} \quad (26)$$

This mapping is enumerated in Eqn. 28-Eqn. 32.

$$\hat{x}_{n+1} = x_n + v_{x,n}dt + 0.5dt^2[a_x \cos(\phi_n) - a_y \sin(\phi_n)] \quad (27)$$

$$\hat{y}_{n+1} = y_n + v_{y,n}dt + 0.5dt^2[a_x \sin(\phi_n) + a_y \cos(\phi_n)] \quad (28)$$

$$\hat{\phi}_{n+1}^* = \phi_n + \omega_z dt \quad (29)$$

$$\hat{\phi}_{n+1} = \text{mod}(\hat{\phi}_{n+1}^* + \pi, 2\pi) - \pi \quad (30)$$

$$\hat{v}_{x,n+1} = v_{x,n} + [a_x \cos(\phi_n) - a_y \sin(\phi_n)]dt \quad (31)$$

$$\hat{v}_{y,n+1} = v_{y,n} + [a_x \sin(\phi_n) + a_y \cos(\phi_n)]dt \quad (32)$$

The uncertainty in the IMU data was taken to be a constant 3x3 covariance matrix $\mathbf{M}$. $\mathbf{M}$ is a diagonal matrix, and the first two diagonal entries are the linear acceleration variance, and the third entry is the variance in angular velocity. At every timestep, the uncertainty in the state estimate is updated from $\boldsymbol{\Sigma}$ to $\hat{\boldsymbol{\Sigma}}$ according to Eqn. 33

$$\hat{\boldsymbol{\Sigma}}_{n+1} = \mathbf{G}_n \boldsymbol{\Sigma}_n \mathbf{G}_n^\top + \mathbf{V}_n \mathbf{M} \mathbf{V}_n^\top \quad (33)$$

where $\mathbf{G}$ and $\mathbf{V}$ are Jacobian matrices of $\mathbf{f}$; consisting of the partial derivatives of $\hat{\boldsymbol{\mu}}$ with respect to $\boldsymbol{\mu}$ and $\mathbf{u}_n$, respectively. These derivatives were computed analytically but are not enumerated here.

The measurement $\mathbf{z}$ consists of the ArUco pose measurements, x , y , and ϕ . The measurement update is completely linear, so this part of our EKF is almost identical to a vanilla Kalman filter's measurement update, given by Eqn. 34.

$$\boldsymbol{\mu}_{n+1} = \hat{\boldsymbol{\mu}}_{n+1} + \mathbf{K}_{n+1}(\mathbf{z} - \mathbf{C}\hat{\boldsymbol{\mu}}_{n+1}) \quad (34)$$

$$\boldsymbol{\Sigma}_{n+1} = (\mathbf{I}_3 - \mathbf{K}_{n+1}\mathbf{C})\hat{\boldsymbol{\Sigma}}_{n+1} \quad (35)$$

Where $\mathbf{C}$ is a 3x5 selection matrix defined with a 3x3 identity matrix in the first three columns and zero elsewhere. The purpose of this selection matrix is to apply the motion update to only the pose variables and not the twist ones. $\mathbf{K}$ is the Kalman gain, computed according to Eqn. 36.

$$\mathbf{K}_{n+1} = \hat{\boldsymbol{\Sigma}}_{n+1}\mathbf{C}^\top(\mathbf{C}\hat{\boldsymbol{\Sigma}}_{n+1}\mathbf{C}^\top + \mathbf{Q})^{-1} \quad (36)$$

Where $\mathbf{Q}$ is the 3x3 covariance matrix for the measurement $\mathbf{z}$. Note that if the measurement variance is infinite, then the Kalman gain will be zero, meaning that Eqn. 34 reduces to

$$\boldsymbol{\mu}_{n+1} = \hat{\boldsymbol{\mu}}_{n+1} \quad (37)$$

This was the case between the frames of the webcam when no tracking measurement was made. During these periods, the state was updated through dead-reckoning alone. This EKF was validated in simulation using a predetermined path with normally distributed noise based on sensor properties and performed satisfactorily.

4.3.3 Experiments

A preliminary test was to leave the satellite in a stationary position and actively bring the robot into contact with it. A vertical bracket (a monitor stand) was clamped to the satellite sled to serve as a surface for the robot to push against for capture.

Instead of using a planned trajectory, the attractor was simply placed slightly

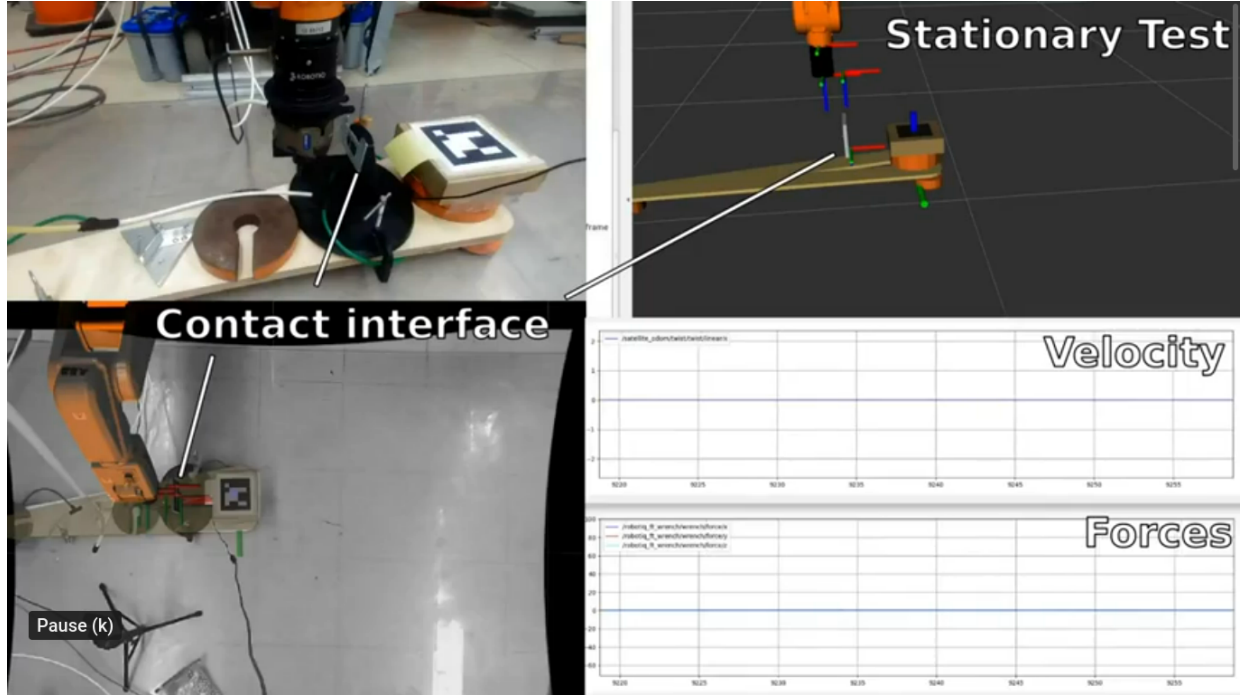


Figure 33: A screenshot of the console during capture testing.

behind the interface bracket until contact is made. Once contact was made, the attractor stayed in the exact same position relative to the bracket as the sled moved. This is similar to the seventh simulation above, shown in Fig. 28. In this test, the robot initially approached the satellite at roughly 2cm/second, made contact with the satellite at close to zero cm/second, and then pushed and maintained contact for several seconds until it left the robot workspace. Based on our criteria this result was a success; no bounce took place and contact was maintained long enough to allow a hypothetical gripper to close.

The next step was to attempt a capture of a moving satellite. For this test, the robot was programmed to hold the attractor in the home position until the satellite came within a specified radius. With the satellite in range, the attractor was placed inside the contact bracket. If the satellite were to leave the radius, the attractor was returned to the home position. This allowed the robot to repeatedly perform capture attempts while the satellite was pushed in and out of the capture radius with various

initial conditions. A video of typical capture experiments can be seen in [38]. Unfortunately, under these test conditions, the satellite bounced away without maintaining contact for any meaningful time period almost every time. This was almost certainly due to a large contact velocity. The reasons for this are discussed below.

5 Discussion and Future Work

The three simple behaviors described herein, PTWL, RWE and dither, were shown to be sufficient for a wide variety of quasi-static tasks, and even some dynamic tasks (e.g. berthing). Compared with more direct implementations of force control or teleoperation, PTWL was observed to be capable of executing a wide variety of manipulation tasks safely and robustly, both under supervisory control and within autonomous state machines. These techniques will allow robots to effectively achieve demanding mission goals for servicing, assembly and manufacturing in harsh environments and subject to communications delays and disruptions.

It is unknown exactly what additional low-level, compliant-motion behaviors might be valuable for either supervisory control or autonomous operation involving manipulation tasks. At present, the three behaviors described here have been found to be remarkably competent and versatile. One potentially useful feature not yet implemented is the option to use a remote center of compliance. Currently, the port of interaction is coincident with the force-torque sensor and cannot be moved by the user. As mentioned above, moving the port of interaction forward can aid in insertion tasks. One of the main difficulties in implementing this is to guarantee passivity.

Other future work could also improve our experimental setup. One of the biggest weaknesses in this setup was the chassis of the robot used for the experiments. The robot was mounted on a mobile base taken from a mobility scooter. The wheels were braced to keep the robot base stationary, but the pneumatic rubber tires themselves are naturally compliant. This has the effect that the entire robot base can comply even when the robot is controlled stiffly. When the robot held stiffly in its home position, the end effector can be moved by around a centimeter with a horizontal force of roughly 20 Newtons. The entire system has a natural frequency around one Hz.

Despite these apparent drawbacks, the experiments conducted above are still useful. For one, the virtual stiffnesses used in all of these experiments were much smaller than the chassis stiffness, so compliance was mostly due to the accommodation controller. Moreover, the robotic arm used for a hypothetical satellite servicing mission will have substantial physical compliance due flexibility in the lightweight arm as well as mobility of the servicing vehicle itself. Regardless, higher-fidelity future experiments should strive to emulate the physical compliance of the robot and vehicle more accurately.

A more serious issue with the experimental setup was with the air-bearing rig. The 3D-printed air bearings and sled have satisfactory performance. The difficulty was in finding a large, flat, level and smooth surface to support the bearings. Smoothness is important so that aberrations do not cut through the thin pressurized film of air and rub against the bearing itself. Flatness and level are important so that the satellite does not experience any gravitational forces that would cause drift. And of course, the surface needs to be large enough to accommodate the range of motion necessary to adequately simulate capture, determined to be around one meter by two meters.

Precision granite slabs, commonly used in manufacturing, would make an ideal surface for the air bearings. Unfortunately they are prohibitively expensive. Other air-bearing satellite testing facilities frequently use epoxy based floors. These floors can be cast in arbitrary shapes and sizes, but the resin is also a substantial cost. The casting process should be done in the absence of particulates that would compromise the surface quality, but it also must be done in-situ. For these reasons, a 2mx6mx2cm polycarbonate sheet that was already present in the lab was used. The sheet was sanded and buffed to ensure adequate smoothness. It was also leveled with cardboard shims at the edges. Unfortunately, despite the sheet's thickness, it was still flexible, and shimming could not completely eliminate warping. This meant that in certain areas of the floor, the whole sled experienced a non-negligible acceleration in various

directions. In areas with especially high curvature, the bearing itself even rubbed against the surface.

These problems were not especially noticeable during the berthing experiments where the robot was constantly holding onto the sled. They did, however, make the capture experiments difficult. For the experiments starting with a stationary satellite, the ever present slopes made it difficult (though not impossible) to place the satellite such that it would not slide out of position. Also, once the satellite is moving and in contact with the robot, it may experience unpredictable forces or accelerations due to this warping.

For the experiment where the robot intercepts a moving satellite, the problem was much worse. It is almost impossible to make the satellite coast toward the robot at a constant velocity; it always accelerates or deflects significantly over the course of its travel. Even in the best case, this makes it impossible to use a preplanned trajectory to intercept the satellite. Instead, the robot must constantly rely on tracking data and either a constantly updating trajectory planner or a simple feedback controller to intercept the satellite. In the worst case, the satellite drifts so far off course that it is entirely impossible to intercept. In order to avoid this scenario, the satellite's initial velocity needs to be much greater in scale compared to the drift acceleration. Unfortunately, higher initial velocities result in a higher contact velocity, which is unrealistic and makes bounceless capture much more difficult. Future research using air bearings should probably opt for a more expensive epoxy floor to avoid these issues.

References

- [1] DJI. *DJI Digital FPV System*. 2022. URL: <https://www.dji.com/fpv>.
- [2] Jan-Philipp Stauffert, Florian Niebling, and Marc Erich Latoschik. “Simultaneous Run-Time Measurement of Motion-to-Photon Latency and Latency Jitter”. In: *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 2020, pp. 636–644. DOI: [10.1109/VR46266.2020.00086](https://doi.org/10.1109/VR46266.2020.00086).
- [3] Nancy J. Currie and Brian Peacock. “International Space Station Robotic Systems Operations - a Human Factors Perspective”. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 46.1 (2002), pp. 26–30. DOI: [10.1177/154193120204600106](https://doi.org/10.1177/154193120204600106). eprint: <https://doi.org/10.1177/154193120204600106>. URL: <https://doi.org/10.1177/154193120204600106>.
- [4] William R. Ferrell. “Remote manipulation with transmission delay”. In: *IEEE Transactions on Human Factors in Electronics* HFE-6.1 (1965), pp. 24–32. DOI: [10.1109/THFE.1965.6591253](https://doi.org/10.1109/THFE.1965.6591253).
- [5] Charles R. Price, Reginald Berka, and John T. Chladek. “Dexterous Orbital Servicing System (DOSS)”. In: 1994.
- [6] Angel Flores-Abad et al. “A review of space robotics technologies for on-orbit servicing”. In: *Progress in Aerospace Sciences* 68 (2014), pp. 1–26.
- [7] William R. Ferrell and Thomas B. Sheridan. “Supervisory control of remote manipulation”. In: *IEEE Spectrum* 4.10 (Oct. 1967), pp. 81–88. ISSN: 1939-9340. DOI: [10.1109/MSPEC.1967.5217126](https://doi.org/10.1109/MSPEC.1967.5217126).
- [8] Wyatt Newman et al. “Autonomous valve turning with an Atlas humanoid robot”. In: *2014 IEEE-RAS International Conference on Humanoid Robots*. 2014, pp. 748–748. DOI: [10.1109/HUMANOIDS.2014.7041446](https://doi.org/10.1109/HUMANOIDS.2014.7041446).

- [9] Zheng-Hao Chong et al. “Autonomous wall cutting with an Atlas humanoid robot”. In: *2015 IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*. 2015, pp. 1–6. DOI: [10.1109/TePRA.2015.7219673](https://doi.org/10.1109/TePRA.2015.7219673).
- [10] Tamar Flash and Neville Hogan. “The coordination of arm movements: an experimentally confirmed mathematical model”. In: *Journal of neuroscience* 5.7 (1985), pp. 1688–1703.
- [11] David A. Rosenbaum. “Chapter 1 - Introduction”. In: *Human Motor Control (Second Edition)*. Ed. by David A. Rosenbaum. Second Edition. San Diego: Academic Press, 2010, pp. xvii–9. ISBN: 978-0-12-374226-1. DOI: <https://doi.org/10.1016/B978-0-12-374226-1.00001-2>. URL: <https://www.sciencedirect.com/science/article/pii/B9780123742261000012>.
- [12] Roger M Enoka and Jacques Duchateau. “Rate Coding and the Control of Muscle Force”. In: *Cold Spring Harbor Perspectives in Medicine* (2017).
- [13] Arvid QL Keemink, Herman van der Kooij, and Arno HA Stienen. “Admittance control for physical human–robot interaction”. In: *The International Journal of Robotics Research* 37.11 (2018), pp. 1421–1444. DOI: [10.1177/0278364918768950](https://doi.org/10.1177/0278364918768950). eprint: <https://doi.org/10.1177/0278364918768950>. URL: <https://doi.org/10.1177/0278364918768950>.
- [14] M. Dohring and W. Newman. “The passivity of natural admittance control implementations”. In: *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*. Vol. 3. 2003, 3710–3715 vol.3. DOI: [10.1109/ROBOT.2003.1242166](https://doi.org/10.1109/ROBOT.2003.1242166).
- [15] Michael J. Ryan. “Implementation of robotic force control with position accommodation”. In: 1992.

- [16] W. Newman, A. Covitch, and R. May. “A client/server approach to open-architecture, behavior-based robot programming”. In: *2nd IEEE International Conference on Space Mission Challenges for Information Technology (SMC-IT’06)*. 2006, 8 pp.–496. DOI: [10.1109/SMC-IT.2006.6](https://doi.org/10.1109/SMC-IT.2006.6).
- [17] J A Maples and Joseph J. Becker. “Experiments in force control of robotic manipulators”. In: *Proceedings. 1986 IEEE International Conference on Robotics and Automation 3* (1986), pp. 695–702.
- [18] Surag Balajepalli. “Modeling, Analysis, And Experiments On A Robot Arm With Force-Feedback Interaction Control”. MA thesis. Cleveland, OH: Case Western Reserve University, May 2020.
- [19] Matthew Habermusch. “Autonomous Skills for Remote Robotic Assembly”. MA thesis. Cleveland, OH: Case Western Reserve University, May 2020.
- [20] Matthew Strube et al. “Ground Simulation of an Autonomous Rendezvous and Tracking System using Dual Robotic Systems”. In: Sept. 2012.
- [21] Shuang Wu, Fangli Mou, and Ou Ma. “Contact Dynamics and Control of a Space Manipulator Capturing a Rotating Object”. In: *AIAA Guidance, Navigation, and Control Conference*. 2017. DOI: [10.2514/6.2017-1048](https://doi.org/10.2514/6.2017-1048). eprint: <https://arc.aiaa.org/doi/pdf/10.2514/6.2017-1048>. URL: <https://arc.aiaa.org/doi/abs/10.2514/6.2017-1048>.
- [22] Craig Carignan, Nicolas Scott, and Stephen Roderick. “Hardware-in-the-Loop Simulation of Satellite Capture on a Ground-Based Robotic Testbed”. In: *International Symposium on Artificial Intelligence, Robotics and Automation in Space 2014*. June 2014.
- [23] Joseph Cressman. *Compliant Berthing*. Youtube. 2022. URL: <https://youtu.be/5Y10A4D1sug>.

- [24] Joseph Cressman. *State Machine Berthing*. Youtube. 2022. URL: <https://youtu.be/QJ6CwK0a3s0>.
- [25] Rahul Pokharna. “Compliant Behaviors for Remote Robotic Operations”. MA thesis. Cleveland, OH: Case Western Reserve University, Jan. 2022.
- [26] Joseph Cressman. *Unlatch, open and close door - first attempt: relaxed translation, manual euler angles*. Youtube. 2021. URL: <https://youtu.be/ViTUKE28ViM>.
- [27] Joseph Cressman. *Saucepan lid installation state machine*. Youtube. 2021. URL: <https://youtu.be/N0j9ks2Byk>.
- [28] Joseph Cressman. *Socket Wrench Insertion*. Youtube. 2022. URL: https://youtu.be/Zv6_qFDisVY.
- [29] Justin McFatter, Karl Keiser, and Timothy Rupp. “NASA Docking System Block 1: NASA’s New Direct Electric Docking System Supporting ISS and Future Human Space Exploration”. In: 2018.
- [30] Evangelos Papadopoulos et al. “Robotic Manipulation and Capture in Space: A Survey”. In: *Frontiers in Robotics and AI* 8 (2021). ISSN: 2296-9144. DOI: [10.3389/frobt.2021.686723](https://www.frontiersin.org/articles/10.3389/frobt.2021.686723). URL: <https://www.frontiersin.org/articles/10.3389/frobt.2021.686723>.
- [31] Joseph Cressman. *bumpless in motion edit 1*. Youtube. 2021. URL: <https://youtu.be/j1tad1EwxFk>.
- [32] Christopher Jewison et al. “Definition and testing of an architectural tradespace for on-orbit assemblers and servicers”. In: *65th International Astronautical Congress*. Oct. 2014.
- [33] Hendrik Kolvenbach and Miro Voellmy. “Design of a Low-Cost, Spherical Air-Bearing Satellite Simulator”. In: *15th Symposium on Advanced Space Technologies in Robotics and Automation*. June 2019.

- [34] David Preiss. *DIY Air Bearings*. Youtube. 2019. URL: https://youtu.be/vuzoEe_ykmM.
- [35] Ben Krasnow. *Make vacuum pre-loaded spherical air bearings without special tools*. Youtube. 2019. URL: https://youtu.be/K_N_h_mKf-4.
- [36] Joseph Cressman. *Air Bearing Porous Graphite in Water*. Youtube. 2022. URL: <https://youtu.be/JfjoZXQDvdI>.
- [37] S. Garrido-Jurado et al. “Generation of fiducial marker dictionaries using Mixed Integer Linear Programming”. In: *Pattern Recognition* 51 (2016), pp. 481–491. ISSN: 0031-3203. DOI: <https://doi.org/10.1016/j.patcog.2015.09.023>. URL: <https://www.sciencedirect.com/science/article/pii/S0031320315003544>.
- [38] Joseph Cressman. *Compliant Robotic Planar Satellite Capture Demo using Air Bearings*. Youtube. 2022. URL: <https://youtu.be/BRKxIuH6v1E>.