The Thesis Committee for Clinton Dean Peterson II
Certifies that this is the approved version of the following thesis:

Industrial Automation and Control in Hazardous Nuclear Environments

APPROVED BY

SUPERVISING COMMITTEE:

Sheldon Landsberger, Supervisor

Mitchell Pryor, Co-Supervisor

Industrial Automation and Control in Hazardous Nuclear Environments

by

Clinton Dean Peterson II, B.S.M.E.

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN ENGINEERING

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2015

Dedicated to my parents.

Acknowledgments

I wish to thank my parents, friends, fellow lab rats, and my advisors Mitchell Pryor and Sheldon Landsberger for their support, encouragement, and advice.

The work done in this document was sponsored by the Nuclear Regulatory Commission Graduate Fellowship and Los Alamos National Lab.

Industrial Automation and Control in Hazardous Nuclear Environments

Clinton Dean Peterson II, M.S.E.
The University of Texas at Austin, 2015

Supervisors: Sheldon Landsberger
Mitchell Pryor

This report discusses the design and implementation of an automated system for use in geometrically-constrained, hazardous glovebox environments. This system's purpose is to reduce a hemispherical plutonium pit into smaller pieces that fit inside of a crucible. The size reduction of plutonium pits supports stockpile stewardship efforts by the United States Department of Energy. The automation of this process increases the safety of radiation workers by handling radioactive nuclear material. This decreases glovebox worker dose and exposure to tools, sharps, and fines.

This effort examines the hardware and software framework developed to support the use of a Port Deployed Manipulator (PDM) for a contact task. This research effort uses a 7 Degree-of-Freedom (DOF) PDM and a micropunch to reduce hemispherical pit surrogates. Formulation of the material reduction execution algorithm involved addressing a variety of topics related to industrial automation:

- Collision detection and object recognition based on user-specified parameters.
- Joint torque monitoring
- Online motion planning for contact tasks
- *Object-in-hand* industrial manufacturing
- Grasping and handling of nuclear material
- Software compliance via robust nonlinear control methods

A high-bandwidth collision detection algorithm involving joint torque monitoring was developed to increase robot safety during operation. The motion planning algorithm developed for this effort takes variable geometric properties to be used with a range of hemishells. The algorithm's feasibility was validated on a hardware test bed in a laboratory setting.

Hardware cold tests conclude that mechanical compliance is sufficient for task completion. However, software compliance would increase performance, efficiency, and safety during task execution. Two different nonlinear force control laws (feedback linearization and sliding mode control) that minimize object shear forces were developed using a simplified material reduction simulation. It is recommended that glovebox automation research continue to increase worker safety throughout the DOE complex.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Mankind has long dreamed of autonomous beings working alongside people. Science fiction writers such as Asimov have depicted robots as benefactors, with circuits programming the robots to be benevolent. This dream would eventually become reality in 1959, when the first prototype industrial robot to be used on the factory floor was created, called the Unimate [41]. It was created by George Devol and Joseph Engelberger, who would later found Unimation Inc. The Unimate was a five *degree-of-freedom (DOF)* automated serial manipulator intended for use on the General Motors factory floor as a die casting machine. This machine intended to replace workers on the factory floor ***for a task that was undesirable and dangerous***.

Figure 1.1: The Unimate robot demonstrating how to golf on the Johnny Carson Show (1966) [1]

In 1966, the Air Force-sponsored study known as Project Hindsight examined several research projects conducted after World War II. The initial report concluded:

> *"A utilized innovation can occur only when there is a conjunction of three elements: (1) a recognized need; (2) competent people with relevant scientific or technological ideas; and (3) financial support."* [50]

This section of the report focuses on the first element: a recognized need. The desire for robots to replace human workers in industry is still present today, with the need for robots expanding past its roots in the automotive industry. Industrial robots are utilized for increasingly complex tasks. The da Vinci Surgical System is a robot operating in the medical field [29]. Robots are commonly used on the factory floor of many companies for manufacturing with welding and grinding robots taking the place of human workers.

Robots are finding more and more use in the nuclear industry as well. The need for improved safety and performance in power plants has led to the introduction of robotic systems working alongside humans [21]. The United States Department of Energy (DOE) is now employing automation for tasks conducted at national labs [55]. The DOE recognizes the need for robots to be employed in national labs. Efforts have been made to automate tasks typically performed by humans to primarily to reduce dose and exposure to hazardous and potentially radioactive nuclear material. This effort will focus on automating one such handling task.

1.1 Nuclear Material Handling

It is necessary to understand the environment of workers in the nuclear industry that directly handle nuclear material. At Los Alamos National Laboratory (LANL), radiation workers in the Plutonium Facility (TA-55) typically handle objects and materials containing a substantial amount of alpha-emitting plutonium isotopes. [1] In order to contain radiological contamination and prevent the spread of toxic material, all workers must perform nuclear handling tasks from outside of a sealed environment. These tasks are typically executed in a lead-lined glovebox. A glovebox is a containment vessel that acts as a physical barrier between the operator and the material inside. Most gloveboxes are made of stainless steel, with lead lining layered inside. The glovebox features window ports for viewing and glove ports to allow for interaction and handling.

There are a number of hazards imposed on DOE glovebox workers by operating in this environment. Structurally, the weakest part of a glovebox is the gloveport itself. The gloves lining the gloveport are much more susceptible to mechanical, chemical, and radiological failure than any other glovebox component [39]. This puts the radiation worker at risk of glove breach during normal operation. Certain safeguards exist to minimize the harmful effects of a glove breach, such as lowering the pressure inside of gloveboxes relative to the exterior. This prevents contaminants from escaping the glovebox. However, a serious breach puts the operator at risk of exposure to unwanted dose. Some machining processes that commonly take place inside of gloveboxes involve the creation of small, jagged pieces of material (known as sharps and/or fines). These sharps and

[1] Alpha particles consist of two protons and two neutrons. Alpha radiation is easily attenuated by even thin material, but is highly damaging if consumed.

fines greatly increase the risk of a glove breach.

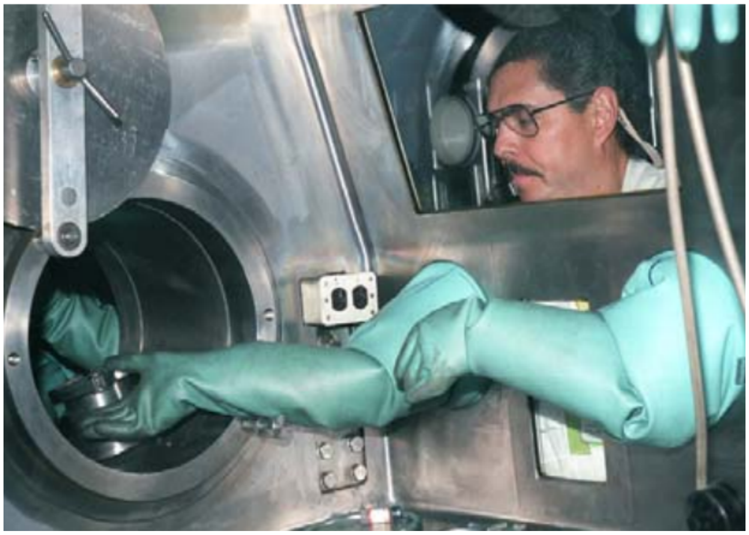


Figure 1.2: Typical glovebox worker environment[32]

An injury from a glove breach or failure would be the most severe glovebox event that could happen to a glovebox worker (excluding criticality events). However, the Glovebox Glove Integrity Program (GGIP) at LANL has rated ergonomic injury as the greatest glovebox environment failure risk to a glovebox worker [39]. The risk evaluation of environment failure is based on not only the severity of the risk, but also on ease of detection and occurrence rate. Ergonomic injuries are particularly common for glovebox workers due to the cramped workspace, varying height of glovebox workers in relation to a fixed gloveport height, and stress. Lost work days and productivity due to ergonomic injuries indirectly cost employers like LANL as well. In 2003, strains and sprains due to injury on the job were the leading causes of missing work days by a wide margin for the United States [32]. The motivations to automate these tasks includes safety, dose, and containment.

1.2 Task Statement and Objectives

This thesis follows the design and implementation of an autonomous industrial robotic manipulator system for the reduction of hazardous materials. The system was developed to be used in gloveboxes commonly found in the DOE complex. Traditionally, a human glovebox operator would handle hazardous radioactive materials via the glove ports. Different glovebox applications each require a specific procedure to be carried out by the glovebox operator. The goal of this system is to reduce risks to DOE workers by removing the need for a human operator from one particular glovebox procedure that involves the size reduction of a hemispherical plutonium shell.

One possibility for autonomous system deployment in a glovebox is the use of a Port Deployed Manipulator (PDM). For an autonomous system to be deployed in a hazardous contained environment, many of the challenges involved in safely deploying a PDM should be addressed. An ideal PDM deployment in a glove port would allow the robot to be safe from radiological and physical damage while maintaining mobility and function. The requirements and solutions for safely deploying a PDM has been previously researched and covered by Joseph Hashem of The University of Texas at Austin's Nuclear Robotics Group (UT NRG) [19]. Hashem's thesis primarily concerns the requirements for deployment and preparation of a flexible robotic system for operation in a glovebox. Based on this research, the work done in this document is primarily focused on the use of a flexible PDM for a specific contact task.

1.2.1 Background

An essential part of an implosion-based nuclear bomb is the plutonium pit. The pit is composed of two hemispherical shells. The United States created many thousand nuclear weapons and pits, competing primarily with the Soviet Union in the nuclear arms race during the Cold War. Both the United States and Russia have taken an interest in the reduction of their nuclear arsenals. Treaties such as the Strategic Arms Reduction were signed with the intent that a number of nuclear weapons should be retired by a certain date. Many thousands of nuclear warheads have been dismantled or retired since the signing of such treaties. The nuclear weapons are transferred from the Department of Defense (DOD) to the Department of Energy (DOE) for staging and dismantling. Historically, the stockpile for plutonium pits was the Rocky Flats site in Colorado. Since its closure, weapons have been taken to the Pantex Plant in Amarillo, Texas, for dismantlement. After dismantlement of the warhead, many pits are staged for interim storage at Pantex itself. Secondary components are shipped to various facilities in the nuclear complex, such as the Y-12 plant in Oak Ridge, Tennessee, or the Savannah River Site in South Carolina. [6]

In addition to policy concerns, plutonium pits have a limited shelf life. The plutonium metal corrodes and oxidizes easily. The particles formed during storage pose a radiological hazard as they could easily be inhaled by a human. Fissile Pu-239 is an isotope of particular interest in plutonium weapons pits. Pu-239 decays by alpha-particle emission to U-235. Alpha particles pose a serious health risk when ingested. Eventually, Am-241 is produced via a decay chain. Am-241 poses a radiological threat by emitting high-energy gammas through decay.

The DOE is interested in continuing the reduction of the nuclear arsenal

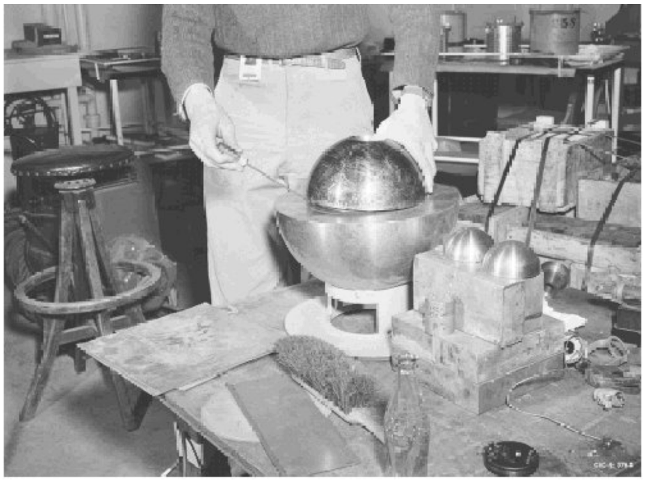


Figure 1.3: Mock plutonium pit; recreation of the 1945 Louis Slotin criticality accident at Los Alamos. [37]

through pit dismantlement and reduction. One option is the conversion of weapons pits to plutonium oxide. This oxide could then be processed and fabricated into Mixed-Oxide Fuel (MOX) used as fuel for power reactors. Another option is the reprocessing of the plutonium in pits to be refabricated, essentially extending the shelf life of existing pits. However, before the DOE can make any use of the weapons pits, the pits themselves need to be rendered usable again via a re-smelting process. This process starts with breaking down the pits into smaller pieces that can fit into an induction crucible. Typically a human radiation worker completes this task using off-the-shelf power tools in gloveboxes to reduce the pit into smaller pieces. The automation of this task is of particular interest to LANL in order to maintain stockpile stewardship.

1.2.2 Task-Defined and System-Defined Performance Constraints

This section outlines the task-defined constraints the autonomous system must follow during operation.

- *The plutonium hemishell should be size-reduced into components that fit within a standard cylindrical crucible (Approximately 8" deep by 3" in diameter)*

Figure 1.4: Molten plutonium in a LANL crucible [48]

The induction crucibles used to melt plutonium components have a standard set geometry. Material entering the crucible must conform to size standards to ensure they fit in the crucible and settle with a low packing factor.[2]

- *All plutonium material should be accounted for within 0.5 grams*

Due to the nature of nuclear nonproliferation policy in the United States, it is necessary to have accountability of material flow throughout the manufacturing processes. For reference, 0.5 grams of plutonium is roughly equivalent to 0.025 cm^3. This is about the size of a grain of rice.

- *Process should adhere to ALARA principles*

[2] A packing factor refers to how well material can fit into a set volume.

Every process involving nuclear material handling must make a reasonable effort to adhere to minimize exposure and dose to radiation workers (ie. ALARA [3]) as defined by NRC regulations 10 CFR Part 20 [53].

- *The entire autonomous system should operate within the geometric constraints of the glovebox environment*

All nuclear material for this task must be contained within the glovebox. Therefore the system should operate within the workspace of the glovebox. This does not necessarily mean that the robot must fit inside of the glovebox entirely. The robot chosen could be a PDM.

- *The system should be able to withstand alpha-emitting radiation from common plutonium isotopes used within glovebox applications*

The robot should be rated for a suitable amount of operating time within a hot glovebox environment. If the robot is a PDM, suitable shielding is required for the robot to operate within the glovebox. The robot would therefore be replaced after its rated operating time to account for robot components degrading due to radiation damage. In addition, LANL complies with a multitude of additional regulation requirements to assume systems operate safely and securely. It is beyond the scope of this effort to address all these constraints. However, UT researchers will regularly communicate with LANL engineers to ensure there are no obvious conflicts with the proposed solution.

The following are system-defined constraints and desires that further define the autonomous system, along with an explanation for each:

[3] As Low As Reasonably Achievable

- *The procedure should be autonomous, with minimal human interaction necessary for completion*

The autonomous system described in this report will most likely differ from the final design iteration to be implemented in an actual DOE hot glovebox. Therefore, the goal of this thesis is to demonstrate that this material reduction task can be fully automated in a feasible manner. However, the general reduction algorithm should fully complete the material reduction task without human intervention regardless of hardware platform.

- *Human interaction with the system should be minimal, clearly defined, and user-friendly*

The end-user of this autonomous system is a DOE glovebox worker or technician. The user should ideally only be required to set up, start, and emergency stop the system. Troubleshooting the system should be intuitive and require minimal hardware or software interaction. It is not necessary to have a user-friendly interface for the purposes of this report, but is necessary for the final software implementation.

- *The robot system must support a range of plutonium pit geometries compatible with current and foreseeable demand*

It is desired to complete this task for multiple types and classifications of plutonium pits. This report assumes a uniform hemispherical pit geometry with variable radius.

- *The motion-planner and inverse kinematic solver should be robust*

The software framework should have online motion planning built in. It is desirable for the end-user and DOE engineering support to have access to a high-level API as well as the low-level inverse kinematics and planning framework. The operator must have control via clear and intuitive interface modes (EEF jogging, joint control, and target pose goal setting, etc).

- *The supervisor must be able to stop and control the robot at any time*

Absolute safety cannot be guaranteed so a user must always be able to intervene in the case of an emergency.

- *The system must be theoretically tested within an appropriate cold-testing environment with geometrically-similar plutonium surrogates*

It is necessary to test operation of the reduction algorithm on a robot system within a controlled cold glovebox before implementing the same algorithm in a glovebox with radioactive material inside. Therefore, a suitable surrogate object should be used for testing purposes. It is desirable for this surrogate to be similar in size and shape to an actual plutonium hemishell. It is also desireable for the surrogate to have certain similar material properties to a plutonium hemishell in order to accurately simulate object behavior during material reduction.

- *The system should reliably perform collision checking and obstacle avoidance during operation*

Collision detection is a robot application that necessitates software support for point-to-point motion planning and reactive motion planning. It is required for the robot system to check for collisions during motion planning to avoid obstacles. It is also required that the robot react to disturbance forces and torques that might otherwise damage the robot, cause a hardware malfunction, lower the performance of the task, or cause a criticality event. For point-to-point motion planning, it is necessary to have *a priori* knowledge of the workspace of the robot. For glovebox applications, the workspace is usually fully defined and all geometric obstacles known prior to starting the automation system. These obstacles should therefore be accountable in software. Collision checking is explained in detail in Chapters 2 and 4.

- *The robot should exhibit partial compliance in material handling when interacting with the material reduction tool*

The material reduction tool will introduce disturbance forces and torques on the end-effector (EEF) of the robot in addition to interacting with the object being manipulated. Therefore, it may be necessary for the manipulator to exhibit compliance during the material reduction task. Compliance can come in the form of reactive motion planning based on sensor inputs, or mechanical compliance in the robot or manipulated component exhibiting finite stiffness.

- *If software compliance is used, the control law for compliance should be robust and stable*

Contact control tasks involving a robotic manipulator are inherently nonlinear due to robot geometry. Software compliance should utilize a suitable nonlinear control law for bounded disturbance forces and torques.

1.3 Industrial Automation

The broadening market for industrial robots was mentioned previously. The International Organization for Standardization (ISO) gives an exact definition of an industrial robot:

> automatically controlled, reprogrammable, multipurpose manipulator, programmable in three or more axes, which can be either fixed in place or mobile for use in industrial automation applications[...] The industrial robot includes: the manipulator, including actuators; the controller, including teach pendant and any communication interface (hardware and software). [2]

Industrial robots are the standard in industry for many subtractive manufacturing tasks. By using robots instead of people, much of the safety concerns for factory workers have vanished. Productivity, process control, and quality control are various aspects of manufacturing that industrial robots have a distinct advantage over humans. The advantage of using an industrial robot for a material reduction task in this instance would be the highly configurable nature of a programmable manipulator in addition to dose reduction. Industrial robots in recent years have been shown to handle more and more tasks as opposed to specializing in one particular task. In Figure 1.5, robots on the Tesla factory floor perform many different tasks such as welding, riveting, bonding, and assembly.

1.3.1 Compliance in Robots

An important topic in industrial robotics is the study of compliance. Compliance refers to either a mechanical or software-driven feedback mechanism to

Figure 1.5: Industrial robots at a Tesla manufacturing plant in Fremont, California [36]

respond to environmental forces in a flexible manner. In many industrial automated tasks, compliance is necessary. Compliance allows for improved safety and motion plan feedback. Compliance could come in the form of passive mechanical joint flexibility, or non-rigid links and EEFs. Often, however, compliance is software-driven in the form of a control law taking force and torque sensor feedback and outputting a desired robot trajectory. Many industrial systems use software compliance for contact tasks such as grinding. Software compliance for industrial robotic systems has been investigated at the Nuclear Robotics Group [56] [28]. In this particular material reduction contact task, reactive force control might be necessary to reduce forces on both the pit and the robot. High cutting forces on a moment arm in relation to the EEF could potentially put unwanted torques on joint servos. Sufficiently high shear forces on the pit lend to unwanted warping of part geometry.

1.4 Summary of Objectives

The primary objective of this thesis is to develop a procedure for a flexible automation system to complete an industrial contact task that might be performed in a geometrically constrained, radiologically hazardous environment. Additionally, further research is done to analyze the nonlinear force control of a simplified, generic serial manipulator for this specific task via mathematical simulation. A software package is developed to simulate the nonlinear control of this specific task for a planar manipulator.

1.4.1 Original Contribution

This research serves to enhance the literature of industrial automation by pushing the boundaries of subtractive manufacturing in the context of constrained glovebox environments using a flexible robotic system. The work done on simulating a nonlinear control task for this application can be applied to future industrial automation contact tasks. The software package developed allows for application simulation and stability analysis for a simplified generic planar manipulator. Therefore, this work is hardware agnostic in the sense that the same control principles are applicable to many other industrial tasks.

1.5 Organization of This Thesis

This chapter introduced the problem of material handling in a radiological hazardous environment, the history and implementation of traditional automated industrial systems, and the task-defined and system-defined performance parameters. These parameters fully define the problem and research motivation. The remainder of this report is organized as follows:

Chapter 2 summarizes previous work done in object handling by serial manipulators, specifically in the areas of grasping, motion planning, and nonlinear control methods for software compliance. The state-of-the-art for nuclear industrial automation is discussed with a summary of the DOE project ARIES.

The tools to introduce an automated system to perform this particular glovebox application process were established based on both task-defined and system-defined process parameters. **Chapter 3** involves the selection of a suitable robotic system based on prior work done in nuclear material handling.

Chapter 4 covers the general path planning algorithm for this particular contact task given a generic geometric model. The implementation of safety measures with force and torque monitoring is discussed in this chapter as well. Chapter 4 concludes with results from the automated cold reduction process demonstration using a stainless steel surrogate.

Chapter 5 covers the development of a software package to simulate a generic nonlinear robotic system for a simplified version of this particular contact task. The implementation of a nonlinear controller allows for fine closed-loop force control based on user-defined parameters.

Chapter 6 concludes this report with a summary of the research. This chapter analyzes the remaining roadblocks for both cold testing and hot cell deployment at LANL. The scope and possibilities for future work are outlined.

Chapter 2

Literature Review

The material reduction contact task specified in this report is similar to existing material reduction contact tasks performed by industrial manufacturers today. Grinding, for example, is a common automated task that subtracts material from a handled object. Therefore, it is important to understand relevant techniques for automated manipulation on the factory floor. It is also important to study how researchers within the nuclear field change these automation techniques in a geometrically confined, hazardous, and potentially radioactive workspace.

This section of the report covers both industry standards and state-of-the-art for motion planning, object manipulation and grasping for serial manipulators, as well as control strategies for software compliance. The necessary geometry background for understanding industrial robot motion planning is described in this section. Additionally, existing techniques for automated object manipulation in gloveboxes are discussed in the context of current DOE project ARIES (Advanced Recovery and Integrated Extraction System) [55]. Other automated systems used in nuclear environments that are of interest to this report are discussed briefly.

2.1 Motion Planning

Motion planning for robotic manipulators in contact tasks refers to calculation of joint-level position, velocity, and accelerations as a function of time from the start state to goal state. In industrial manufacturing tasks, it is desirable to control the path of the robot in the workspace both to perform the required task and avoid physical collisions in the robot's workspace. Motion planning for a manipulator in industrial manufacturing processes is accomplished via joint-interpolated and Cartesian EEF motion.

2.1.1 Joint Space Motion Planning

In joint-space planning, the angular position, velocity, and acceleration of each joint is calculated to make up the trajectory of the robot. Typically joint-space motion planners are used for start-to-goal point moves and does not allow for fine EEF control. In order to generate desired trajectories, a robust motion planner must be used to translate high-level desired objectives into low-level hardware motion while avoiding specified constraints such as workspace collisions and singularities within the reachable robot configuration. Figure 2.1 below illustrates a simple motion plan in a workspace.

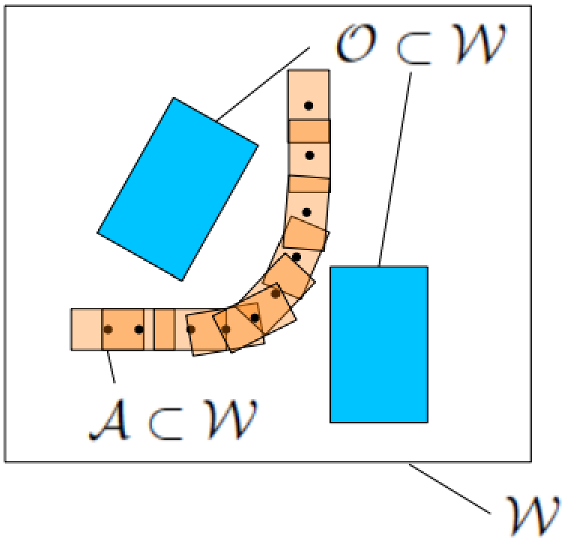

$\mathcal{O} \subset \mathcal{W}$
$\mathcal{A} \subset \mathcal{W}$
$\mathcal{W}$

Figure 2.1: Generic workspace motion plan [26]

We take the workspace W shown in the figure above as $W = \mathbb{R}^3$ where R^3 is the representation of the physical world in Cartesian space assuming dimensions x, y, and z. In this case, Cartesian space is a human-intuitive understanding of Euclidean space in the specified world/workspace frame. The obstacle region is represented by O within the workspace W while the robot is represented by A within the workspace W. The robot configuration is defined by $q \in C$, where C is the configuration space [26].

$$q = (x_0, y_0, \alpha) \tag{2.1}$$

Additionally, the obstacle region $C_{obs} \subseteq C$ is

$$C_{obs} = q \in C | A(q) \cap O \neq 0 \tag{2.2}$$

This defines the set of configurations where the robot is in collision. The free space is therefore

$$C_{free} = C \backslash C_{obs} \tag{2.3}$$

The goal of any motion planning algorithm is therefore to find a continuous path from a start state to a goal state with C_{free} in time t. The path generated will vary based on user-specified costs. Usually, the path should be short, have minimal execution time, and maximal distance to obstacles. There are many planning algorithms used to accomplish trajectory generation of start-to-goal paths, including visibility graphs, grid-based planning, and potential fields. However, sampling-based motion planners are considered to be the most successful and common planning algorithm for serial manipulators [31].

2.1.2 Sampling-based Motion Planning

Sampling-based motion planners are considered to be state-of-the-art for motion planning in industrial automation [3] [31]. The reason sampling-based planners are used as opposed to other planning algorithms such as combinatorial planners is because they do not take exact robot definition into account, are easy to implement, and work in conjunction with a separate collision detection algorithm. Because of their effectiveness, sampling-based planners are widely used in the field of robotics. Most single-query, sampling-based motion planning algorithms follow the general procedure [31] :

1. Determine free space based on configuration space, robot, and obstacle region.
2. Sample the space based on random or pseudo-random sampling for robot state milestones. Determine which points lie in the free space.

3. Create additional milestones for the start and end goal states.

4. Construct a roadmap by finding links between milestones. If the links are valid within the free space, they are added to the roadmap.

5. Connect the start and end goal state using a series of local paths specified in the previous step. The generated path will vary depending on user-specified cost functions and how the particular planning algorithm processes information. The user, for example, may specify the generated path to maximize distance between obstacles and minimize time from start state to goal state.

6. In order to choose a path from start to goal, the algorithm must choose a starting milestone (This step is known as the Vertex Selection Method [31].) to connect to a neighboring milestone. The Local Planning Method will choose which local path to take to the next milestone. This step is iterated until a complete valid path has been generated within the search graph.

Consider the following images in Figures 2.2 and 2.3 which demonstrate the Probabilistic Roadmap (PRM), a common sampling-based planning algorithm.

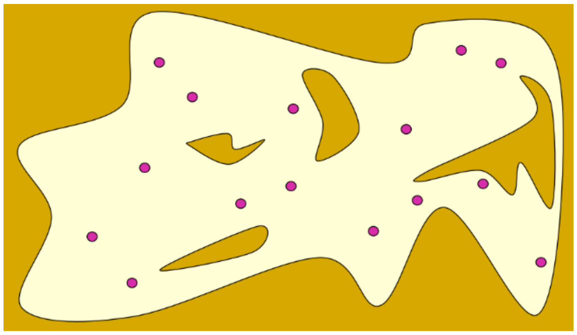


Figure 2.2: PRM milestone sampling in free space [3]

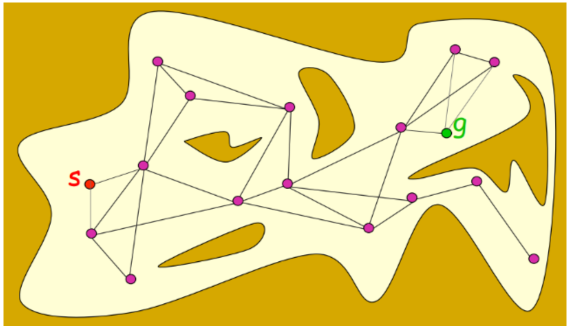


Figure 2.3: PRM milestone roadmap [3]

Figure 2.2 illustrates that the workspace has been randomly sampled for valid milestones and milestones which are invalid due to robot configuration limits and collisions with obstacles have been eliminated. The remaining milestones are located in the free space. Figure 2.3 illustrates the inclusion of start and goal states, which have been clearly defined as milestones. All milestones in free space have been connected by local paths. PRM planners typically connect a milestone to a specified number of nearest neighboring milestones or all milestones within a certain distance. After the roadmap is dense enough, construction is complete and the roadmap completes the algorithms search graph [3].

This planning process of connecting the start and goal states based on sampled milestones will be iterated until the user-specified cost functions have been met or the planning time is exceeded. Figure 2.4 illustrates a successfully-generated path from start state to goal state. Continuing the PRM example, the construction of a path from start to goal states is known as the query phase, where the path is obtained using Dijkstras Shortest Path query [23].

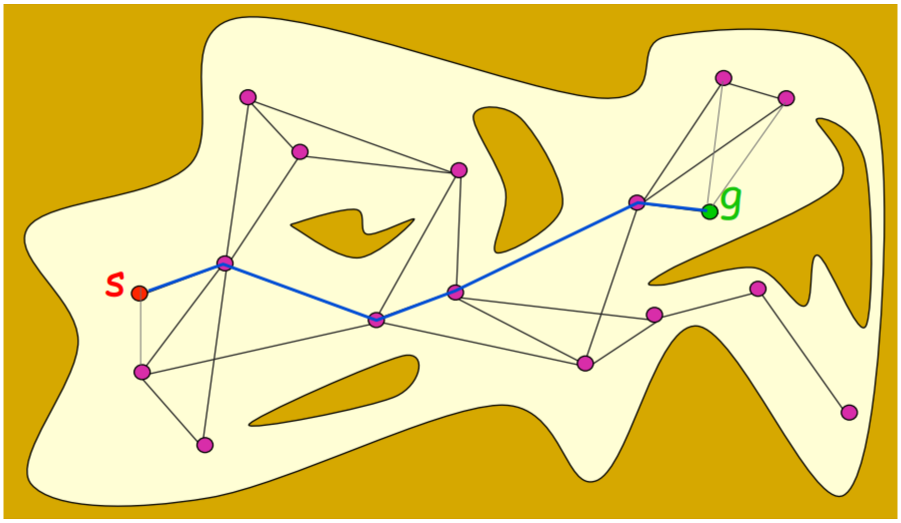


Figure 2.4: PRM generated motion plan [3]

The generic path generation in the search graph for any sampling-based algorithm is a non-trivial problem. There are many different implementations of sampling-based algorithms that solve this problem besides PRM described above. Other popular examples include the *A* algorithm*, the *D* algorithm*, and *Rapidly Exploring Random Tree algorithm* [44] [52] [30].

One of the distinct advantages of using a sampling-based motion planning algorithm is that the algorithm searches for possibilities within the solution set C_{free}, making trajectory path planning itself hardware agnostic if C_{free} is known. As stated previously, this free planning space is defined by the workspace obstacles, robot location, and robot configuration. However, the robot configuration is hardware-specific and is defined by the robots *Denavit-Hartenberg (DH)* parameters [9]. Sampling-based algorithms typically have a fast runtime while being relatively easy to implement with high-DOF robotic systems. Additionally, as planning time increases, the probability that a valid trajectory is generated approaches 100%.

2.2 Cartesian Space Motion Planning

Cartesian space planning is used to finely control the desired state of the EEF. In a serial chain manipulator, the EEF is directly linked to the Cartesian world frame via a series of frame transformations about each robot joint. In order to move the EEF to a desired pose (position and orientation in space), it is necessary to solve the inverse kinematics (IK) problem. This statement holds true for both joint-space planning and Cartesian space planning. However, joint-space planning is accomplished by smoothing a series of linking IK solutions in the free space with joint-angle position, velocity, and acceleration functions. The sequence of IK solutions provided by the planning algorithm (sampling-based or otherwise) may result in complex EEF motion in space. It is often necessary when handling hazardous material to directly control the path of the held object in space. For example, the held material must adhere to a straight path as it is placed in an oven, or an open contain must be held such that contents are not spilled. In our case, the path planning problem could be considered a hybrid of the joint and Cartesian approaches as the target points for the hole punch track a Cartesian path covering a hemisphere with a 1-3 cm spacing. A move from one point to the next can be performed using a joint trajectory sampling method, but the solution space is severely restricted when the part is located in the punch. Thus, it may be necessary to consider both joint and EEF trajectory planning strategies as a part of this effort.

Consider the following equation:

$$y_n = f_n(x_1, x_2, \ldots, x_n) \tag{2.4}$$

Where y_n is the output (such as the EEF location), and f_n is the function that gives this output given inputs x_1 through x_n. It is possible to describe any number of functions y as

$$Y = F(X) \tag{2.5}$$

If it is desired to express an instantaneous change in Y as a function of the instantaneous change in X, then the equation becomes

$$\partial Y = \frac{\partial F}{\partial X} \partial X \tag{2.6}$$

The matrix of partial derivatives $\frac{\partial F}{\partial X}$ is known as the Jacobian [7]. The Jacobian in the context of serial manipulators is a linear, time-varying mapping of velocities in X to velocities in Y. It is often desirable to describe the transformation of joint angle velocities to EEF velocity in the Cartesian space. The above equation can be described as:

$$^{o}v = {}^{o}J(\theta)\dot{\theta} \tag{2.7}$$

where the vector ^{o}v is a *6x1* vector, a combination of a *3x1* Cartesian linear velocity vector and a *3x1* Cartesian rotational velocity vector for the tool point.

The vector θ is a *6x1* angular velocity vector for each joint. The Jacobian is therefore a *6x6* linear transformation matrix. If the Jacobian is well-conditioned, joint angular velocities can be calculated using a desired Cartesian velocity vector and the inverse of the Jacobian for an instantaneous time step:

$$\dot{\theta} = J^{-1}(\theta)v \tag{2.8}$$

This relation between joint angle velocities and EEF Cartesian velocities is fundamental to industrial automation and manufacturing and allows for fine object manipulation by perturbation of the desired velocity vector. However, for redundant manipulators (>6-DOF) such as the *SIA5*, then the pseudoinverse of the Jacobian is taken as the Jacobian is a non-square matrix. The following equation summarizes the pseudoinverse calculation for redundant manipulators, where $J^{-1}(\theta)^*$ is the pseudoinverse:

$$\begin{aligned}
J(\theta)\dot{\theta} &= v \\
J(\theta)^T J(\theta)\dot{\theta} &= J(\theta)^T v \\
\dot{\theta} &= \left[J(\theta)^T J(\theta)\right]^{-1} J(\theta)^T v \\
\dot{\theta} &= J^{-1}(\theta)^* v
\end{aligned} \tag{2.9}$$

The mapping described here is a powerful tool that lets the user have control of the EEF in space during each control cycle.

2.3 Contact Manipulation

Robots in industrial manufacturing typically perform tasks in two different manipulation scenarios: *Tool-in-hand* and *Part-in-hand* [16]. *Tool-in-hand* manufacturing refers to the robot manipulating a finishing tool over a fixed object while *Part-in-hand* manufacturing refers to the robot manipulating the object over a fixed processing tool. Contact tasks such as welding may have a working tool-in-hand fixed on the robot EEF while contact tasks such as grinding may have the robot grasping a fixed part and performing a motion plan to change the grasped object properties using a tool point fixed relative to the world frame.

Tool-in-hand manufacturing usually do not require the operator to formulate a grasping strategy as the tool is affixed to the robot EEF and its properties will remain virtually unchanged for the duration of the task. This problem has been solved for tasks such as Remote Laser Welding (RLW). A typical *Tool-in-hand* contact manufacturing process strategy, including simulation and motion planning, is outlined in [15]. It should be noted that in both *tool-in-hand* and *part-in-hand* manufacturing scenarios, the objects geometric, material, and mechanical properties are typically known prior to the implementation of any process strategy.

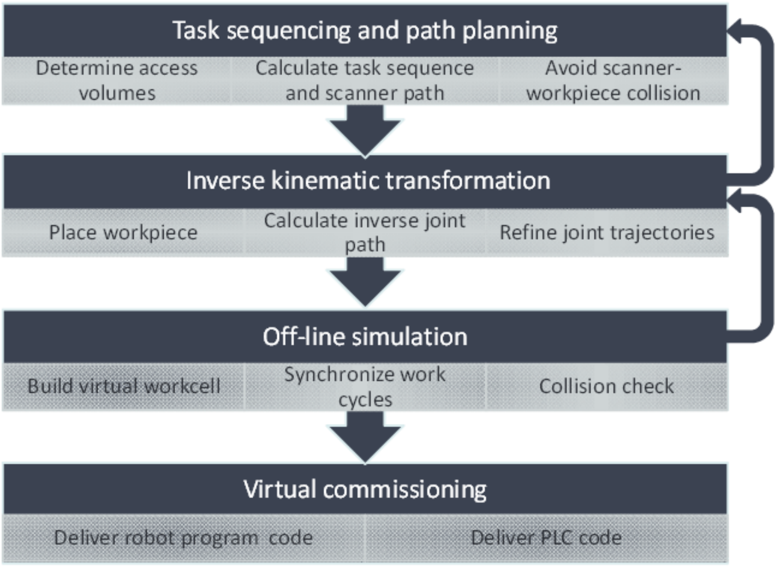

Task sequencing and path planning
Determine access volumes
Calculate task sequence and scanner path
Avoid scanner-workpiece collision
Inverse kinematic transformation
Place workpiece
Calculate inverse joint path
Refine joint trajectories
Off-line simulation
Build virtual workcell
Synchronize work cycles
Collision check
Virtual commissioning
Deliver robot program code
Deliver PLC code

Figure 2.5: RLW task process strategy [15]

However, *Part-in-hand* manufacturing usually requires the object manipulated to first be grasped by the robot arm and the part's properties may change during the task. Therefore, it is necessary for a grasping strategy to be formulated to allow the robot to manipulate the object in space. A typical grasp process will follow the procedure described by Figure 2.6.

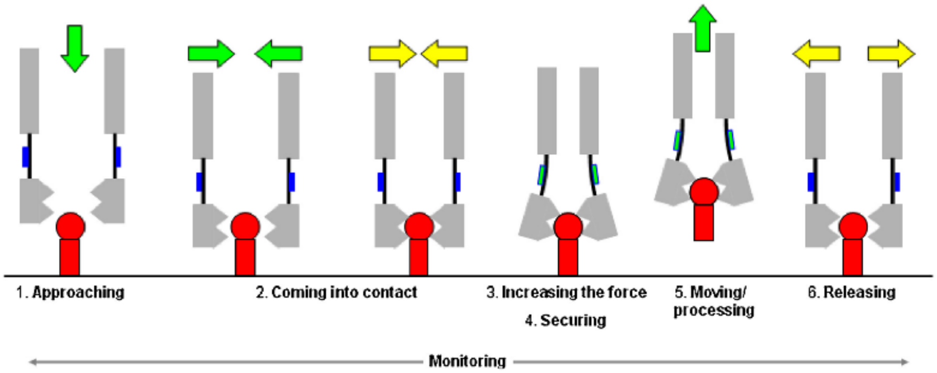


Figure 2.6: Typical grasp procedure outline [11]

Typically it is necessary to monitor the task through the grasp phase with feedback sensors to ensure grasp success. Monitoring refers to three sensing principals: Presence, Force/Torque, and Position/Orientation [11]. For example, a force-torque sensor could determine material weight loss in a *part-in-hand* subtractive manufacturing scenario while a vision sensor could determine a successful grasp based on the objects location in space.

Depending on the object the user wishes to manipulate, grasping can prove to be a much more challenging task for certain EEFs. Some factors that manufacturers consider for contact tasks is the geometry of the object, mechanical compliance of the EEF, position of the object relative to sensors and tool point of the robot, and motion planning of the robot with a grasped object.

Grasp strategies have been researched for DOE glovebox objects within the UT-NRG group. Brabek formulates an algorithm for grasp strategies based on known object primitives in the form of 3D CAD models [5].

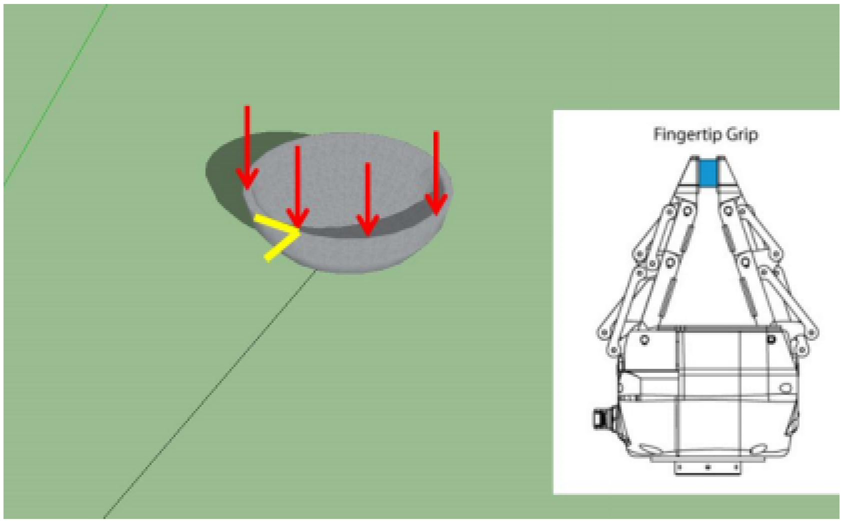


Figure 2.7: Bowl grasp approach vectors for Robotiq 3-finger gripper

Motion planning and control strategy for the robot manipulator is heavily dependent on the specific EEF hardware interface option used, requiring a task-specific interface mechanism. Assuming the arm, software, controllers, safety equipment, and fixtures are chosen properties of the configuration space, choosing a suitable end-of-arm hardware interface is the final step to formulating the complete robot workcell. The selection of a suitable EEF is described in the next chapter.

2.4 Industrial Automation Control Methods

Usually automated contact tasks require some sort of mechanical or software compliance to be exhibited by the EEF based on external forces acting upon the tool point or manipulated object. The focus of this report is on software compliance. Modern industrial manufacturers accomplish software compliance via real-time motion planning, employing nonlinear controllers at a high (>*60 Hz*) control rate to change the object position in space according to the system state and disturbance forces felt [7]. The goal of software compliance is therefore to reduce or eliminate disturbance forces on the robot tool point that might adversely affect robot joint hardware or the quality of the manufactured product.

2.4.1 A Time-Delayed Nonlinear Control Problem

Whitney *et al.* describe a closed-loop automated system for completing a weld bead grinding task using a flexible grinding disk attached to the EEF of a PUMA 560 robot [8]. Feedback arrives in the form of Kalman-filtered vision data and force-torque data from the EEF frame. The attached flexible abrasive disk exhibits a degree of mechanical compliance, which significantly affects the stability and performance of the system. The force/displacement relationship of the grinding disk plant was measured, recorded, and fit with a nonlinear (exponential) model that resembled a hardening spring with a hysteresis curve.

The software caused a significant (approximately 0.1 seconds) time delay in the force/position data feedback. The open-loop transfer function is shown below in Figure 2.8, with apparent phase lag. This phase lag is a major problem for time-delayed systems, and will destabilize a closed-loop system operating with high gains. One method of compensating is to enforce low gains. The method employed by Whitney *et al.* to overcome time delay while operating with relatively high gains is to formulate a closed-loop control law based on the Smith predictor [8] [54]. Three signals are fed back to the controller: The predicted plant model output for the future time step, the immediate signal output of the system, and a delayed system response. These signals prevent the robot from overcompensating due to a time-delayed system output while still preserving the ability to utilize high gains and maintain robustness. One of the advantages of this particular system is that the output feedforward forces could be computed at previous timesteps, allowing for fast system response.

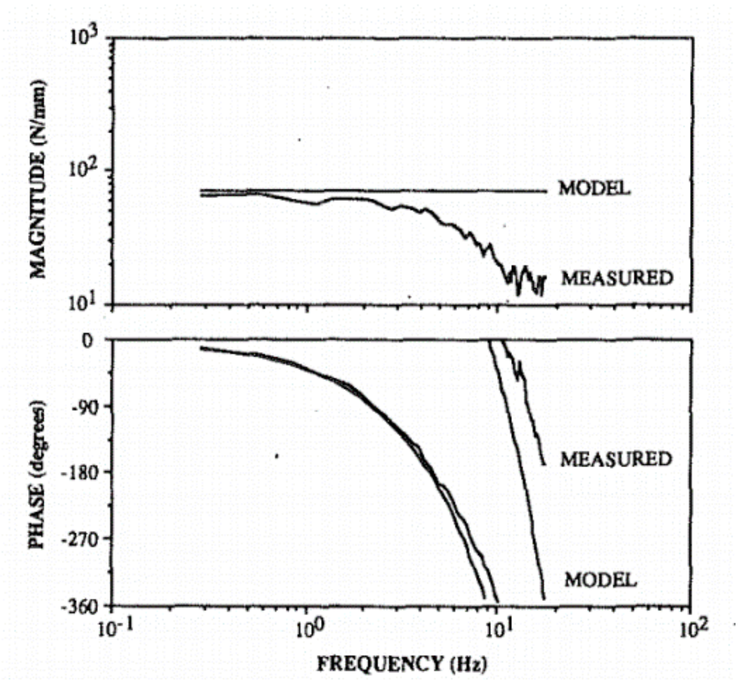


Figure 2.8: Linear open-loop transfer function response for PUMA grinding robot [8]

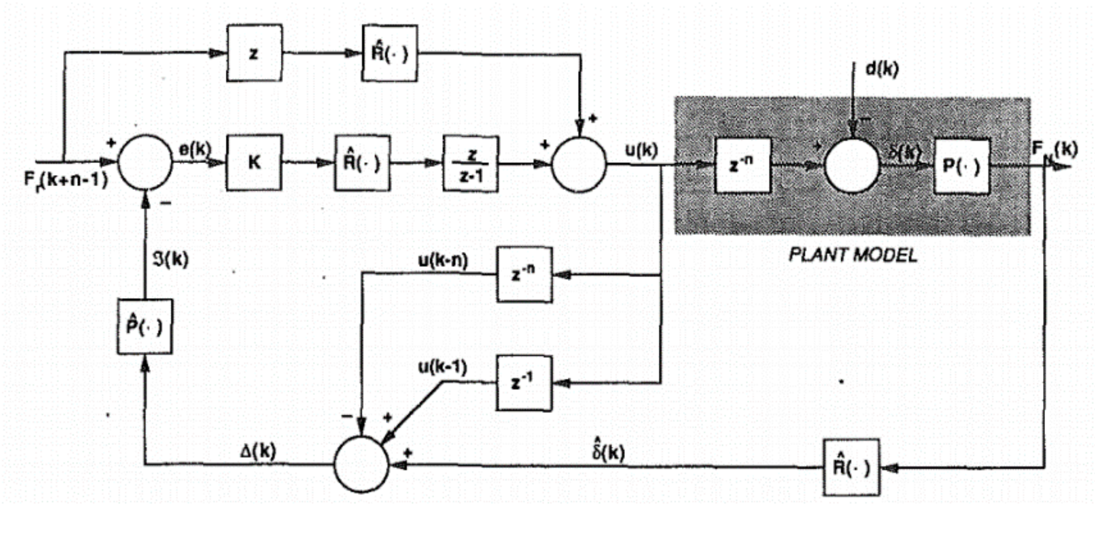


Figure 2.9: Time-delay compensated control law for force/position displacement feedback [8]

The nonlinearity of the mechanically-compliant grinding disk may introduce limit cycles. The stability of this system is directly coupled with the nonlinear gain increase introduced by the hardening spring behavior of the disk. As a consequence, the limit cycle introduced has the possibility of being unstable. A linear controller will therefore not be suitable for such a system, as a force outside of the limit cycle will cause a gain increase, introducing high oscillations. For initial conditions inside of the limit cycle, a linear controller will have a very small gain, leading to slow input response. The solution was the modelling of the nonlinear behavior of the plant with an exponential function and an appropriate nonlinear gain function with on a time-varying error signal.

2.4.2 Gain Scheduling

A common method used to control nonlinear systems is to break the nonlinear system apart into regions where a linear controller provides an acceptable closed-loop output. This method is known as *gain scheduling*, and typically utilizes a family of linear controllers that operate based on different equilibrium points of the system. There are numerous different gain scheduling algorithms, including *fuzzy logic-based scheduling*, and *Linear Parameter Varying (LPV) scheduling* [54]. All of these algorithms serve to approximately linearize a nonlinear problem. The advantage of linearizing a nonlinear problem is the ability to employ well-understood and relatively simple and computationally cheap control methods. Disadvantages include requiring accurate equations to model a given system, and susceptibility to instability.

2.4.3 Impedance Control

One approach for addressing disturbance forces acting on a manipulated object is to design a controller that mimics a human response. Most conventional controllers for nonlinear robotic systems fall into the realm of position control, or operated in the force domain. Duchaine argues that such a controller is not ideal for cooperative systems based around perturbations to the EEF [10]. Final position from force input is unknown, so small position increments commanded during each hardware control loop would lead to substantial position errors due to a buildup of small encoder errors. Additionally, compliant position-controlled systems are not as responsive as velocity position-controlled systems and can be difficult to stabilize [10].

Hogan says [20]:

> *"Seen from the environment along any degree of freedom, physical systems come in only two types: admittances, which accept effort (e.g., force) inputs and yield flow (e.g. motion) outputs; and impedances, which accept flow (e.g. motion) inputs and yield effort (e.g., force) outputs."*

His proposed impedance control describes a strategy of controlling the interaction of a physical system (the manipulator) and its environment by modelling the environment as an admittance. Therefore, a disturbance response from motion deviations for the manipulator is modelled as an impedance. A bond graph of the proposed impedance controller is shown in Figure 2.10.

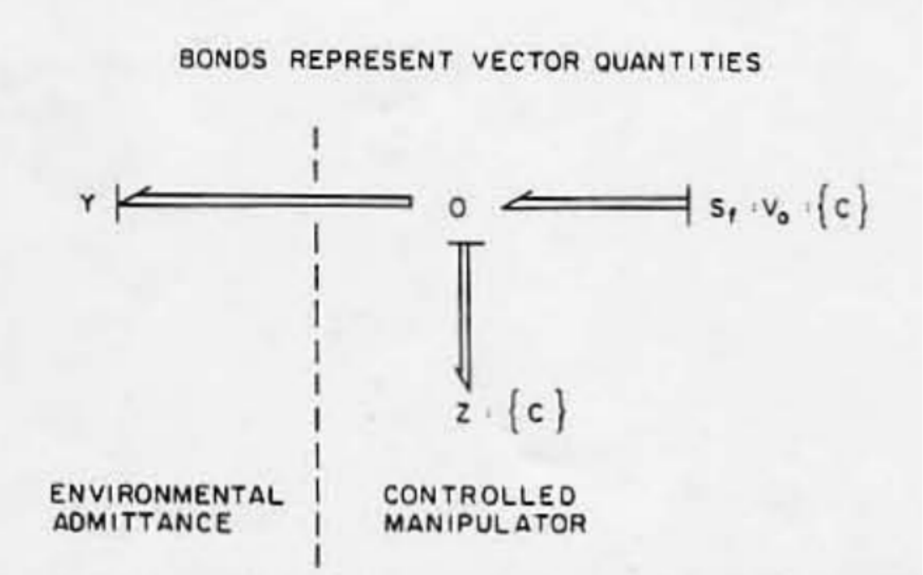


Figure 2.10: Bond graph for multiaxis manipulator interacting with environment [20]

Following this theory, a velocity and impedance can be commanded based on the manipulator interaction forces and position displacement. Many impedance control algorithms utilize this knowledge by modeling the EEF of a manipulator as a virtual spring to achieve software compliance. This method of control drastically simplifies realistic modeling of dynamic manipulator interaction, but proper selection of impedance values for complex or poorly modeled system may require a tedious trial and error process by the developer.

2.5 Automated Nuclear Material Handling

2.5.1 DOE Plutonium Pit Disassembly System ARIES

The automation of plutonium pit disassembly has been largely completed on the Advanced Recovery and Integration Extraction System (ARIES) at LANL. The purpose of the ARIES is to demonstrate the automated dismantling of nuclear weapons in a controlled glovebox environment to support the Pit Disassembly and Conversion Facility (PDCF) at the Savannah River Site (SRS) in South Carolina. The primary reason for the introduction of automation in a glovebox is to reduce dose to the radiation worker. Certain machining processes during a nuclear weapon pit disassembly will occasionally cause the glovebox operator to exceed the 500 mrem dose limit set for nuclear workers [45].

ARIES receives nuclear weapon pits and separates it into two hemishells via a lathe operation [55]. These hemishells are disassembled in the glovebox and reduced into plutonium oxide. The oxide is then physically and chemically processed and analyzed, and packaged into subcritical canisters. The ARIES mechanical components consist of a lathe, a 5-DOF robot, a glovebox, and a universal controller external to the glovebox that controls the hardware inside. Peripheral and support components consist of a material handling conveyer system, receiving module, decontamination and canning modules, nondestructive assay modules, and instrumentation system.

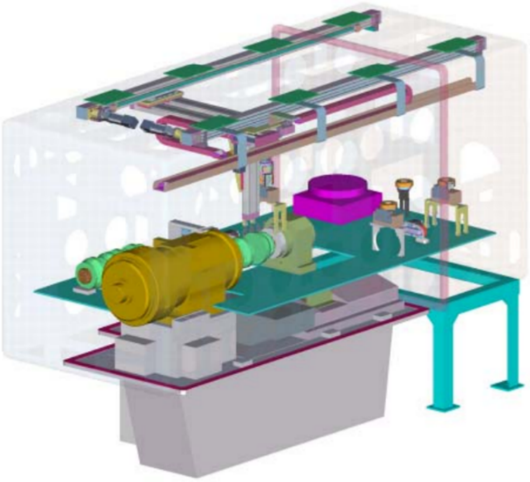


Figure 2.11: ARIES lathe, glovebox, and robot [55]

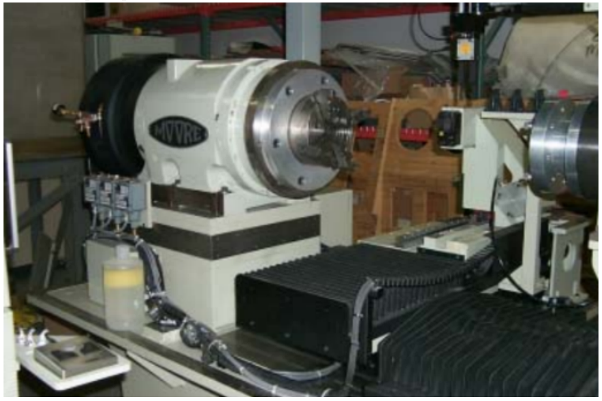


Figure 2.12: Lathe hardware [55]

Figures 2.11 and 2.12 show an early iteration of the ARIES lathe in a glovebox. The lathe is a computer-controlled 2-axis custom-built machine tool. Instead of the original tool changer, the robot changes the tools. The lathe is bolted directly to the bottom of the glovebox and uses a vacuum chuck on both axes to hold the pit in place.

The robot system is a 5-DOF gantry device. It was custom built and is composed of linear sliding sections. The X-axis is belt driven while the Y and Z axes are ball screw driven. The tool changer is located on the Z-axis rail connected to a force sensor. The force sensor is connected to a yaw-axis motor and pitch-axis motor. Robot and lathe tooling were made specifically for this glovebox application and can be swapped out using the tool changer. This entire 5-DOF gantry robot is a semi-autonomous system with preprogrammed motions used sparingly. Most of the gantry motion is teleoperated.

Separate from the glovebox lathe is the canning operation (ICAN), which is fully autonomous. The system features separate hot-cell/cold-cell/fluid-processing modules [38]. Both the hot-cell and cold-cell have a 5-DOF Fanuc LR Mate 100i robot inside. Each robot is expected to last at least 20,000 hours of glovebox lifetime. Fissile material comes through the hot-cell in a small can, which the robot places into a larger DOE 3013 can. This can is welded via an automated welding support system. The robot then places the can into an airlock, where the can is automatically decontaminated. The cold-cell robot takes the can and performs a radiation survey and weld check. The entire automated components of ARIES, including automatic support systems, contain a total of 25-DOF. The system is the most complex automated glovebox process to date, featuring varying levels of autonomy, and is an example of a hybrid fixed/flexible automation system.

2.5.2 Other Automated Material Handling Systems

Although ARIES is a prime example of automated pit reduction, the study and implementation of robotics systems has been done for other tasks in the DOE complex. Williams of the UT NRG studied the feasibility of incorporating robotics into an americium-241 conversion task from americium oxide [25]. This task would take place within a glovebox, with the robot performing tasks that a human would normally complete, such as picking and placing material containers. The purpose of incorporating such a robot would be to minimize worker dose by reducing worker time in a radiological environment and increasing worker distance from radiation sources. The report concluded that incorporating a robotic system in a glovebox for this task is feasible for a select number of tasks. Certain tasks would be difficult for a robot to complete, and would need to be completed by a glovebox worker. These tasks include door opening and breaking solids and stirring oxalates. This study is important from the perspective of this material reduction tasks as it describes the benefit of co-robotics: the act of robots working in the same workspace as humans.

Another example of co-robotics is a spherical vessel cleanout task at LANL. This task involves a robot coming into contact with the environment during task execution. Schroeder of the UT NRG developed a joint torque collision detection algorithm in order to detect unwanted environmental disturbances [28]. The results of this study indicate a need for not only model-based collision detection but real-time force and torque monitoring as well.

Dual-arm robots are also being used to dismantle materials in hazardous environments. The Dual Arm Work Module (DAWM) was deployed at Argonne National Lab to help with decontamination and dismantlement (D&D) of the CP-5 reactor in 1995 [35]. Two 6-DOF high-payload arms were mounted on a 5-DOF torso and controlled remotely. The purpose of this robot was to reduce worker dose and injury in a hazardous and radiological environment.

2.6 Literature Review Summary

The beginning of this section summarized how industrial robots move and operate geometrically in space in the context of both joint-space motion and Cartesian-space motion. Sampling-based motion planning algorithms were summarized for the purpose of understanding modern joint-space trajectory planning and execution. Motion in Cartesian space was described as a perturbation of a linear mapping between a desired Cartesian-frame pose and joint angles. Relevant grasping and nonlinear control strategy literature was surveyed. Multiple control strategies will be evaluated for the material reduction task. The methodology and hardware of ARIES, the state-of-the-art for DOE glovebox automation, was introduced. Finally, other relevant research is presented that is of particular interest for material reduction automation.

Chapter 3

Formulation of Autonomous System

The motivation for formulating an autonomous system was discussed in Chapter 1. In this section, the selection of the autonomous system is detailed. Successful integration will depend on the robustness of component selection. The final robotic system chosen is a flexible PDM that consists of a 7-DOF industrial manipulator with vacuum gripper EEF. Fixed vs. flexible automation is discussed with respect to prior attempts at automating the material reduction task. Hardware selection is justified in the context of path planning constraints. Finally, the formulation of the software framework that wraps joint-space and reactive motion planning with the hardware is detailed.

3.1 Fixed vs. Flexible Automation

The difference between fixed and flexible automation in glovebox environments was addressed by members of the NRG at UT Austin [24]. In this study, it was concluded that the level of flexibility of the integrated robotic system depends on the level of structure of both the task and the working environment. Figure 3.1 illustrates different robotic applications as they appear on the autonomy spectrum, changing with respect to the amount of task and environment structure.

Fixed automation refers to the design of a robotic system to execute a

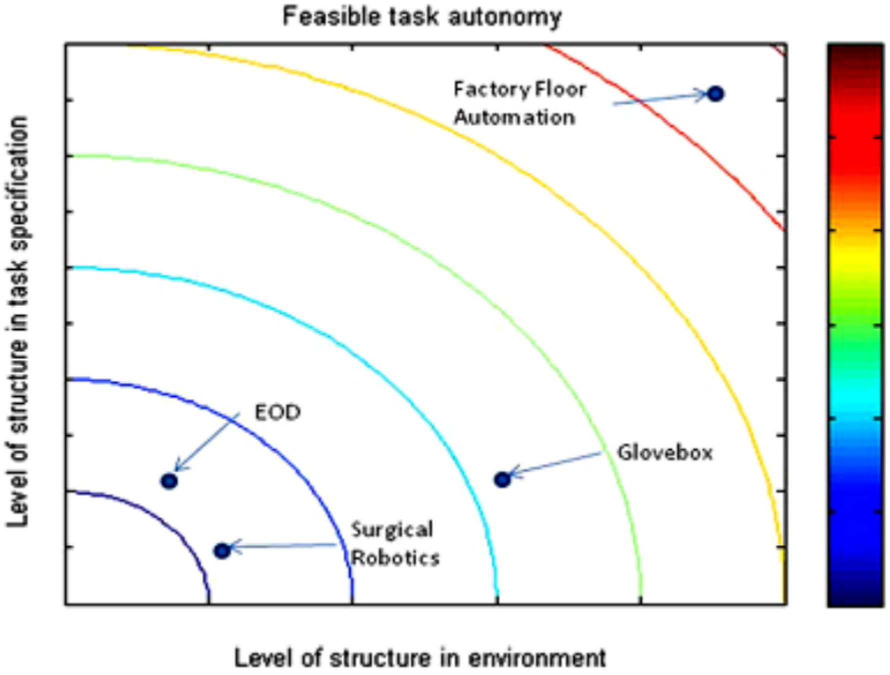
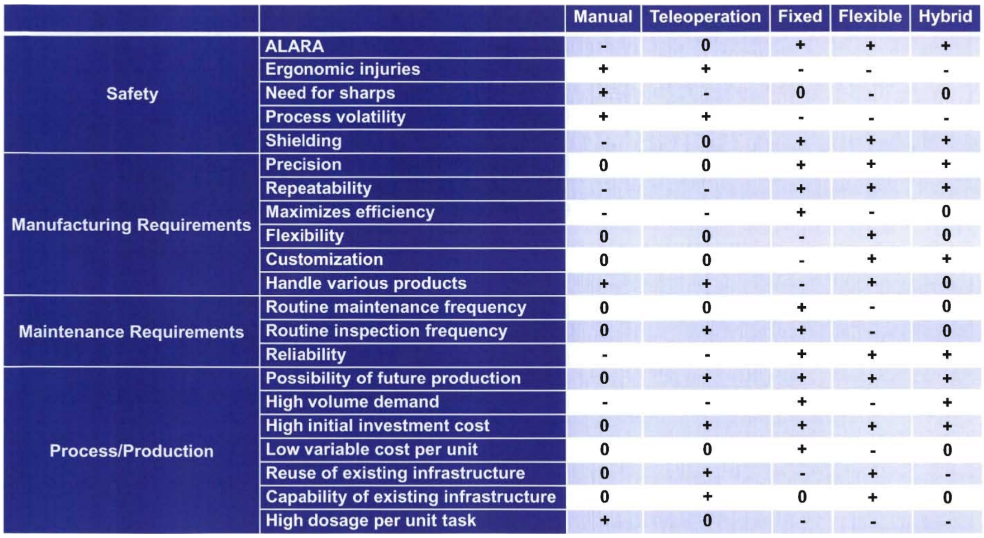


Figure 3.1: Autonomy spectrum for robotic systems [24]

small number of tasks. Usually both the tasks and environment exhibit a high degree of structure. Flexible automation refers to the design of a robotic system that can be programmed for a variety of different tasks. The degree of task and environment structure may vary per application. Flexible automation can complete tasks with a higher degree of uncertainty than fixed automation. Hashem and O'Neil claim there are multiple levels of autonomy for glovebox automation [24]. Five possible categories include manual, teleoperation, fixed, flexible, and a fixed/flexible hybrid. The advantages and disadvantages of each method are depicted in Figure 3.2.

Based on the task and design constraints specified in Chapter 1, human interaction with the robot and workspace should be kept to a minimum in order to



		Manual	Teleoperation	Fixed	Flexible	Hybrid
Safety	ALARA	-	0	+	+	+
	Ergonomic injuries	+	+	-	-	-
	Need for sharps	+	-	0	-	0
	Process volatility	+	+	-	-	-
	Shielding	-	0	+	+	+
Manufacturing Requirements	Precision	0	0	+	+	+
	Repeatability	-	-	+	+	+
	Maximizes efficiency	-	-	+	-	0
	Flexibility	0	0	-	+	0
	Customization	0	0	-	+	+
	Handle various products	+	+	-	+	0
Maintenance Requirements	Routine maintenance frequency	0	0	+	-	0
	Routine inspection frequency	0	+	+	-	0
	Reliability	-	-	+	+	+
Process/Production	Possibility of future production	0	+	+	+	+
	High volume demand	-	-	+	-	+
	High initial investment cost	0	+	+	+	+
	Low variable cost per unit	0	0	+	-	0
	Reuse of existing infrastructure	0	+	-	+	-
	Capability of existing infrastructure	0	+	0	+	0
	High dosage per unit task	+	0	-	-	-

Figure 3.2: Pros and cons for different autonomy levels [24]

adhere to ALARA principles. Manual methods and robot teleoperation should be kept to a bare minimum. Due to the varying geometric qualities of the plutonium pits, a certain degree of flexibility should be exhibited by the system in order to account for eccentricities and error. Prior work in fixed, flexible, and hybrid automation must be studied.

3.1.1 Previous Design Work

Prior attempts at automating the material reduction task have consisted of multiple undergraduate senior design projects at UT Austin in addition to the study by Hashem and O'Neil [24]. The fixed automation system proposed by the design teams consists of the same nibbler power tool (*TRUMPF TruTool N700*) used by glovebox operators for this task and two rotating holonomic wheels driven by a pair of programmable DC motors. A linear actuator provides a normal force perpendicular to the pinch point between the wheels in order to keep the plutonium pit surrogate (a stainless steel bowl) in place. The software that interfaces with the DC motors rotates the holonomic wheels to “feed” the bowl into the nibbler cutting tool opening. The nibbler power tool and holonomic wheels and fixed and mounted onto a base plate. The finished prototype can be seen in Figure 3.3.

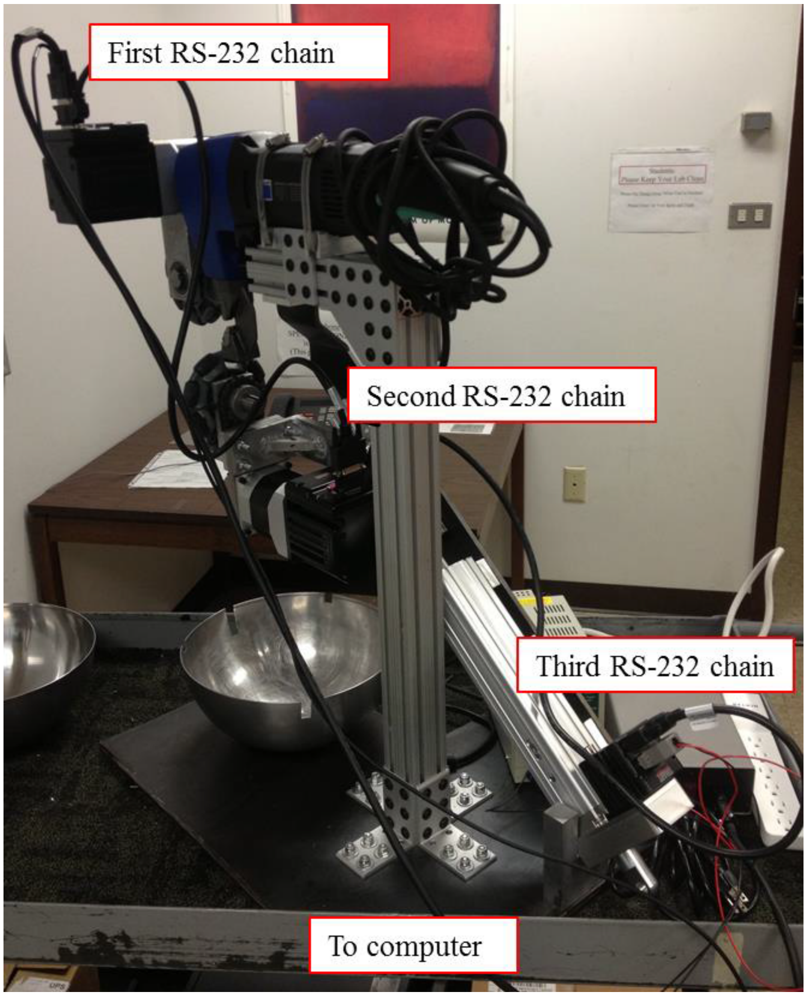


Figure 3.3: Undergraduate design team material reduction prototype

The figure illustrates the three separate actuators, each connected to each other via an RS-232 daisy chain. A separate computer relays commands to each motor during the reduction process. Although a novel design, the prototype has difficulties producing a repeatable and consistent actuating motion, as the bowl will slightly shift and move during the cutting process. The prototype also encounters difficulty rotating the bowl away from the nibbler as the tool interface will catch onto the uneven bowl edges produced by the cutting process. The process cannot account for any errors during the cutting processes, and any significant position error will result in a failed execution and the bowl will have to be manually reset. This design also requires a human operator or robot to place the bowl into the prototype for reduction. Finally, one of the most significant drawbacks of this automation method is the this design is only valid for a limited range of geometries. It would be difficult to extend this range.

The report by Hashem and O'Neil researches the possibility of a hybrid automation system, using a PDM to place the bowl into the reduction prototype. While this method effectively solves the problem of using a human operator to initially insert the bowl, it does not address the drawbacks associated with using a fixed automation system to solve the material reduction task. For these issues to be addressed, it is necessary to formulate a flexible robotic system.

3.2 Hardware

Using an industrial serial manipulator instead of a fixed automation system allows for finer object manipulation and accommodates a variety of object geometries. Choosing suitable EEFs allows for different grasping approaches to be used. A flexible system also has the potential to be utilized for a variety of contact tasks, as opposed to being specialized for one. The robot chosen for completing the material reduction task is a 7-DOF Yaskawa Motoman SIA5, which has been used extensively for research by members of the NRG and evaluated for multiple applications at LANL. This robot is a demonstratively viable PDM [19], and features extensive software support. It is also the same robot used by Hashem and O'Neil for the hybrid material reduction automation scheme, and can manipulate a bowl surrogate using both joint-interpolated moves and Cartesian jogging.

3.2.1 Micro-Punch

Glovebox operators currently use a nibbler power tool (see Figure 3.3) to manually reduce pits. The undergraduate design team also uses the nibbler as the primary reduction tool in their prototype. The downside of using a nibbling tool is that it creates sharps and fines, and will induce vibrations on the system. In order to remedy this, a different power tool was used. A micro-punch has distinct advantages over a nibbler. The most obvious advantage is that it reduces the amount of sharps and fines, creating circular or rectangular slugs of material instead. Because it operates by individually punching holes, the speed of actuation can be controlled by the operator. Vibration is only introduced into the system during the punching action. Material is more accountable this way, as material is removed with more predictable shapes and sizes.

The punch and robot arm can be seen in Figure 3.4. This picture illustrates a bowl surrogate being held by a three-fingered *Robotiq* gripper EEF. One of the more difficult problems of this reduction task is selecting an appropriate gripper, as the grasp orientation of the object with respect to the tool point will drastically affect motion planning and execution.

3.2.2 Gripper

The gripper chosen for the final implementation of the material reduction task is a pneumatic vacuum gripper. Vacuum grippers are commonly used in industrial pick-and-place applications for handling objects with smooth and known surfaces. One of the difficulties associated with grasping a bowl with a fingered gripper is the limitation of possible arm positions for proper bowl orientation into the micro punch. This problem is illustrated in Figure 3.4.

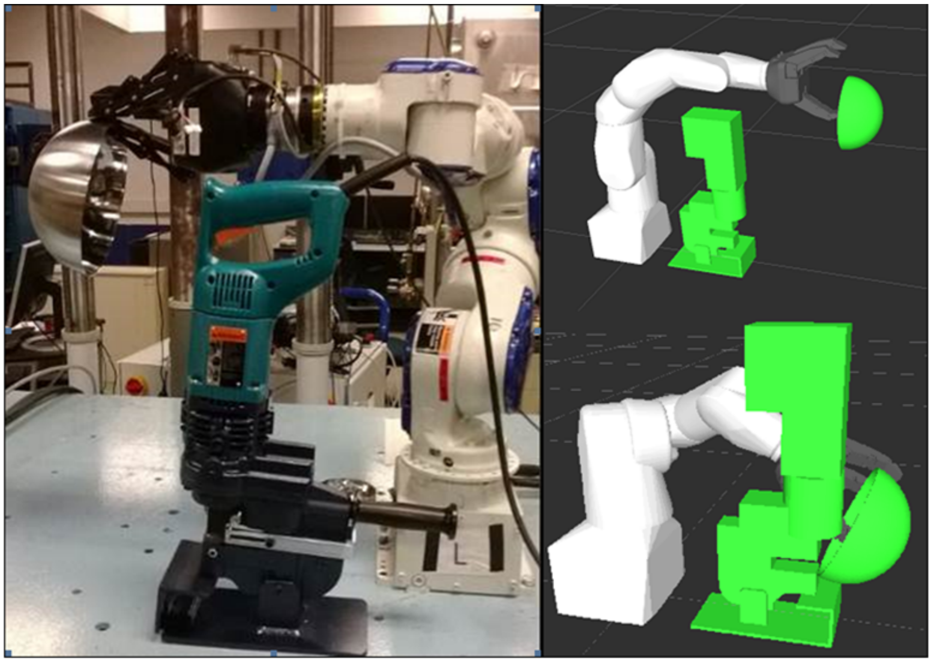


Figure 3.4: *SIA5* robot arm using a finger gripper

The simulation shows that in order to orient the bowl to fit into the punch, the arm position must be placed into a very constrained position. Based on open-loop motion planning simulation, in this orientation the robot would be required to place the bowl down and reorient the grasp pose after a few punches. This would introduce errors that could compound over time without precise feedback from multiple sensors. Alternatively, a different gripper could grasp the object with a more advantageous pose with respect to motion planning. Because the bowl can only be placed in the punch at a specific orientation, the geometric transform between the bowl and punch is fixed for every punch. An alternate tool-to-object transform allows for more possible paths or options.

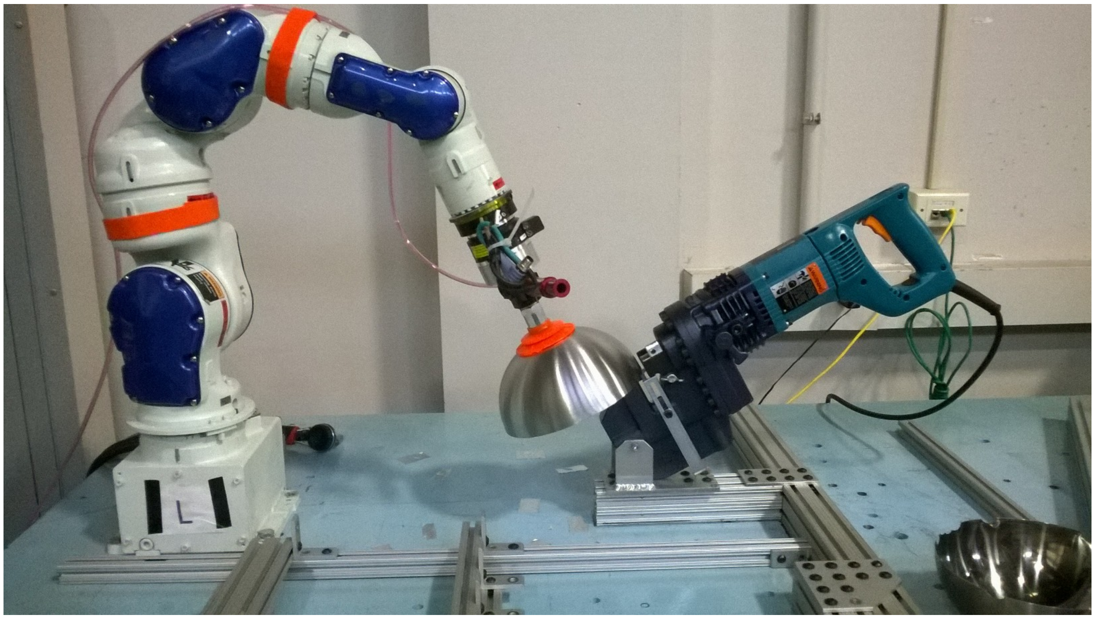


Figure 3.5: Material reduction testbed

A vacuum gripper allows for continuous motion planning, as the tool-to-object transform is altered to allow for more Inverse-Kinematic (IK) solutions to be found in the robot's free space. This is illustrated by Figure 3.5, which shows the final hardware setup, with a much more advantageous tool-to-object transform than the gripper setup. In addition to a vacuum gripper, an ATI force/torque sensor was mounted to the EEF. This allows for fine force control, and force and torque monitoring throughout any contact application.

A final summary of the system components is included in Table 3.1.

Table 3.1: Robot hardware components

1.	SIA5 7-DOF arm and controller
2.	Micro-punch
3.	Computer running Windows OS
4.	Computer running Linux OS
5.	ATI force/torque sensor
6.	Arduino microcontroller
7.	Vacuum gripper and suction cup
8.	5VDC actuated relays
9.	Pneumatic valve

3.3 Software

In order to control the robot hardware, proper software support must be implemented to issue commands and feed back data to the application at a high control rate. The *SIA5* robot utilizes a custom, high-bandwidth (approximately 500 Hz), PLC-based controller manufactured by *Agile Planet*. Real-time control and robot commands are executed by an embedded Windows CE operating system (licensed as “CeWin”). The software that handles motion planning is the proprietary AX software, also developed by *Agile Planet*. All programming is done on a separate Windows 7 operating system, and launched using CeWin.

3.3.1 ROS

In order to accomplish motion planning, robot trajectory execution, and integration with other hardware, it is necessary to have a suitable software framework. Robot Operating System (ROS) is a software framework for developing robotic applications [46]. ROS provides a level of hardware abstraction, utilizing open-source motion planning libraries and other packages for messaging, visualization, and more. By specifying the basic geometric properties of their robot, the user is able to take advantage of a vast amount of software libraries for their particular robot application. This makes it easy for users to program robots at a high level while still having control of low-level processes. ROS is utilized in the NRG in combination with AX, by writing application code in C++ or Python with ROS, and sending generated commands and trajectories to AX on CeWin for additional processing and execution. This allows users to program high-level commands easily using state-of-the-art motion planning libraries while maintaining a high hardware control rate.

Figure 3.6 illustrates the operational framework for most robotic applications in NRG. Custom ROS and AX software drivers were written by members of the NRG in order for both ROS and AX machines to communicate with each other via TCP/IP.

In this diagram, the desired planned trajectory z_d is queued by the ROS point streaming thread, and sent via a TCP-IP socket to the CEWIN system. The AX trajectory interpolator further smooths the trajectory and pushes each desired trajectory to the AX control loop. This control loop tells the controller to execute a motion v_d. The controller gives the robot a current input i_d. The robot feeds back encoder information x_a to the controller. The actual robot state is given by v_a and sent by the controller to the CEWIN driver. The monitoring thread converts this information to a readable state. This information z_d gets sent by the TCP thread back to the ROS machine via a TCP-IP socket. The state publisher updates the state of the robot in ROS.

ROS also allows for hardware besides robots to be integrated into the application code. For this task, it is necessary for the micro-punch to be automated and the pneumatic vacuum gripper valve to be toggled to open and close. This

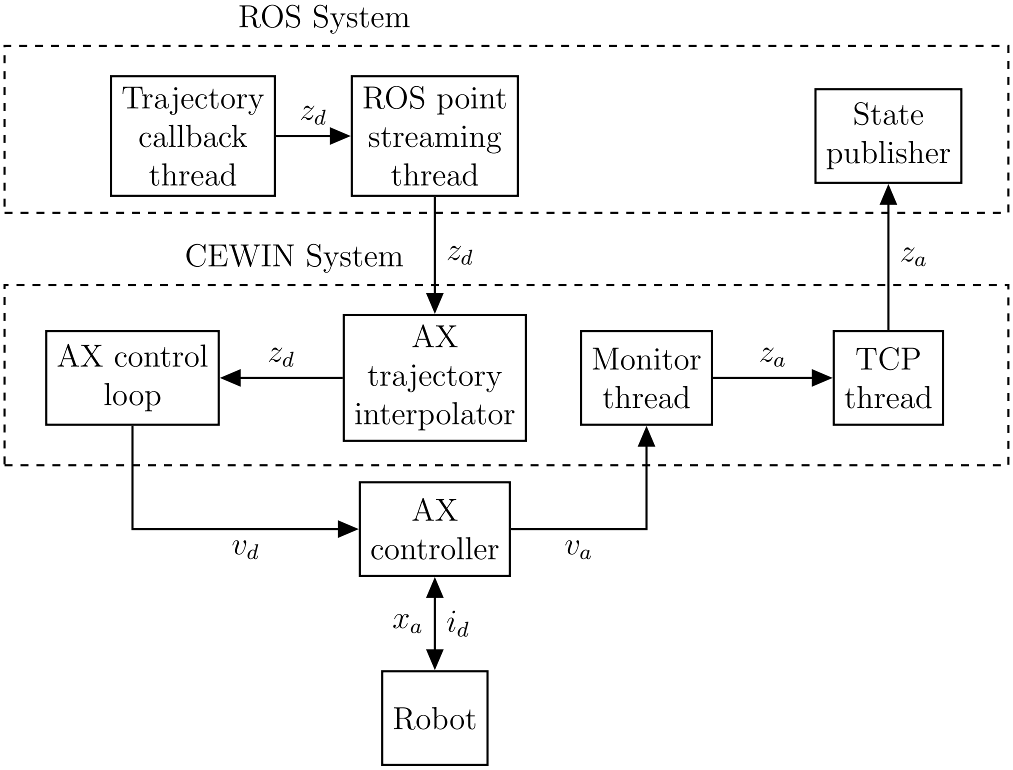


Figure 3.6: NRG robot network setup

can easily be done via publisher/subscriber nodes within the ROS framework. ROS libraries also allow for the integration of the Arduino microcontroller IDE using the ROS communication protocol [46]. By publishing commands within the application code, the user can use an Arduino microcontroller to send high and low voltage signals to different pins to actuate the gripper valve and micro-punch on and off.

3.3.2 OMPL

ROS comes packaged with the Open Motion Planning Library (OMPL), which contains a number of popular sampling-based motion planning algorithms [22]. OMPL allows the user to use these planners in their application code by inputting minimal high-level commands. All code is released under the Berkeley Software Distribution (BSD) license, allowing the source to be available for users to contribute and improve. OMPL is written with class and object abstraction in mind, allowing ease of integration for a variety of robot types. The material reduction task utilizes a version of the Rapidly-expanding Random Trees (RRT) sampling-based planning algorithm included in OMPL to achieve fast, collision-free motion plans.

3.4 Collision Detection and Obstacle Avoidance

OMPL's planners all perform model-based Collision Detection (CD). However, multiple layers of robot safety should be implemented for robot deployment in a glovebox environment. A typical control architecture is described by Table 3.2. This table lists the abstraction of control layers commonly found in a robot's system architecture from highest-level (Collision Avoidance) to lowest-level (Industrial controller collision detection).

Table 3.2: Robot safety control architecture

1.	Collision Avoidance
2.	Model-based CD
3.	Sensor-based CD
4.	Torque-based CD
5.	Redundant external system CD
6.	Industrial controller (low-level) CD

Collision avoidance is handled by the motion planning software, which will generate collision-free paths with the environment. Model-based CD is also typically handled by the motion planning software, which will take user-specified 3D models to generate a valid workspace. Sensor-based CD is accomplished with external devices, such as vision sensors or EEF-mounted force/torque sensors. Torque-based CD occurs at the driver level, typically using controller feedback from joint servos to determine unwanted contact with the environment. A torque-based CD algorithm was developed for this report and is described in the next chapter. Redundant external system CD can refer to any number of backup safety systems, including operator-controlled emergency stop (E-stop) switches or light curtains. Industrial controller CD often relies on detection of a servo fault or current limit spike. Collision detection at this level avoids major damage to robotic components but should be avoided to preserve the integrity and lifespan of these components.

Figure 3.7 illustrates the state machine that describes UT NRG CeWin driver operation. Any collision the driver detects results in braking the servos and locking the robot in place. The driver must be restarted if it enters this state. At this point the system requires operator intervention for safety purposes. A robust control architecture will implement error checking after a collision is detected and respond to events accordingly. In a majority of cases this will require operator intervention, although software drivers can be modified to allow the controller to automatically respond to various events. For example, a force detected above an operator-specified threshold could result in the robot backing away from the source of the collision. There are other methods of collision detection that could be implemented, but were not included in this report. For example, light curtains could be used in the event that the robot shares the workspace with a human worker. Tripping a light curtain would indicate a workspace violation from either

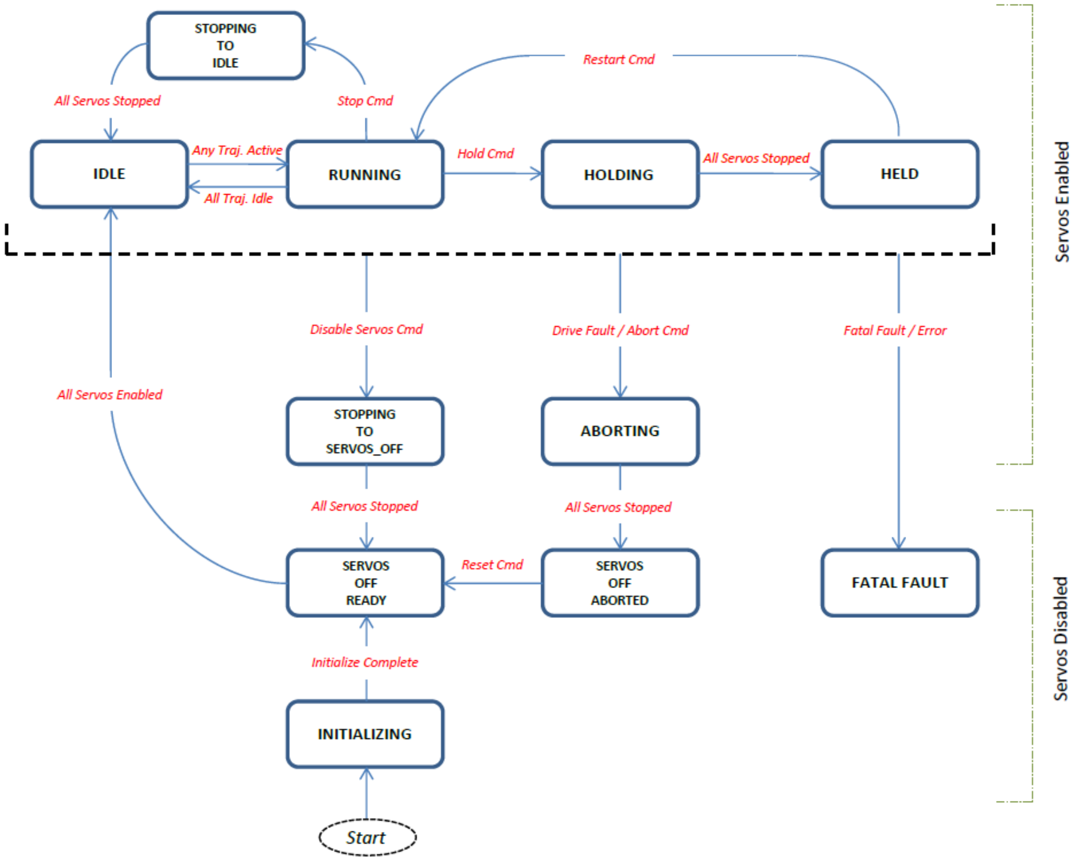


Figure 3.7: *SIA5* state machine

the robot entering the human's working range or a human entering the robot's working range. Visual feedback in the form of point cloud data could detect when an undesirable object enters the workspace. More safety features should be investigated before hot cell deployment to ensure proper robot operation and worker safety.

3.5 Summary of Autonomous System Integration

The decision to formulate a flexible autonomous system was based on both prior work and the advanced capabilities of current available software. The main robot used is a 7-DOF Motoman SIA5 robot commanded by a high-bandwidth Agile Planet Controller. The punch, gripper actuation, and motion planning take place on a machine running ROS. The commands generated by this machine are send to another machine running the virtual CeWin operating system, which sends commands directly to the robot controller. The final workspace setup can be seen in Figure 3.5.

Chapter 4

Material Reduction Demonstration

This section presents the formulation and demonstration of the material reduction task on a stainless steel surrogate. A robust motion planning algorithm is essential for task completion. The planning algorithm was implemented in C++ utilizing the ROS software framework [46]. It takes a user-specified geometric properties for a hemishell as input and generates and executes the trajectories necessary for material reduction. The utilization of ROS allows for hardware-agnostic motion plans to be generated using a high-level C++ API. The algorithm is therefore relatively simple and easy to modify as shown in the presented pseudocode (see Appendix A).

In order for this task to be automated in a glovebox, the system must be robust and incorporate redundant safety features when handling hazardous material. Reliable safeguards[1] must be implemented for all realistically conceivable failure scenarios. Even if a robot handling nuclear material collides with the workspace or punch, sensors must detect the collision and react accordingly. Thus, a joint-torque monitoring system developed at the driver level for robot glovebox applications is detailed. This section concludes with hardware demonstration results of the material reduction algorithm on a hemishell surrogate using hardware already operational at LANL. Because the surrogate used in the hardware demon-

[1] Such as path checking to ensure collision-free paths using modeled and sensed data. These safeguards were discussed in detail in the previous chapter.

stration is a stainless steel bowl, the hemishell will be referred to in this section as either a hemisphere or bowl.

4.1 Material Reduction Planning Algorithm

To keep the material reduction task simple for the operator, the algorithm takes the geometric properties of the bowl and punch tooling as inputs. The algorithm then generates a valid motion plan. The process itself is outlined below:

1. The user places the center of the bowl in a predetermined location[2] in the workspace and starts the material reduction code.

2. The robot executes a Cartesian move to the bowl in a pre-grasp pose, grasps the bowl by automatically actuating the vacuum gripper, and returns to the pre-grasp pose.

3. The robot moves the bowl to an approach pose for the punch, and executes a Cartesian move to place the bowl into the punch for reduction.

4. The punch is actuated automatically by the code to generate a circular slug of material.

5. The robot rotates the bowl using the seventh redundant joint until the first “ring” of material is cut.

6. The robot executes a Cartesian move away from the punch. It then rotates the bowl inwards so the the next “ring” of material can be cut. The robot

[2] The capability to visually recognize the bowl pose information has been developed by NRG members citeoneildiss. However, this capability was unnecessary for this demonstration.

executes a Cartesian move towards the punch in preparation for cutting the next ring.

7. This process is continued until the entire bowl is reduced.
8. The robot executes a Cartesian move away from the punch. The robot returns to its original stow position.
9. The user manually removes what is left of the bowl.
10. In an ideal scenario, the robot will reduce the whole bowl into pieces that fit into a crucible. These pieces are automatically gathered by the robot in a subcritical container.

The bowl is grasped directly at the apex of the hemisphere with a vacuum gripper. Assuming the bowl is approximately symmetric, the rotation of the last joint of the EEF aligns with the bowl's axis of rotation. This allows for the material reduction algorithm to be used with variable radii and other different geometric properties. This method takes full advantage of the robot's redundancy.

To plan a joint move in ROS, the user specifies a desired EEF frame pose. For grasping objects, planning should be done with respect to the object's frame instead of the EEF frame by calculating the EEF-to-object transform defined by the grasp. This transform remains constant throughout the entire process as the gripper is always in contact with the bowl until task completion. The robot spins the bowl about the redundant joint to feed material into the punch. A *ring* of material refers to a layer of material removed by the punch in this way. These rings are visualized in Figure 4.1.

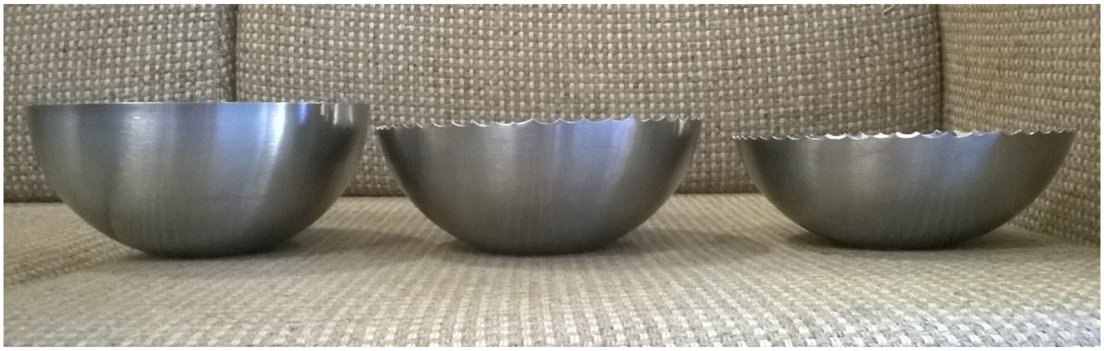


Figure 4.1: Bowl before reduction (*left*), bowl with one ring punched out (*center*, and bowl with two rings punched out (*right*)

The number of punches required to fully reduce the bowl is determined by the size of the punch and die set as well as the bowl.[3] Two angles, θ and ϕ, define object rotation in the punch with respect to the local object coordinate frame. The origin of the coordinate frame for the bowl is located at the center of the plane touching the edge of the bowl (as shown in Figure 4.2). If the bowl were turned upside-down on the table, the plane would be the same as the surface. This defines the XY plane of the coordinate frame. The angle of rotation of the redundant joint θ starts at -180 degrees and advances to 180 degrees for each ring iteration. ϕ refers to the angle offset of the center axis in the XZ plane. The location of the angles and planes are illustrated by Figure 4.2.

Th user-specified radius of the unmodified bowl is given by R. The number of rotational steps (rings) in the XZ plane ϕ is given by N_{ϕ}, leaving room for the distance $d_{suction}$ taken by the vacuum gripper suction cup. Using only R and the radius of the tool r_{tool}, these variables are calculated for each ring iteration, shown in equations 4.1 through 4.3.

[3] If we assume the bowl is a true hemisphere.

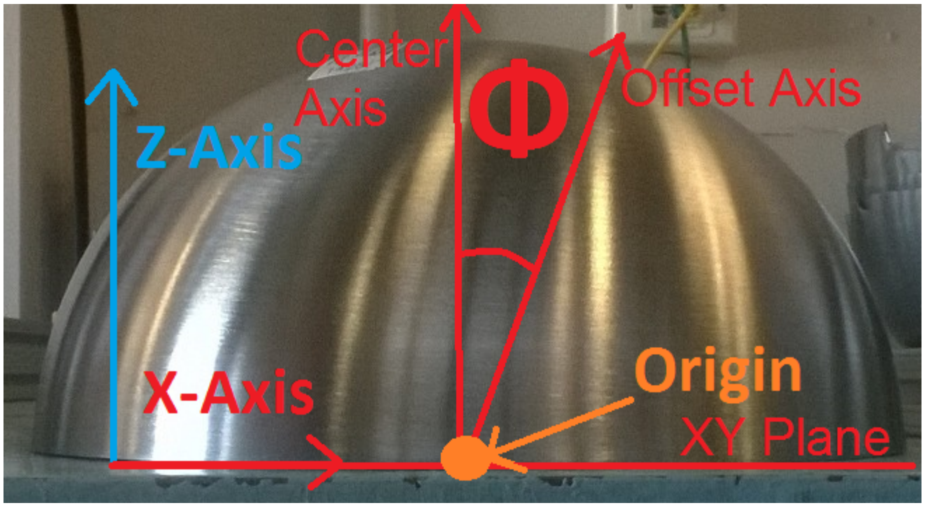


Figure 4.2: Bowl coordinate convention in the XZ plane

$$\theta = \frac{r_{tool}}{R} \tag{4.1}$$

$$\phi = \theta \tag{4.2}$$

$$N_{\phi} = \frac{R - d_{suction}}{r_{tool}} \tag{4.3}$$

4.1.1 Pseudocode

The process planning algorithm is implemented in ROS with C++. The steps are described in the previous section. The implementation is generalized using pseudocode shown in Appendix A.

4.2 Joint Torque Monitoring

The previous chapters discussed the motivation and previous work done by UT students in implementing collision detection and workspace object recognition with robots. Chapter 3 detailed the greater safety control architecture. This section describes a real-time collision detection procedure developed for contact tasks using joint torque monitoring. Hardware application of this technique is demonstrated using the SIA5 to validate this aspect of the control architecture.

Motion planning algorithms included in software libraries such as OMPL utilize 3D models in the workspace for collision avoidance during motion planning. However, discrepancies between the actual workspace and the virtual workspace used in software may result in suboptimal motion planning. In a worst-case scenario, an improperly calibrated workspace may lead to a robot arm colliding with the glovebox environment. This could possibly lead to hardware failure or to a criticality accident.

Real-time joint torque monitoring allows for automatic controller response to unwanted impact anywhere on the robot. Joint-torque monitoring is a necessary implementation for automated glovebox systems as the workspace geometrically constrains the robot. Unlike other methods of collision detection such as force-monitoring, joint torque monitoring detects collisions for each robot link; force/torque sensors typically are only mounted on the EEF of robots.

The effectiveness of torque monitoring is highly dependent on the sample rate of the software and speed of the robot. Higher control rates and slower robot motion allow a faster response time to unwanted collisions. A robotic system that utilizes a high data sampling rate will react faster to environmental disturbances faster than any human operator can. The demonstration described in this section uses the same AX controller setup as described in Chapter 3 with the same (approximately *500 Hz*) control rate.

In order to prove the effectiveness of torque monitoring, a driver-side software thread was integrated into the existing AX framework. This software thread examined the difference between the actual torque values for each joint read by the hardware and the calculated (expected) torque values for each joint based on the configuration of the robot at each time step. A large discrepancy between these two torque values for multiple consecutive time steps indicates an unwanted collision for that particular joint. The following demonstration illustrates this collision detection process:

1. The robot servos are activated at time t = 0 seconds with the robot in a stowed position.
2. A human introduces an external force on the robot by shoving it with his hands in a random location. This happens while the robot is still at rest in the stow position.
3. The robot starts a joint-interpolated move from the stow position at time t = 16 seconds.
4. A human introduces an external force on the robot by shoving it again in a random location. This happens while the robot is in motion (Figure 4.3).
5. The demonstration ends with the robot at rest at time t = 20 seconds.

The absolute value of the difference between the expected and actual joint torques for each joint is logged for each time step of the demonstration. The result

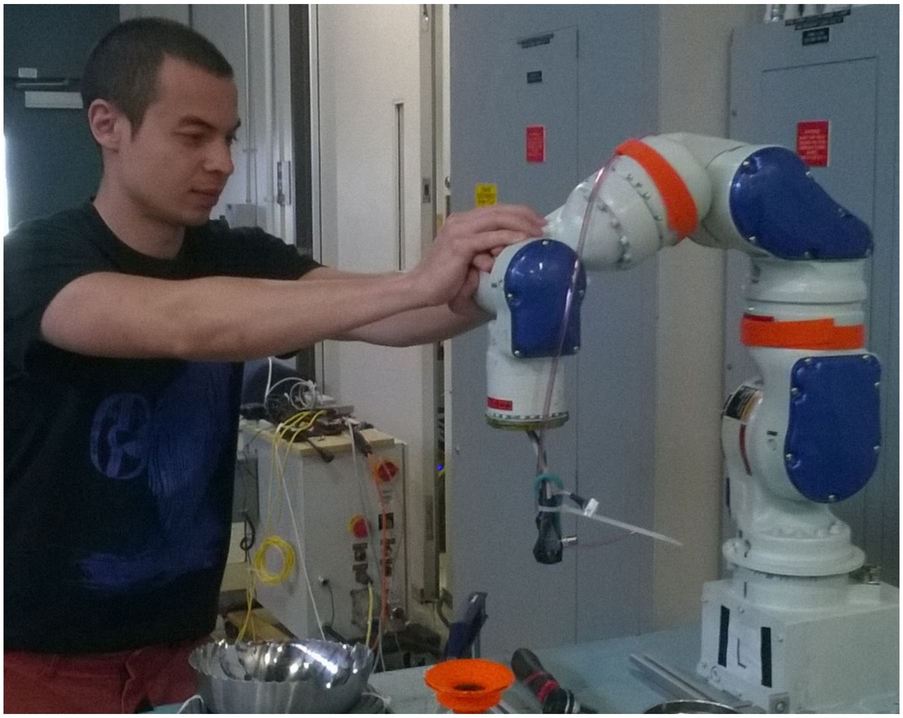


Figure 4.3: Inducing an unmodeled disturbance on the robot during motion

of one of these demonstrations can be seen in Figure 4.4. The torque difference for joint 'L' (the second revolute joint in the robot chain) increases rapidly at approximately time t = 9 seconds. A second collision can be seen at approximately time t = 19 seconds. However, the raw data is inconsistent and noisy. In order to better distinguish between normal robot operation and a collision, data processing becomes necessary.

4.2.1 Filtering Torque Data

Many modern systems that incorporate digital readings from sensors will use some form of signal processing to increase system performance. During typical robot use, it can difficult to distinguish between normal operation and unwanted collisions due to signal noise, especially when the robot is in motion. A digital filter can reduce the noise and help clearly define unwanted joint collisions. The filter used in this demonstration is an Exponential Moving Average (EMA) filter, a simple moving average filter variation [43]. A moving average filter applies a weighting factor to data taken at the current time step as well as previous time steps, effectively producing a convolution of a data set over a period of time. An EMA filter applies an exponentially-decaying weighting factor to all data taken. Equation 4.4 describes the calculation of an EMA at any time step.

$$\begin{aligned} S_1 &= Y_1 \\ S_t &= \alpha Y_t + (1-\alpha) S_{t-1} \end{aligned} \tag{4.4}$$

In this equation, S_t represents the EMA value at time step t, and S_{t-1} represents the EMA value at the previous time step. Y_t represents the raw data value at time step t. α is a user-defined weighting value between 0 and 1. A larger value of α will place higher priority on newer data samples while discounting older data samples faster. Because the weighting factor of any time step will always be greater than 0, the EMA can be seen as a low-pass, infinite impulse response filter. In this particular demonstration, the α value was chosen to be 0.05. The filtered torque difference values for joint 'L' can be seen in Figure 4.5.

By monitoring filtered joint torque differences, collisions are detected by specifying a torque difference threshold for each joint. By examining torque data for each joint for multiple controlled collisions, a threshold value is specified in the hardware driver. If a collision is detected for multiple time steps, the robot enters an emergency-stop mode where the servos brakes engage. At this point the operator must restart the robot and manually jog the robot into a safe position for application reset.

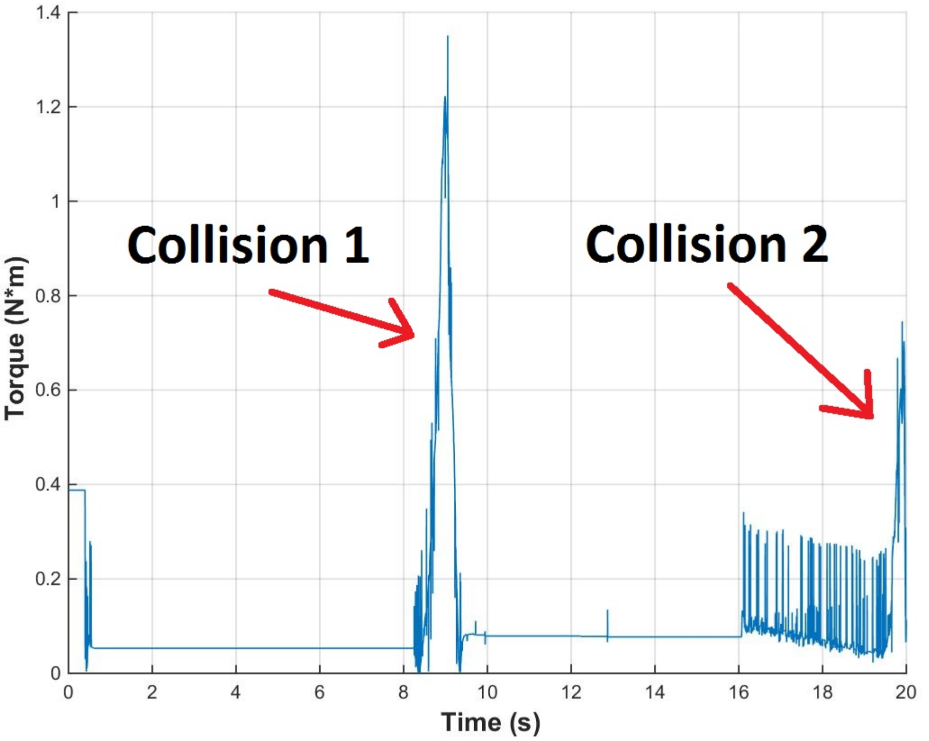


Figure 4.4: Unfiltered torque difference values for joint ’L’ for two collisions

It should be noted that although joint torque monitoring was developed for this report, futher safeguards could and should be implemented as future work. Redundant safety measures will most likely need to be developed for actual automated system installment in a hot DOE glovebox. Force monitoring via a force/torque sensor mounted on the EEF could more finely monitor external forces on the end link of the manipulator. A closed-loop control law could reduce forces on the EEF when the punch comes into contact with the bowl. Such a control law is simulated for a simplified robot arm in the next chapter.

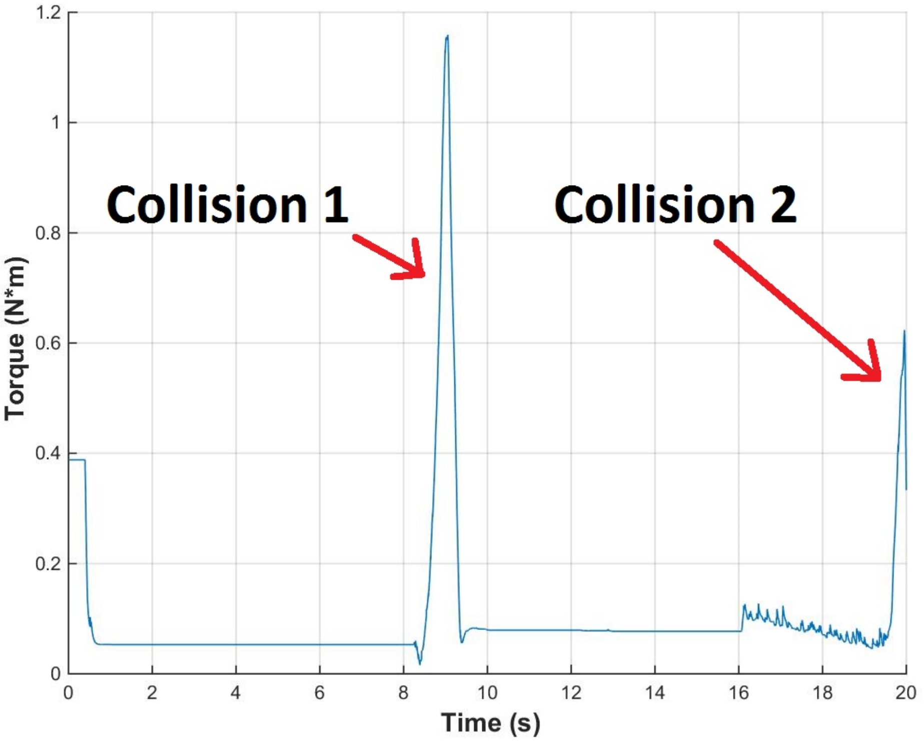


Figure 4.5: EMA-filtered torque difference values for joint 'L' for two collisions

4.3 Demonstration Results and Conclusions

A demonstration of the material reduction algorithm has been completed on the *SIA5*. The variables used are listed in Table 4.1.

Table 4.1: Material reduction algorithm parameters

r_{tool}	25/32 `inches`
R	4 `inches`
θ	0.2 `rads`
ϕ	0.2 `rads`
N_{ϕ}	6
$d_{suction}$	0.5 `inches`

The micropunch tool used a circular $^{25}/_{32}$ inch custom punch and die set. An 80/20 aluminum framing mechanism held the bowl in place for an adequate robot grasp. For a 4-inch radius hemispherical bowl surrogate, the total run time of the program was approximately 30 minutes to complete 6 ring iterations, with the punch speed and die size as the limiting factors. The results of the program can be seen in Figure 4.6. The remaining stainless steel piece held by the vacuum gripper would have to be manually reduced by a human worker if it cannot fit inside of the crucible. However, the majority of the work at the end of the application has been done by the robot.

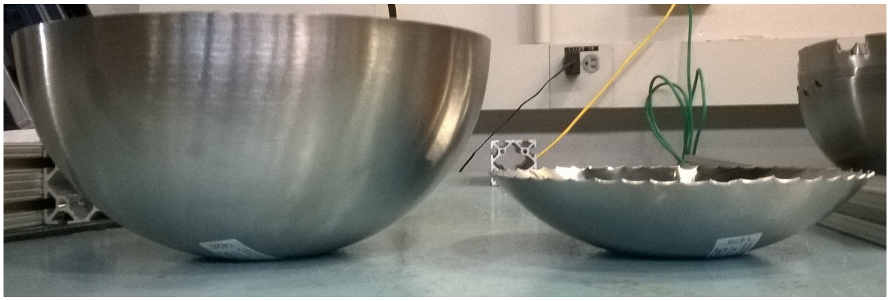


Figure 4.6: Bowl before reduction (*left*) and the remainder of the bowl after the material reduction algorithm run (*right*)

The successful demonstration of this material reduction algorithm proves the viability of automating this glovebox process using a relatively low-cost PDM. By removing the human operator from the majority of the task, automation significantly decreases worker dose rate while maintaining high task performance and repeatability. The material reduction algorithm remains valid for hemispherical surrogates for a certain range of surrogate geometries by utilizing robust motion planners within the ROS-side OMPL. However, this demonstration does not conclude the development of material reduction automation. Extensive future work must be done in a cold glovebox environment before successful deployment into a DOE hot cell. Specifically, the work that should be done in the near future includes:

- Generalize the algorithm to accept variable geometries besides a hemisphere. Perhaps incorporate path planning by interpolating the surface of a variable 3D CAD model.
- Validate hardware procedure for a variety of bowl geometries.
- Integrate vision capabilities such as object pose estimation based on point cloud data.
- Induce other faults that may occur during hot cell scenario to validate procedure.
- Develop a material capture system.

It is likely that a possible robot hardware platform switch or EEF hardware changes and modifications will occur to better suit a proper hot environment. Future work is discussed more extensively in Chapter 6.

Chapter 5

Nonlinear Control Simulation

The material reduction demonstration was successfully conducted without significant software feedback[1] or reactive motion planning, as the mechanical compliance of the vacuum gripper was sufficient for task completion. However, this particular hardware setup will not necessarily be used in the final implementation of the material reduction task in a hot glovebox. Therefore, a form of robust software feedback for a multi-DOF robotic system must be investigated. In a hot glovebox, the degradation of components in a radiological environment may necessitate the use of a different EEF. Variable shear forces acting on the object held by the robot may warp the object and thus require a reactionary force control law. Therefore, a real-world implementation of the material reduction algorithm should account for variable stiffness of the EEF and tool-point interaction forces via software compliance.

This section of the report details a simulation of a simplified multi-DOF arm and external forces acting on it. Creating a robot arm simulation for a simplified system allows for fast control design and analysis before hardware implementation. In this case, the goal is to minimize interaction forces on the EEF during a punch by controlling the input torques in each joint. When the punch tool interfaces with the surface of a bowl held by a human, the intuitive reaction

[1] Sensor feedback to the controller was used for collision detection and safety only.

is to allow the bowl to move spatially in the direction of the punching force. The following control scheme modules were developed in this regard:

1. A high-level EEF force/velocity control law. A reactionary force on the EEF translates to a desired EEF Cartesian velocity vector.
2. An Inverse-Kinematic (IK) mapping of desired EEF states to desired joint states.
3. An outer-loop linear tracking control law that translates joint state errors to desired linear input.
4. An inner-loop nonlinear control law that decouples each link of the robot and input-output linearizes joint states to input torque.

A simplification of the robot is given at the beginning of this section, followed by formulation of the robot dynamic system model based on Euler-Lagrange equations. Two different nonlinear controllers were simulated, compared, and analyzed. An outer-loop linear PID control law is formulated, followed by a joint-space to Cartesian mapping scheme. Finally, the high-level EEF force control law is introduced to simulate the punching force applied to the robot EEF by the hole punch operation.

5.1 Robot Simulation Simplification

In this section, the formulation of a robot trajectory control law is presented by using a simplified version of the current hardware setup. Consider Figure 5.1. In this picture, it is apparent that the motion planning algorithm only utilizes the workspace as an approximately planar space for the majority of the demonstration. The robot can be further simplified into a planar free body diagram.

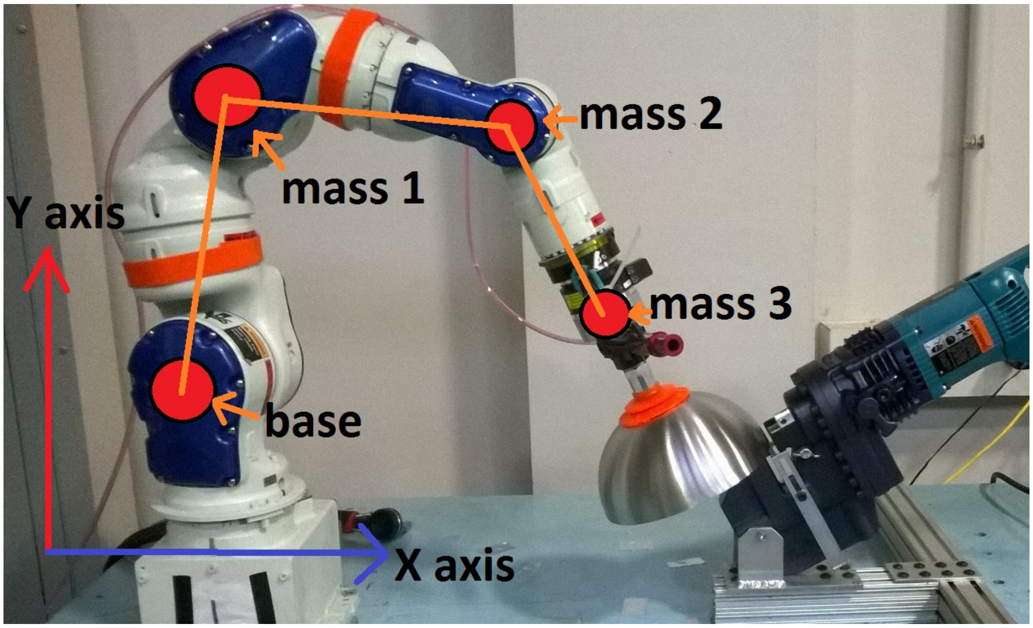


Figure 5.1: Simplification of material reduction workspace

Figure 5.2 above shows a simplified version of a generic multi-DOF robot arm that lies in the planar space. There are a number of assumptions that are made by simplifying the system in this manner.

5.1.1 Assumptions

The following assumptions are made for the purposes of full simulation of the autonomous system. Although the purpose of this chapter is to provide a detailed analysis of a particular planar robot arm, the methodology and control law can be applied to any generic, rigid, multi-DOF system.

1. A multi-DOF robot arm can be approximated by lining up its joints in

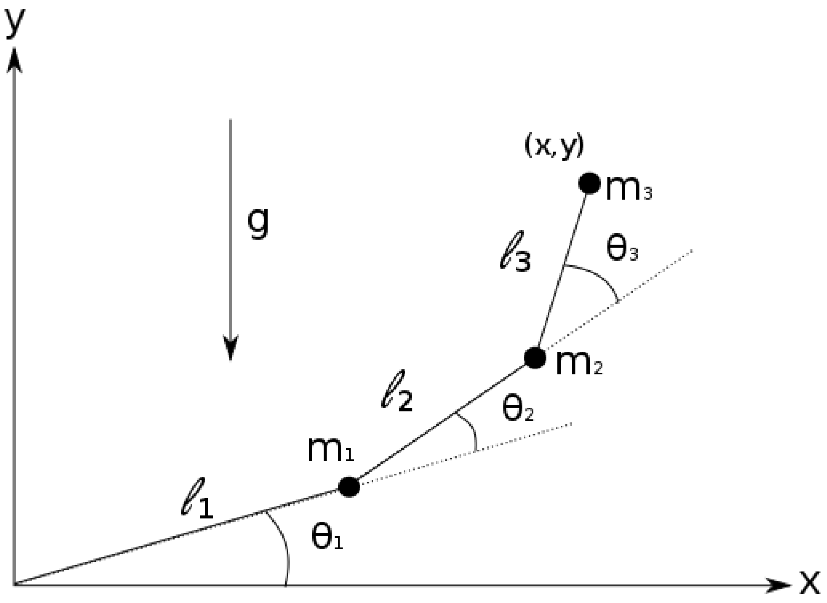


Figure 5.2: Free body diagram of *3-DOF* planar robot arm

a planar fashion. This constrains and simplifies motion planning to a 2D surface.

2. The robot link masses are approximated by point masses at the ends of each link.

3. In order to account for the inertia matrix of the object held by the EEF, the mass of the object is added to the point mass of the last link for simplicity. This neglects another link to be added to the EEF, but is necessary to simplify the system state equations.

4. The input torques are applied at each joint.

5. Friction consists of dynamic and viscous friction at each joint.

6. The formulation of the system state equations from the Lagrange-Euler dy-

namics of the robot account for the inertia, gravity, centripetal force, and disturbance torques but does not account for quantization, noise, etc.

7. The model and controllers are continuous in nature.
8. This simulation disregards input torque saturation.

Using a simplified robot model allows for faster controller design and gain tuning. It also provides a better understanding of the low-level dynamics of the robot arm. These same controller designs and concepts can be applied to a full 7-DOF arm and implemented on hardware after extensive testing and simulation on a simplified system.

These assumptions are acceptable for modeling the dynamics of the robot model. However, there are certain technical challenges that remain unaddressed. Input saturation is not taken into account during the simulation so controller parameters can be adjusted after simulation should saturation become an issue. The main technical issue that is not addressed in this section is the discrete nature of hardware systems. The entire simulation is continuous, and effects such as noise, aliasing, discrete control, and quantization are not covered here. Chapter 6 contains detailed discussion of future work.

5.2 Robot Dynamics

Before implementing any sort of control law for a mechanical system, it is useful to model the system mathematically. To construct a model describing the dynamical behavior of a generic planar manipulator, Lagrange's equations of motion are used. The crux of this derivation lies in formulating a relationship between the energy of the system and the input force or torque required.

Lewis *et al.* gives the basis for *Lagrange's equation of motion* [14]:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \tau \tag{5.1}$$

In this equation, the vector q is a set of coordinates, τ is a set of torques, and L is the *Lagrangian*, given by equation 5.2 as the change in kinetic and potential energy of the system.

$$L = K - P \tag{5.2}$$

Consider again Figure 5.2. In this system, it is desired for the input torques to change the output joint angles. Therefore, q can be specified as a vector containing the joint angles θ_i and τ can be specified as a vector containing the input torques τ_i.

$$q = \begin{bmatrix} \theta_1 & \theta_2 & \dots & \theta_n \end{bmatrix}^T \tag{5.3}$$

$$\tau = \begin{bmatrix} \tau_1 & \tau_2 & \dots & \tau_n \end{bmatrix}^T \tag{5.4}$$

Using equation 5.1, it is necessary to compute the kinetic and potential energies as a function of the joint states $q, \dot{q}, \ddot{q}$. Calculation of equation 5.1 gives a set of n coupled nonlinear differential equations, where n is the DOF of the robot arm. These equations can be arranged into the form given in equation 5.5.

$$M(q)\ddot{q} + V(q, \dot{q}) + G(q) + F_v(\dot{q}) + F_d(\dot{q}) + \tau_{dist} = \tau \tag{5.5}$$

Equation 5.5 is the standard form of Lagrange's equation for a robot arm and is used to model the dynamics of any multi-DOF manipulator. The full derivation of Lagrange's equation for a 3-link planar rigid robot can be seen in Appendix B.

The first term M represents the $n-$by$-n$ inertia matrix, V is the $n-$by-1 vector describing the Coriolis and centrifugal forces, and G is the $n-$by-1 gravity vector. Friction for each joint is formed by a $n-$by-1 viscous friction vector F_v and a $n-$by-1 dynamic friction vector F_d. The torque τ is the $n-$by-1 input vector, while τ_{dist} is the $n-$by-1 disturbance torque vector. In a general nonlinear system equation, this input would be described by the term u.

It is acceptable to model disturbances as a torque because any external force acting on the robot will be translated into joint torques.[2] An important property of this equation is the fact that the inertia matrix M is square, symmetric, invertible, and positive definite. If τ is the control input and $\ddot{\theta}$ is the desired reference output, the system is both controllable and observable [51].

Finally, it is also important to note that uncertainty exists for measured constants such as link masses m. Therefore, the nonlinear terms M, V, and G are simply approximations $\hat{M}$, $\hat{V}$, $\hat{G}$ of the actual system states. In the following simulations, mass uncertainties are noted as δm_i.

5.3 Feedback Linearization by Computed Torque Method

One of the most common methods of compensating for nonlinearities in a control design problem is to transform the model into a linear problem. This is

[2] In other words, $\tau = J^T F_{dist}$, where J^T is the transpose of the Jacobian matrix.

partially accomplished by “brute force” methods such as gain scheduling, which rely on linear approximations. However, a more elegant solution is to mathematically cancel out all nonlinearities using a single control law that is appropriate for the controllable and observable subspace of interest. This is known as feedback linearization, an exact linearization method that accounts for the internal dynamics of the system and doesn’t rely on linear approximations. If a nonlinear system model can be successfully feedback linearized, the user can implement their choice of linear control law (pole-placement, PID, etc.).

Slotin *et al.* summarizes the control design based on input-output linearization in three steps [51]:

- *differentiate the output y until the input u appears*
- *choose u to cancel the nonlinearities and guarantee tracking convergence*
- *study the stability of the internal dynamics*

Suppose the nonlinear system is of the general form 5.6.

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x) \end{aligned} \tag{5.6}$$

The process of output differentiation is given by equation 5.7. This equation involves taking Lie derivatives of the output function $h(x)$ and input function $g(x)$ with respect to the system function $f(x)$.

$$\begin{aligned} y^r = {L_f}^r(h) + L_g({L_f}^{r-1}(h))u \\ L_g({L_f}^{r-1}(h)) \neq 0 \end{aligned} \tag{5.7}$$

The variable r refers to the relative order of the system, which is when the input u appears in the rth derivative of y. If the relative order of the system is equal to the nonlinear system order n, the relative order of the system is said to be *well-defined*. This is the case of a rigid body manipulator, where the relative order of the system is equal to the system order ($r = n = 2$) [51]. Because of this property and since the inertia matrix M is invertible, the feedback-linearization process for a rigid arm is trivial. Consider again the standard form of the arm dynamics equation 5.5. This equation can be rearranged into the form given in equation 5.8.

$$\ddot{q} = v = M^{-1}(-V - G - F_v - F_d - \tau_{dist} + \tau) \tag{5.8}$$

This control law is commonly referred to as the computed torque method, or the inverse-dynamics equation [14]. This control law is illustrated in Figure 5.3, where the outer loop feedback control law is a user-specified linear control law.

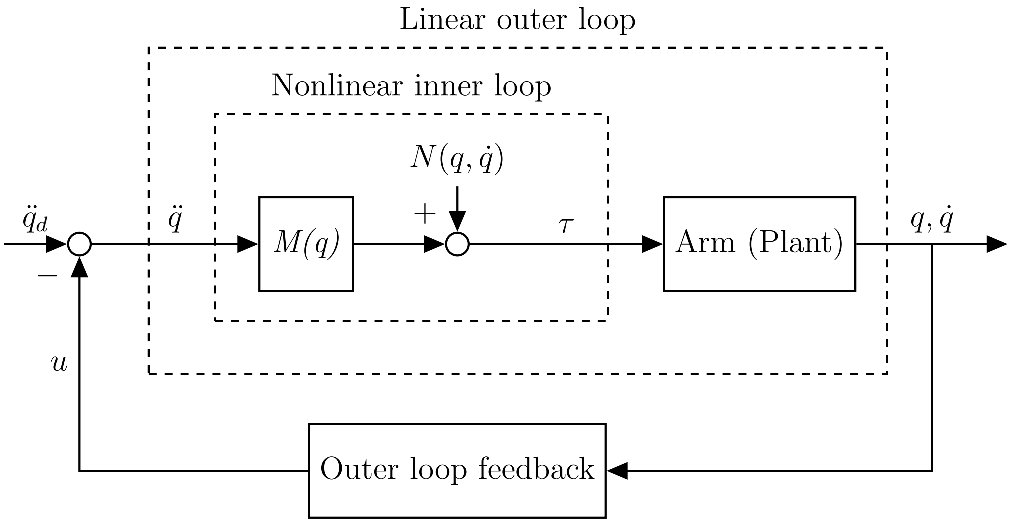


Figure 5.3: Block diagram of computed-torque control model [14]

5.3.1 PD Computed Torque Law

Consider the case where the desired linear control law is a PD controller. The general computer torque control law introduced in equation 5.8 would take the form of equation 5.9.

$$
\begin{aligned}
v &= -K_d\dot{e} - K_p e \\
\tau &= M(q)(\ddot{q} + K_d\dot{e} + K_p e) + V(q,\dot{q}) + G(q) + F_v(\dot{q}) + F_d(\dot{q})
\end{aligned}
\tag{5.9}
$$

Where the desired states $[\ddot{q}, \dot{q}, q] = [\ddot{\theta}, \dot{\theta}, \theta]$ and tracking errors are represented by $[\dot{e}, e]$. A simulation of a robot arm controlled by a PD computed torque controller as in Figure 5.3 was constructed using MATLAB using parameters given in Table 5.1. All simulations featured in this chapter use values from this table, so any unused values for a particular simulation is used for another simulation.

Table 5.1: 3-DOF simulation parameters and system bounds

q_0	=	$[\frac{\pi}{2}, -\frac{\pi}{2}, -\frac{\pi}{4}]$ rads	l_1	=	0.4 [m]
$\dot{q}_0$	=	$[0,0,0]$ rads/sec	l_2	=	0.3 [m]
$\ddot{q}_0$	=	$[0,0,0]$ $\texttt{rads}^2$/sec	l_3	=	0.3 [m]
$\hat{m}_1$	=	2.00 [kg]	δm_1	=	0.100 [kg]
$\hat{m}_2$	=	1.50 [kg]	δm_2	=	0.075 [kg]
$\hat{m}_3$	=	1.00 [kg]	δm_3	=	0.050 [kg]
g	=	9.81 [kg/$\texttt{s}^2$]	τ_{dist}	=	[3.0,-4.0,1.0] [N·m]
f_v	=	$[0.05, 0.02, 0.01]$	f_d	=	$[0.10, 0.05, 0.03]$
k_p	=	200	γ	=	200
k_d	=	20	η	=	3
k_i	=	200	λ	=	100

The simulation goal is to track a constant joint angle value for each joint. Equation 5.10 gives the desired goal state.

$$
\begin{aligned}
q_d &= \left[\frac{2\pi}{3}, -\frac{3\pi}{4}, -\frac{\pi}{3}\right] \\
\dot{q}_d &= [0, 0, 0] \\
\ddot{q}_d &= [0, 0, 0]
\end{aligned}
\tag{5.10}
$$

The results of this simulation are seen in Figures 5.4 through 5.6. Due to a constant disturbance torque on each joint, the steady-state tracking error is significant. This tracking error is unacceptable and necessitates the use of a different controller.

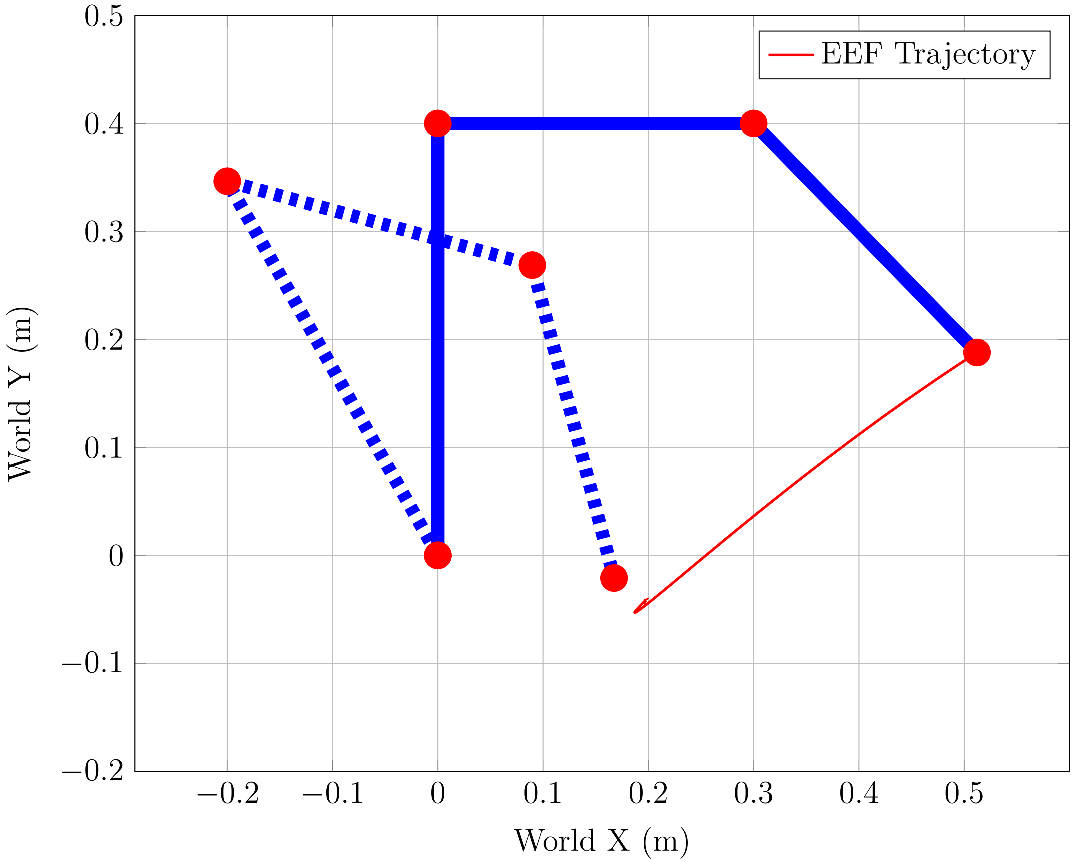


Figure 5.4: Computed Torque Method PD Setpoint Tracking (dashed lines represent desired robot setpoint) with significant disturbances

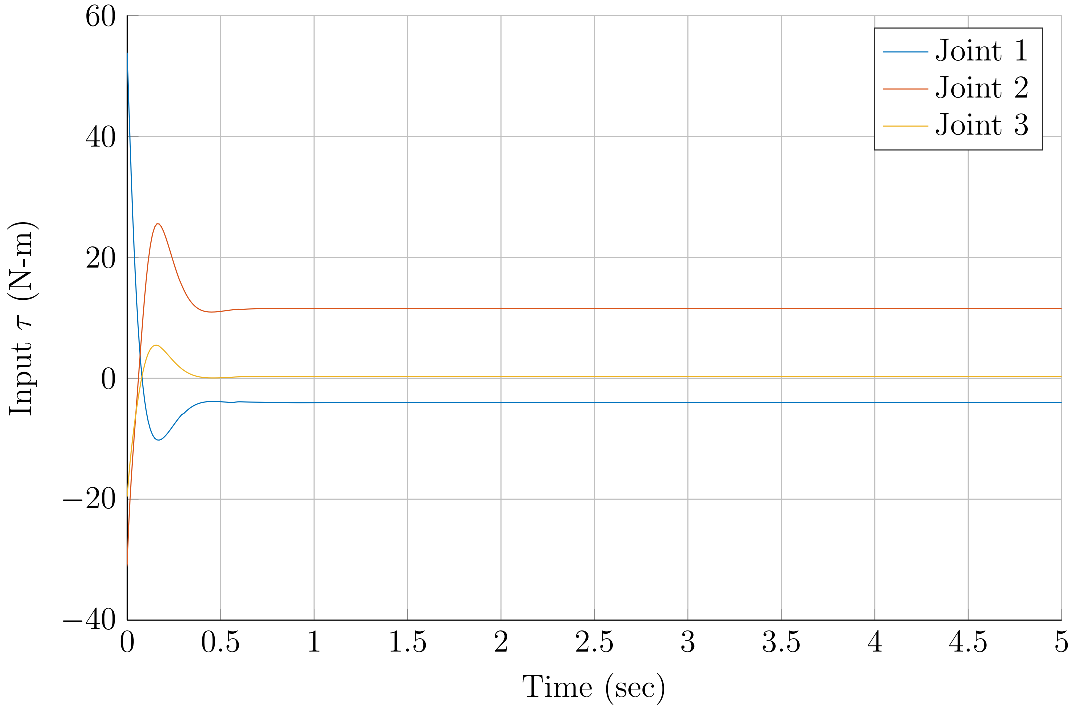
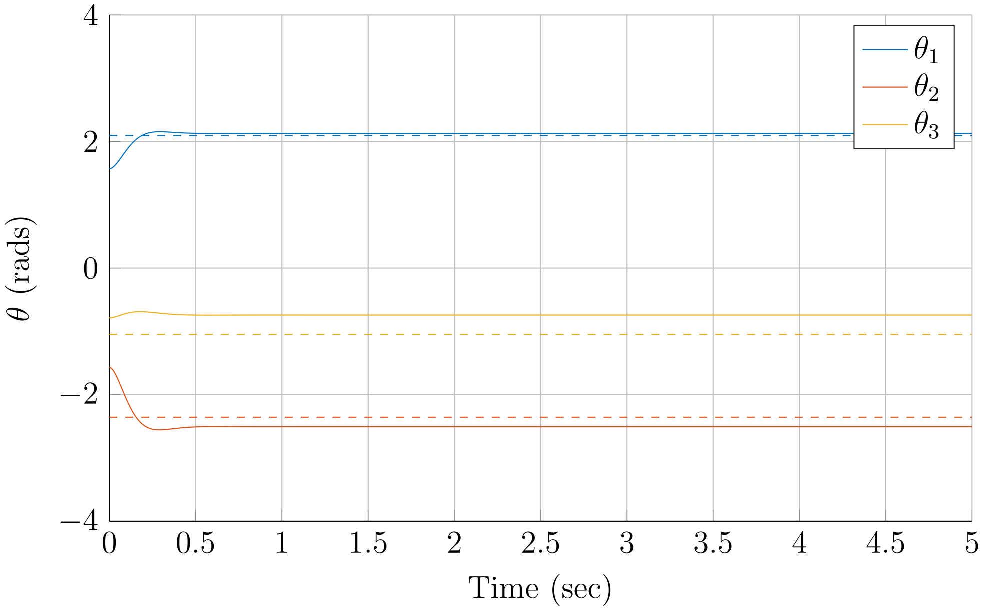


Figure 5.5: Computed Torque Method input torque



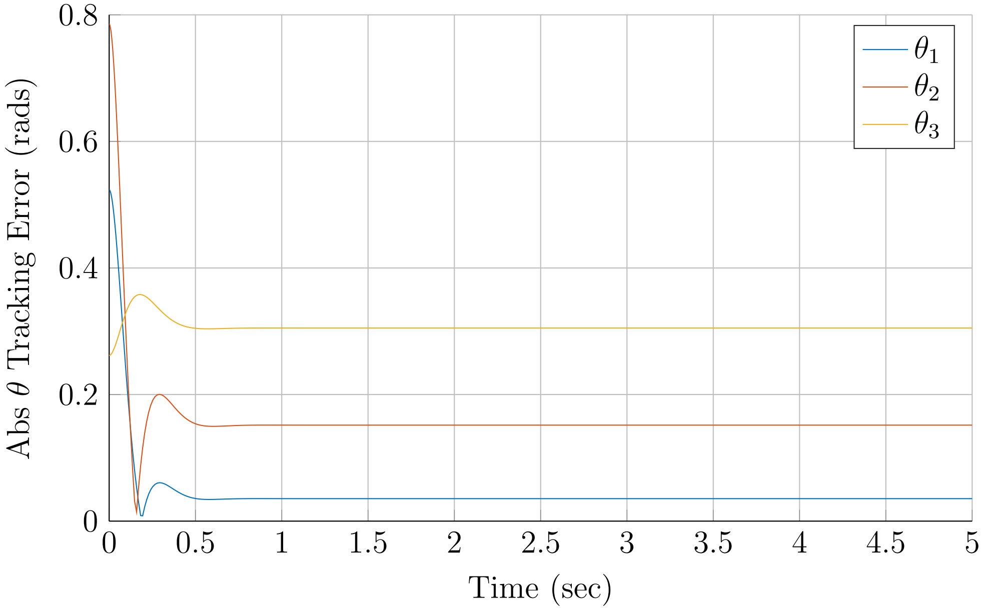


Figure 5.6: Computed Torque Method joint angles (above, dashed lines represent desired tracking joint angles) and joint angle tracking error (below)

5.3.2 PID Computed Torque Law

The addition of an integral term to the outer-loop PD control law compensates for steady-state tracking error. The new control law is given in equation 5.11.

$$\begin{aligned}
&\dot{\epsilon} = e \\
&v = -K_d\dot{e} - K_p e - K_i\epsilon \\
&\tau = M(q)(\ddot{q} + K_d\dot{e} + K_p e + K_i\epsilon) + V(q,\dot{q}) + G(q) + F_v(\dot{q}) + F_d(\dot{q}
\end{aligned} \tag{5.11}$$

In this equation ϵ is the integral of the tracking error e. This eliminates most of the steady-state tracking error seen in the previous section, even with constant high disturbance torques acting on each joint. The result of the same constant joint tracking simulation with a PID controller can be seen in Figures 5.7 and 5.9.

Almost all of the simulations in this report feature high starting torque values. This is due to the robot starting at rest, requiring a high amount of energy to reach the desired state based on controller gains. Higher controller gains will require higher input torques, but lower controller gains equate to suboptimal tracking. Adaptive PID controller tuning is discussed in Chapter 6.

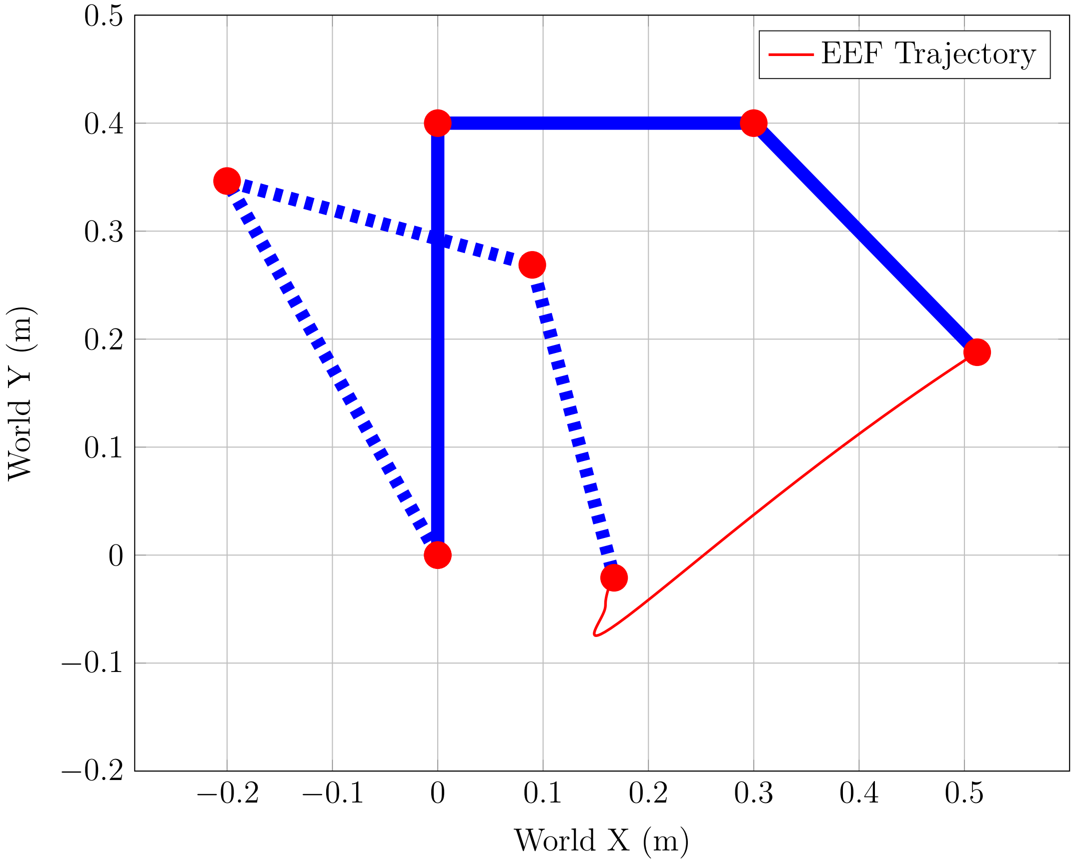


Figure 5.7: Computed Torque Method PID Setpoint Tracking (dashed lines represent desired robot setpoint) with significant disturbances

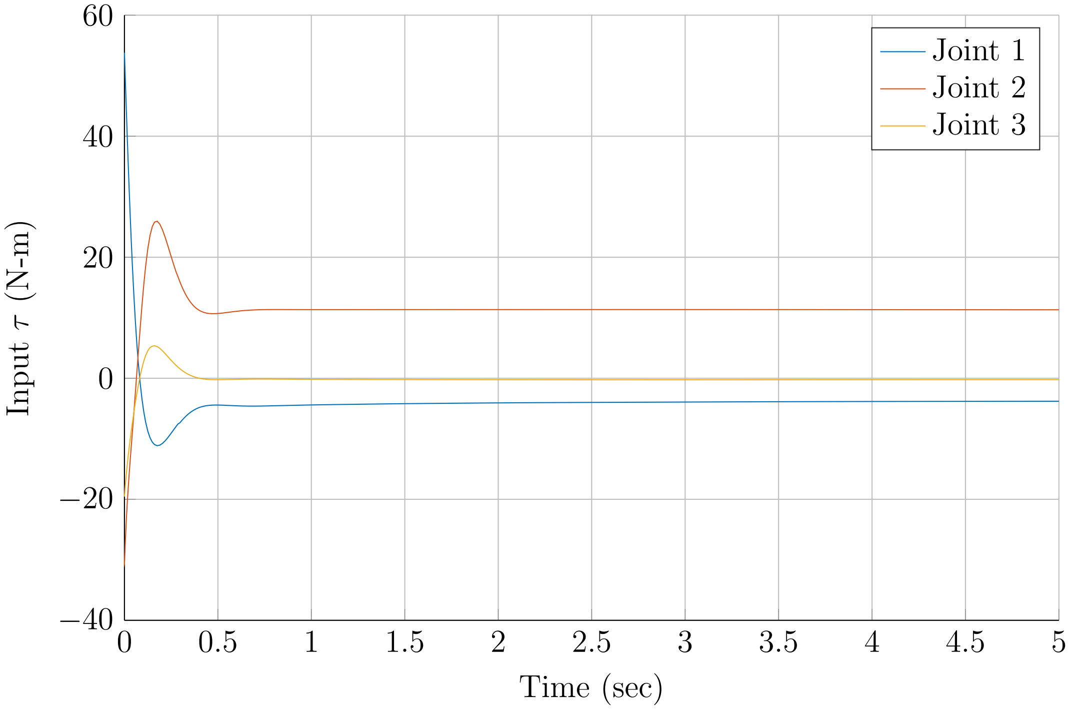


Figure 5.8: Computed Torque Method input torque

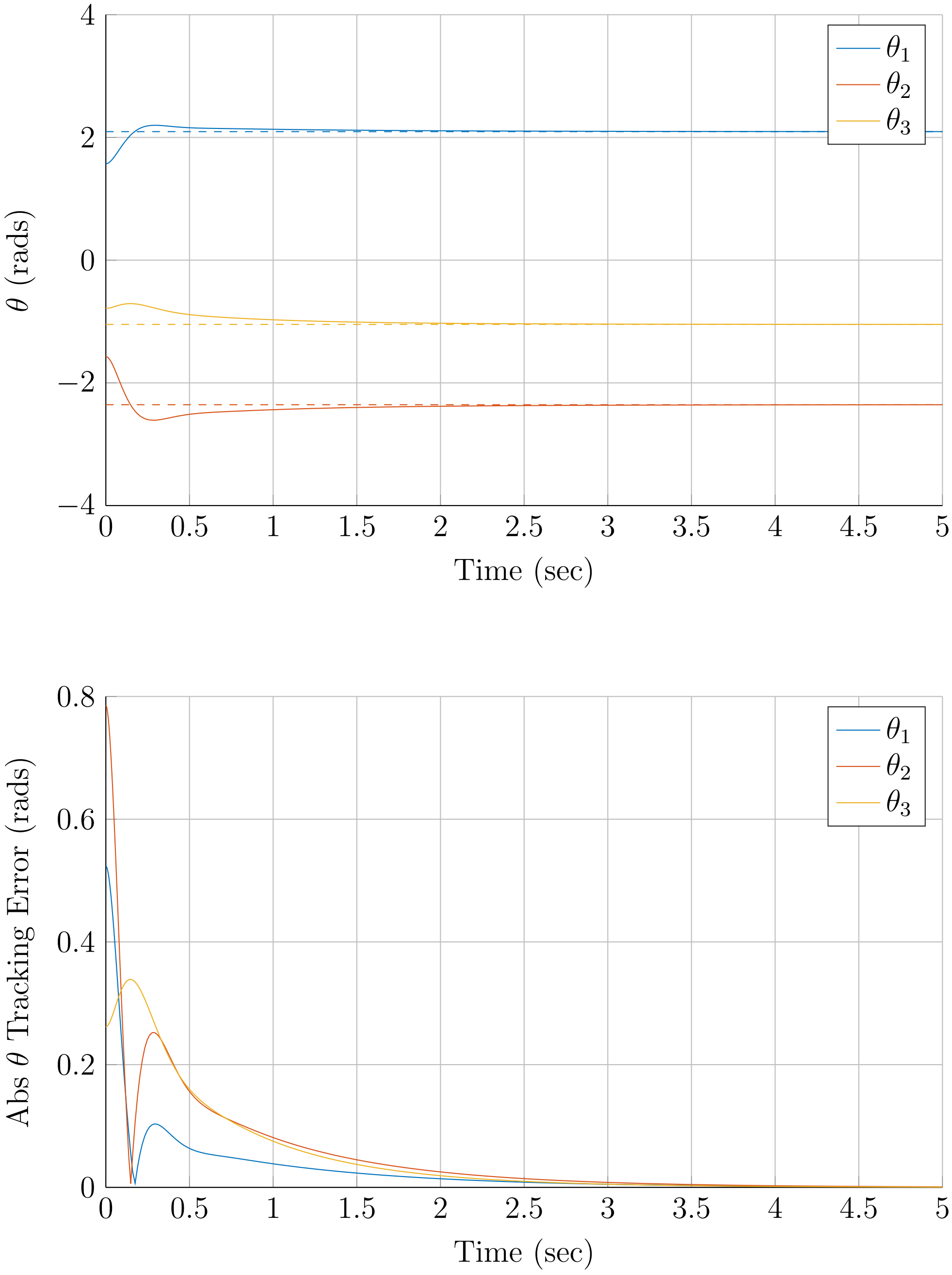


Figure 5.9: Computed Torque Method joint angles (above, dashed lines represent desired tracking joint angles) and joint angle tracking error (below)

5.3.3 Routh's Stability Criteria

The closed-loop stability of the outer-loop PID controller can be calculated using Routh's Stability Criteria [17]. The characteristic cubic polynomial of the PID controller is given by Equation 5.12.

$$s^3 I + K_d s^2 + K_p s + K_i \tag{5.12}$$

Where the K_d, K_p, K_i terms are diagonal matrices corresponding to derivative, proportional, and integral gains k_d, k_p, k_i respectively. The Routh array is given by Table 5.2.

Table 5.2: Routh array for PID controller

s^3	1	k_p
s^2	k_d	k_i
s^1	$\frac{k_d k_p - k_i}{k_d}$	
s^0	k_i	

So for all roots to have negative real parts, the condition 5.13 must be satisfied.

$$k_i < k_d k_p \tag{5.13}$$

As long as the integral gain is not too high relative to the other gains, then the closed-loop system is stable according to the Routh Criteria.

5.4 Sliding Mode Control

Although feedback linearization of this system through explicit modeling of the robot dynamics is a common and relatively trivial simplification for robotic systems, it is not without its drawbacks. The computed torque method of input-output linearization assumes that the system model is perfect. In reality, the robot dynamics are merely an estimate of the system. Unstructured parameter uncertainties and disturbances are not known beforehand and are therefore not accounted for. An alternative is to use a more robust linearization process that overcomes uncertainties and disturbances while eliminating nonlinearities.

Sliding Mode Control (SMC) is a nonlinear control method that places a priority on robustness. This can be applied to input-output linearization in the form of Sliding Mode Linearization (SML) as described by [4]. In this paper, Fernandez *et al.* describe a linearization of both Single Input Single Output (SISO) systems and Multi Input Multi Output (MIMO) systems by implementing SML. A linear controller can then be used to make the system track a user-defined trajectory with bounded unstructured uncertainties and disturbances. This is done by attracting the system to a defined sliding surface s in the phase plane by using high gain feedback. Fernandez *et al.* claims that sliding mode guarantees system attraction to the surface provided that the disturbances are bounded and matching conditions are satisfied [4]. Another advantage of sliding mode control is the system is guaranteed to reach stability at a desired equilibrium point in a finite time.

The general nonlinear system equation is written again in 5.14.

$$\begin{aligned} \dot{x} &= f(x) + g(x)u + d \\ y &= h(x) \end{aligned} \tag{5.14}$$

The reference y_d is the desired tracking signal for the output. The sliding surface s is defined by equation 5.15.

$$
\begin{aligned}
s &= \sum_{k=0}^{r-1} \beta_k \frac{d^k e}{dt^k} \\
e &= y_d - y
\end{aligned}
\tag{5.15}
$$

The sliding surface s requires that the following condition in equation 5.16 is met [4].

$$\lim_{s \to 0} s \cdot \dot{s} < 0 \tag{5.16}$$

By differentiating 5.15 and satisfying the condition 5.16, the time derivative of the sliding surface s is chosen.

$$\dot{s} = -\eta sign(s) \tag{5.17}$$

5.4.1 Sliding Mode PID Controller

As with the feedback linearization method described previously, the output y is differentiated with respect to time until the input u is seen in the equations. This defines the relative order r of the system. By combining equations 5.11, 5.15, and 5.17, a control law for the input u can be defined by modifying the existing computed torque control law. This is seen in equation 5.18.

$$
\begin{aligned}
\dot{\epsilon} =& e \\
v =& - K_d \dot{e} - K_p e - K_i \epsilon \\
s =& \dot{e} + \lambda e \\
\tau =& M(q)(\ddot{q} + K_d \dot{e} + K_p e + K_i \epsilon) + \dots \\
& \quad V(q, \dot{q}) + G(q) + F_v(\dot{q}) + F_d(\dot{q}) + \tau_{dist} + \eta sign(s)
\end{aligned}
\tag{5.18}
$$

$$sign(s) = erf(\gamma s) \tag{5.19}$$

The addition of the sliding function to the existing control law ensures that the controller will push the system tendency towards the sliding surface to track the desired reference value. The sliding mode controller used in the following simulations follows the block diagram shown in Figure 5.10.

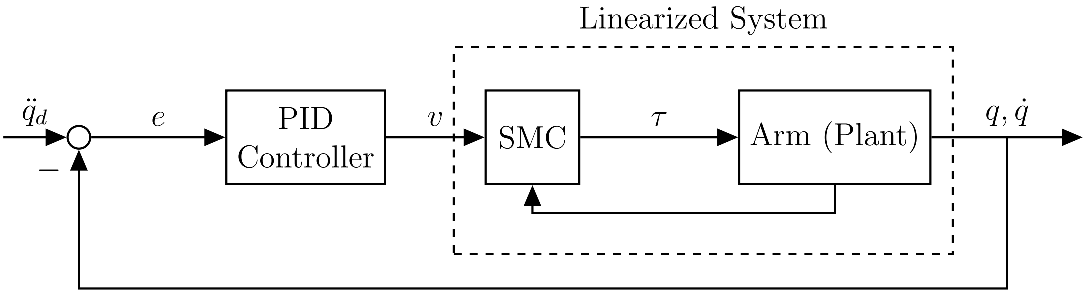


Figure 5.10: Block diagram of SMC control model [4]

The error function in equation 5.19 is used instead of an actual `signum` function. This was done in order to minimize computation time, as the error function is continuously differentiable. The γ value dictates the thickness of the boundary layer of the sliding surface. The same constant joint tracking simulation was completed with a PID sliding mode controller with the same parameters and disturbances. In addition, $\gamma = 200, \lambda = 100, \eta = 5$. The selection of the PID gains and the values of λ and η determine the behavior of the sliding mode controller. The results of the simulation can be seen in Figures 5.11 through 5.13.

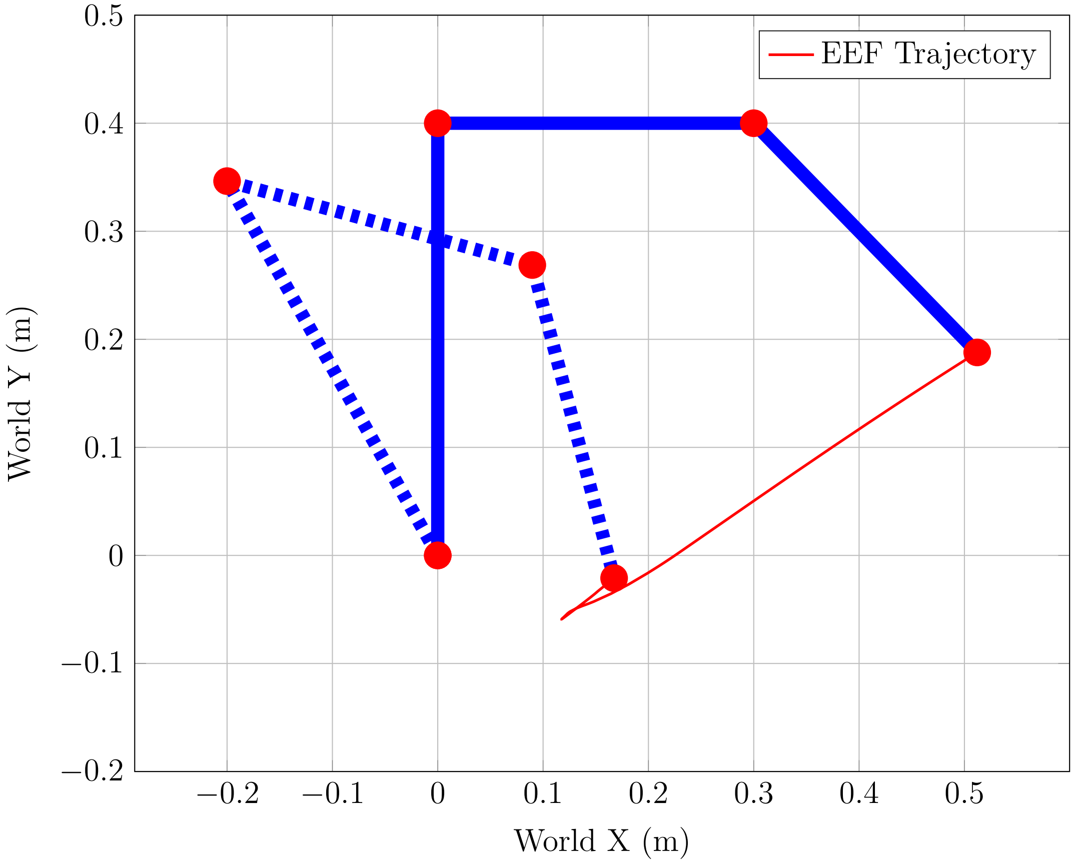


Figure 5.11: Sliding Mode PID Setpoint Tracking (dashed lines represent desired robot setpoint) with significant disturbances

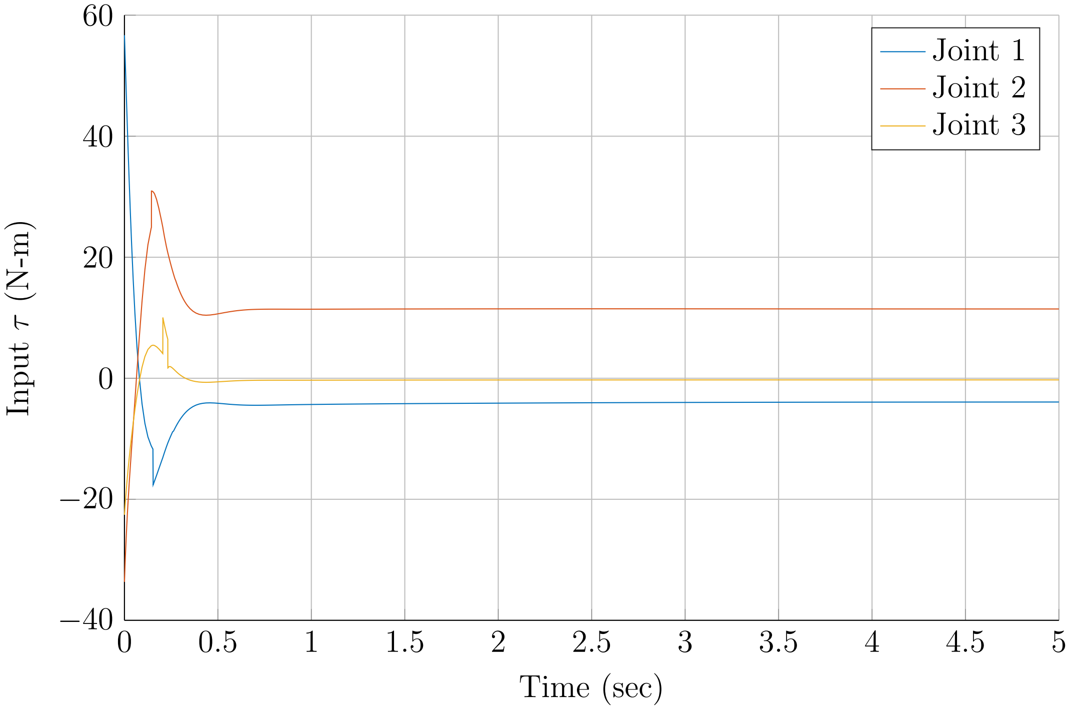


Figure 5.12: SMC input torque

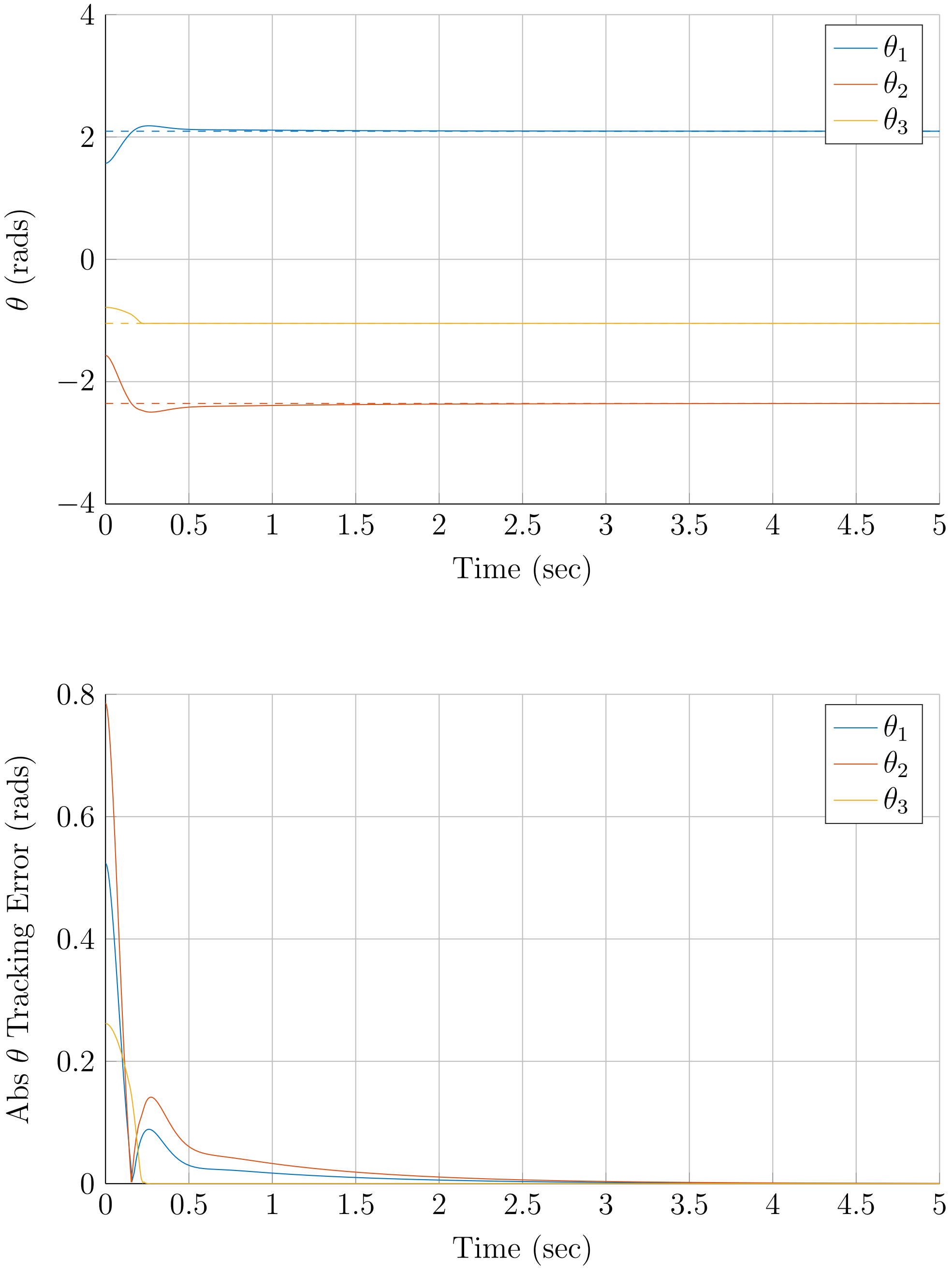


Figure 5.13: SMC joint angles (above, dashed lines represent desired tracking joint angles) and joint angle tracking error (below)

5.4.2 Lyapunov Stability

To prove the stability of the controller in the presence of disturbances and uncertainties, Lyapunov Stability Theory is used. The Lyapunov function for this system is defined by equation 5.20. According to [51], this equation is valid given the skew-symmetry of $\dot{M} = -2V$, which can also be written as $\dot{M} = V + V^T$.

$$V(t) = \frac{1}{2}[s^T M s] \tag{5.20}$$

Differentiating

$$\dot{V}(t) = s^T(M\ddot{q} - M\ddot{q}_d) + \frac{1}{2}s^T\dot{M}s \tag{5.21}$$

Given skew-symmetry, and substituting $M\ddot{q}$, and N representing other nonlinear terms:

$$\begin{gathered} N(\dot{q}, q) = V - G - F_v - F_d \\ \dot{V} = s^T(\tau - M\ddot{q}_d - N) \end{gathered} \tag{5.22}$$

If the control input τ is defined by equation 5.18, the derivative of the Lyapunov function becomes

$$\dot{V} = S^T(\tau_{dist}) - \sum_1^n \eta_i |s_i| \tag{5.23}$$

As long as the components of the η vector are greater than the bounded disturbance forces and modeling errors, the sliding condition is satisfied and the sliding surface $s = 0$ can be reached in a finite time [51]. The trajectory tends toward q_d exponentially.

5.5 Cartesian to Joint Space Mapping

Both the PID Sliding Mode and Computed Torque controllers work by tracking a desired joint angle trajectory. A convenient method of converting a desired Cartesian trajectory command to a desired joint trajectory command was introduced in Chapter 2. Recall that the Jacobian is a linear mapping of joint velocities in joint space to Cartesian space. This relationship is repeated here in equations 5.24 and 5.25.

$$^{o}v = {}^{o}J(\theta)\dot{\theta} \tag{5.24}$$

$$\dot{\theta} = J^{-1}(\theta)v \tag{5.25}$$

The Jacobian matrix for a spatial manipulator is only invertible for a 6-DOF robot. If $n \neq m$[3], the pseudo inverse of the Jacobian must be used for all other cases (sometimes referred to as the Moore-Penrose pseudoinverse) [14].

5.6 Results

To understand the effect of both the sliding mode and computed torque controllers, two sets of simulations were conducted. The first set of simulations used both controller types to track a circular path with and without disturbances. The second set of simulations added a force control element to simulate a perpendicular force on the EEF. This would simulate the force profile felt on the EEF of the robot during a punch. This simulation was done using both controller

[3] This occurs when the number of inputs differs from the number of outputs

types with and without disturbances. Both controllers used an outer loop PID controller with parameters listed in Table 5.1.

5.6.1 Circular trajectory tracking simulation

The goal for this set of simulations was to track a desired trajectory with the EEF. Both the computed torque and the sliding mode controllers were used to track the perimeter of a circle with radius 0.1 meters. The robot EEF tracks the circle for two revolutions, with the starting and stopping poses equal to the initial pose of the robot (q_0 as specified in Table 5.1).

5.6.1.1 Computed Torque Method Results

The first set of simulations was conducted without any disturbances acting on the system. The results can be seen in Figures 5.14 through 5.16.

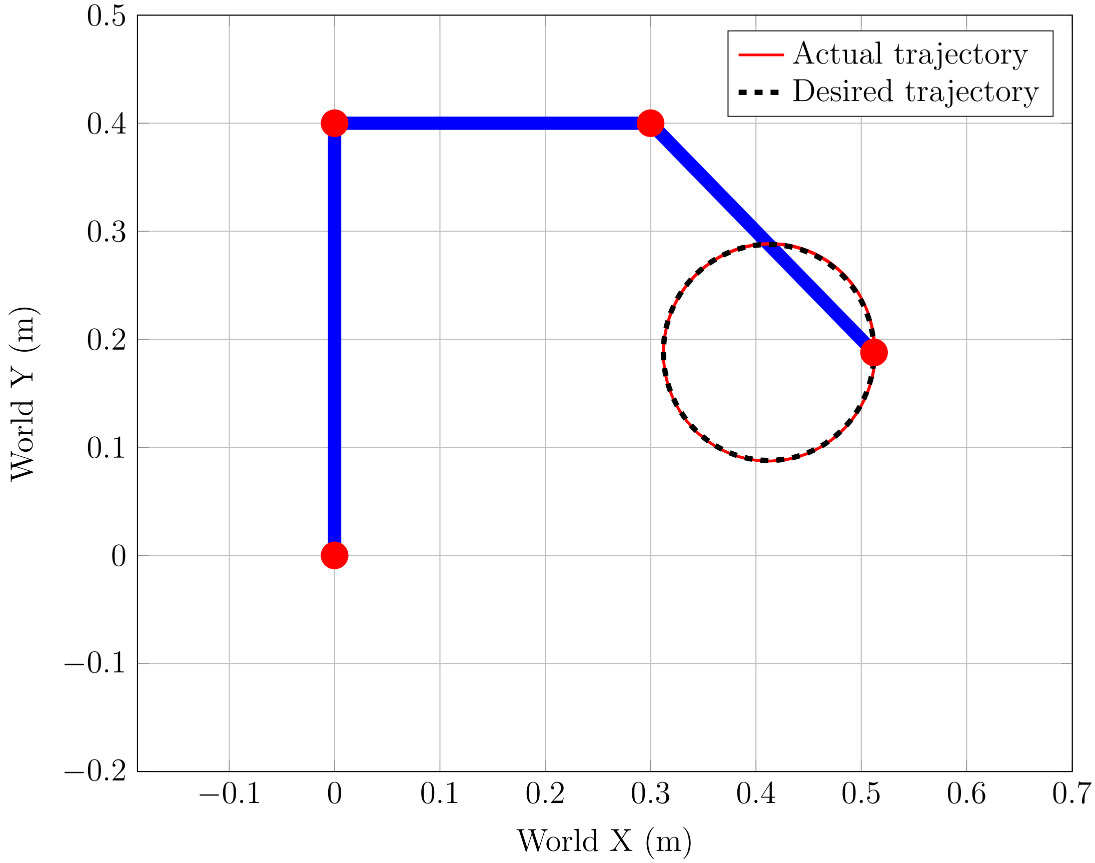


Figure 5.14: Computed Torque Method circle tracking simulation (with no disturbances)

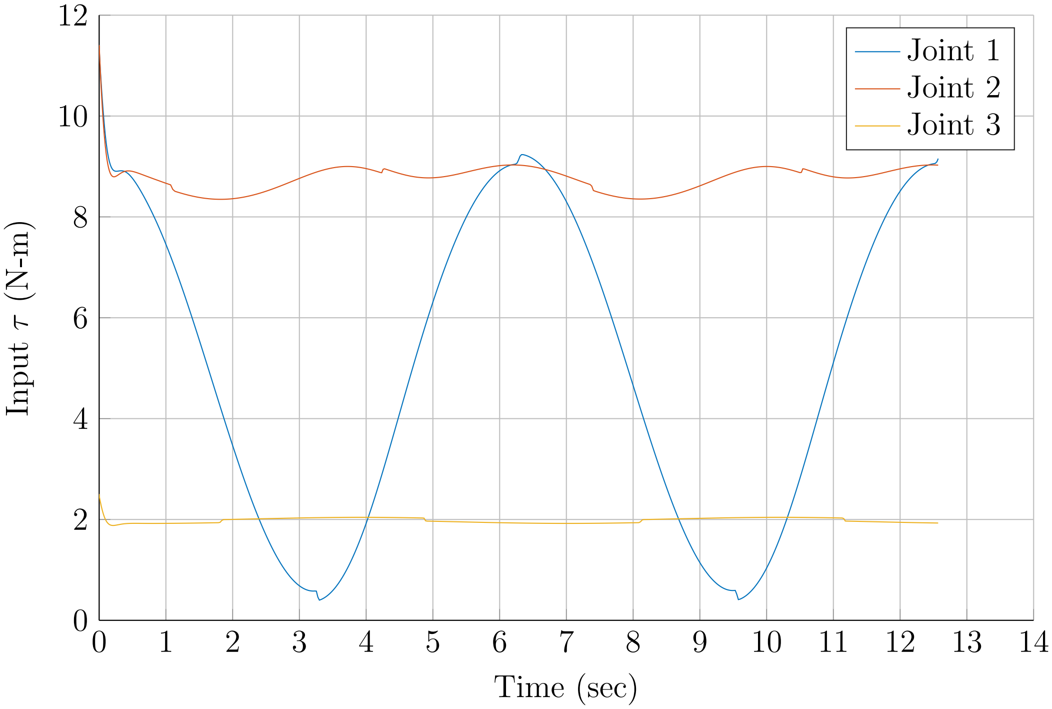


Figure 5.15: Computed Torque Method input torque

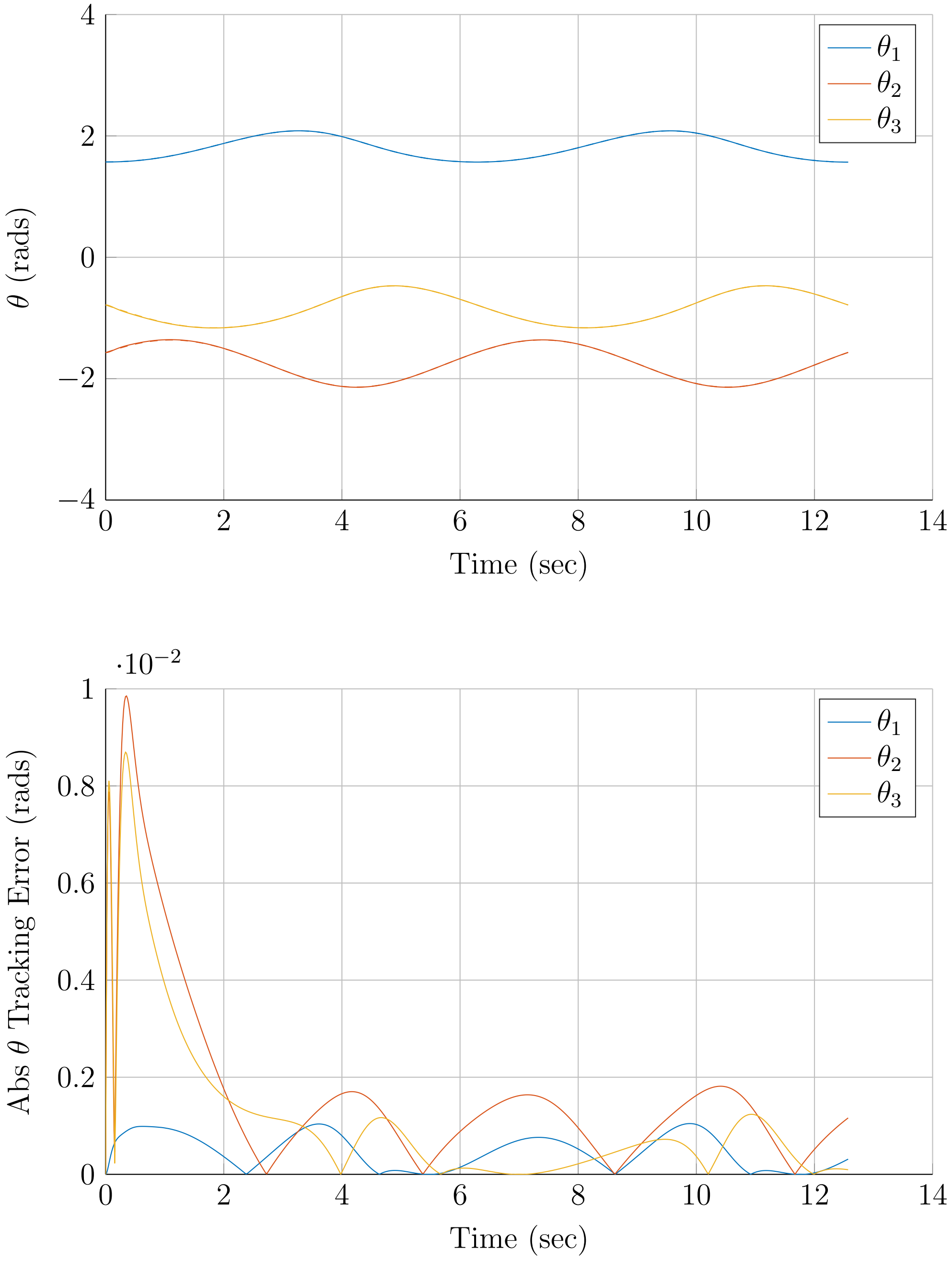


Figure 5.16: Computed Torque Method joint angles (above) and joint angle tracking error (below)

Figures 5.17 through 5.19 give the results of the simulation using a computed torque controller with significant disturbance torques. The disturbance torque values are given in Table 5.1.

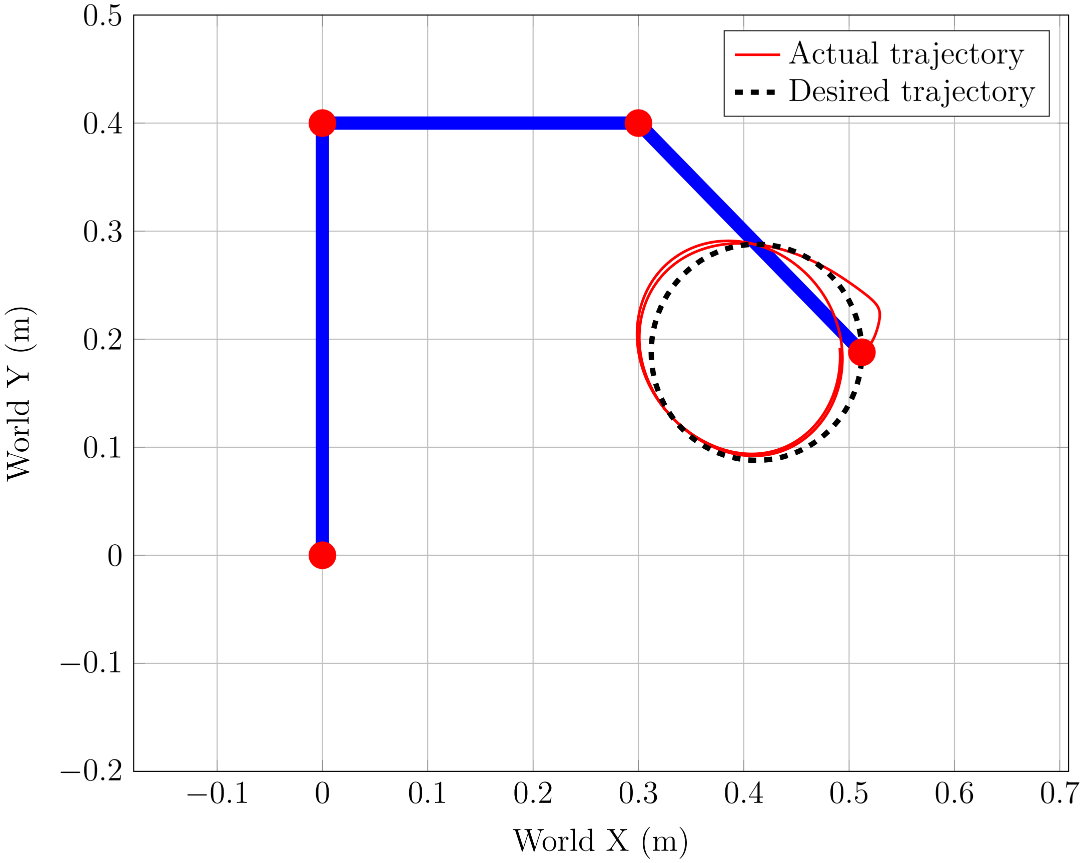


Figure 5.17: Computed Torque Method circle tracking simulation (with significant disturbances)

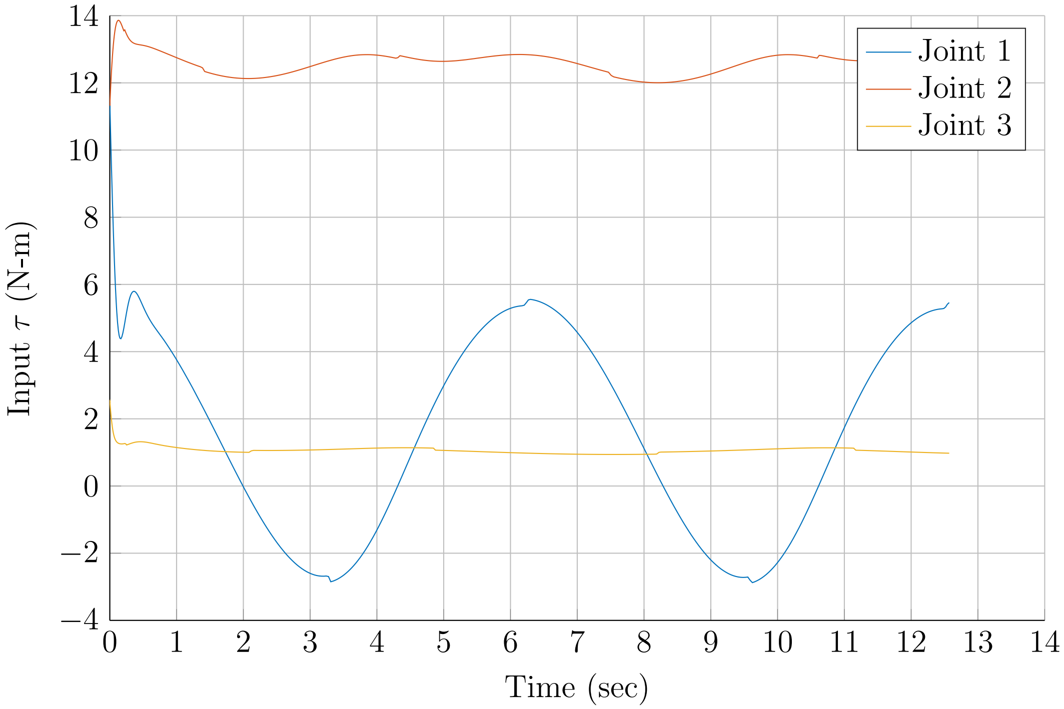


Figure 5.18: Computed Torque Method input torque

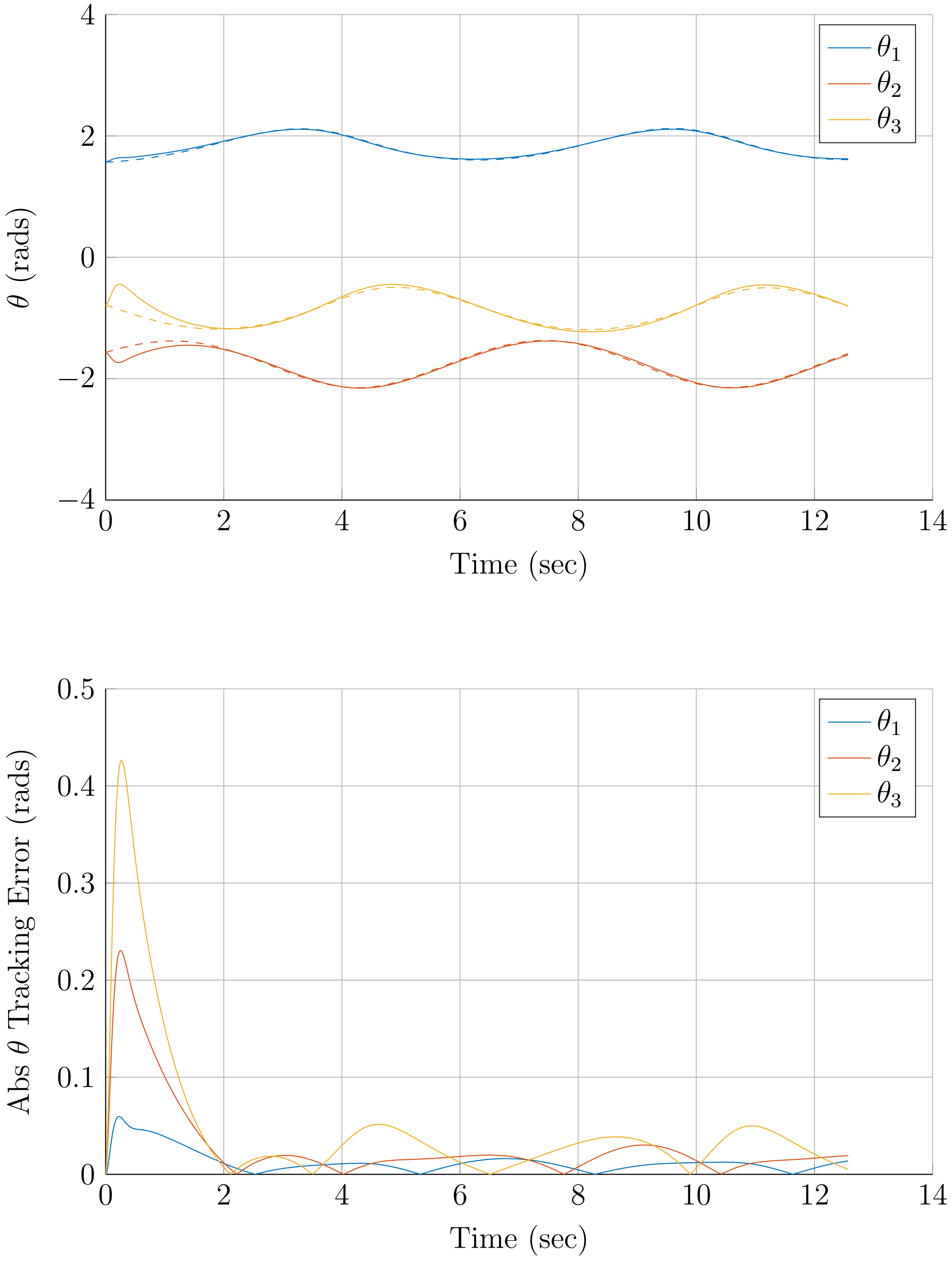


Figure 5.19: Computed Torque Method joint angles (above) and joint angle tracking error (below)

5.6.1.2 Sliding Mode Control Results

Figures 5.20 through 5.22 give the results of the simulation using a sliding mode controller with significant disturbance torques. The disturbance torque values are given in Table 5.1. This controller allows the robot to track the reference value with minimal error despite significant disturbances.

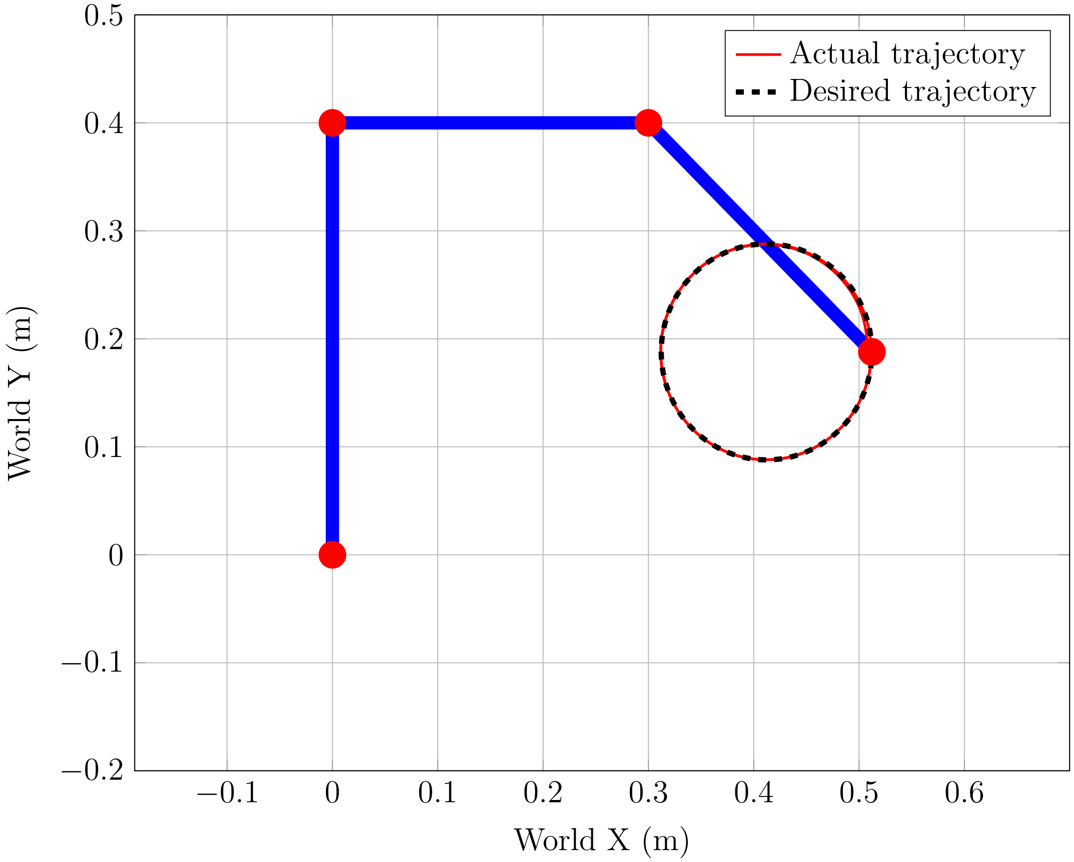


Figure 5.20: SMC circle tracking simulation (with significant disturbances)

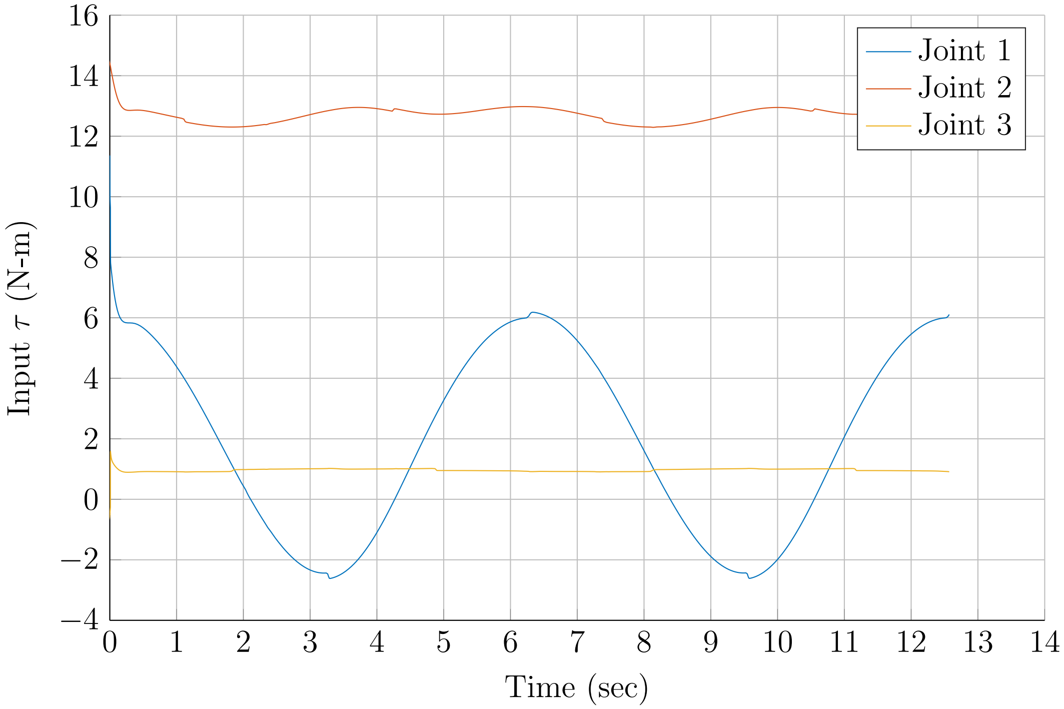


Figure 5.21: SMC input torque

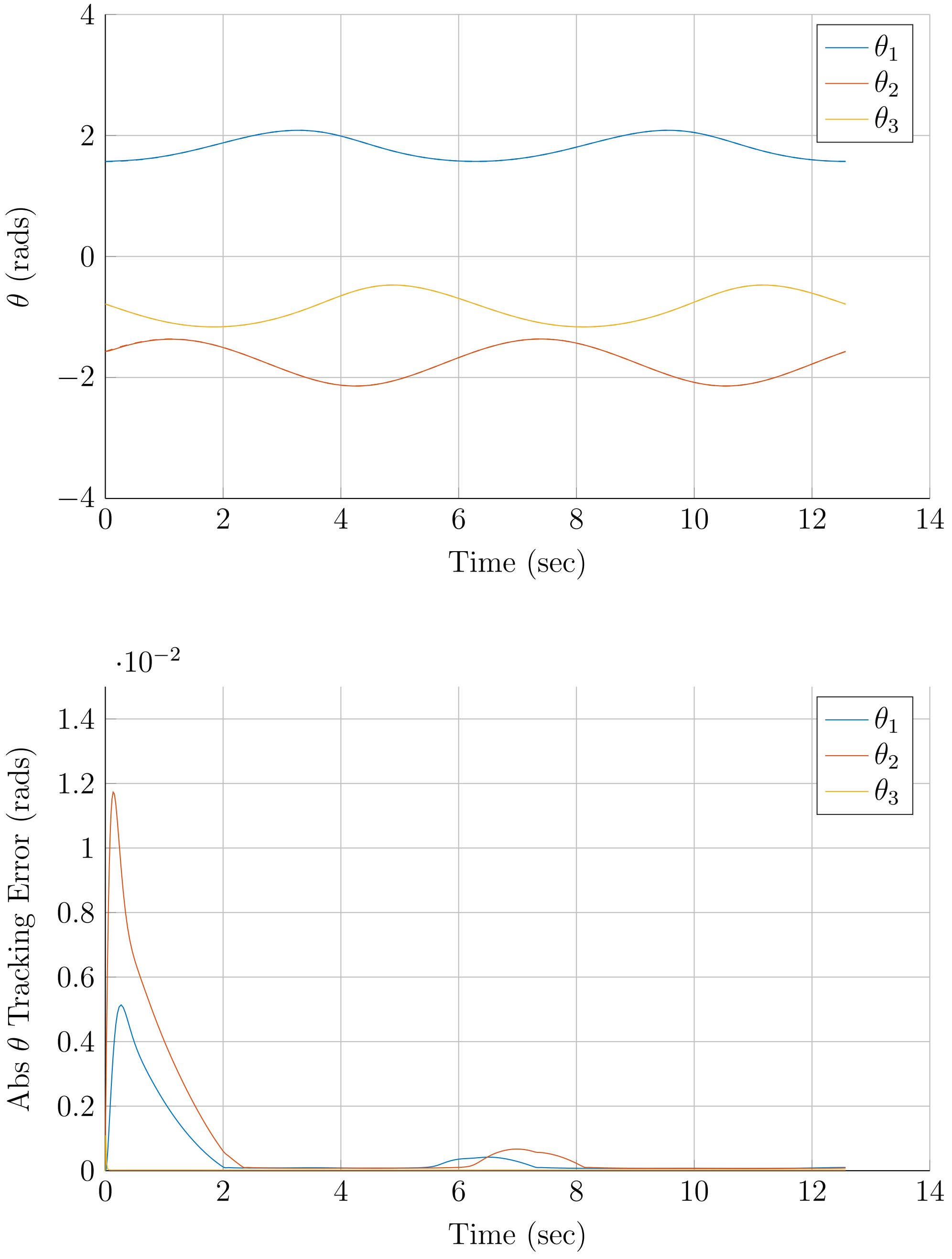


Figure 5.22: SMC joint angles (above) and joint angle tracking error (below)

It was stated previously that the sliding surface s was defined by equation

5.19. The error function changes in stiffness as the γ value increases. The γ value can be changed to a much smaller number to increase the boundary layer of the sliding surface. Figures 5.23 through 5.25 illustrate the same simulation with the sliding mode controller but with a γ value of 0.2.

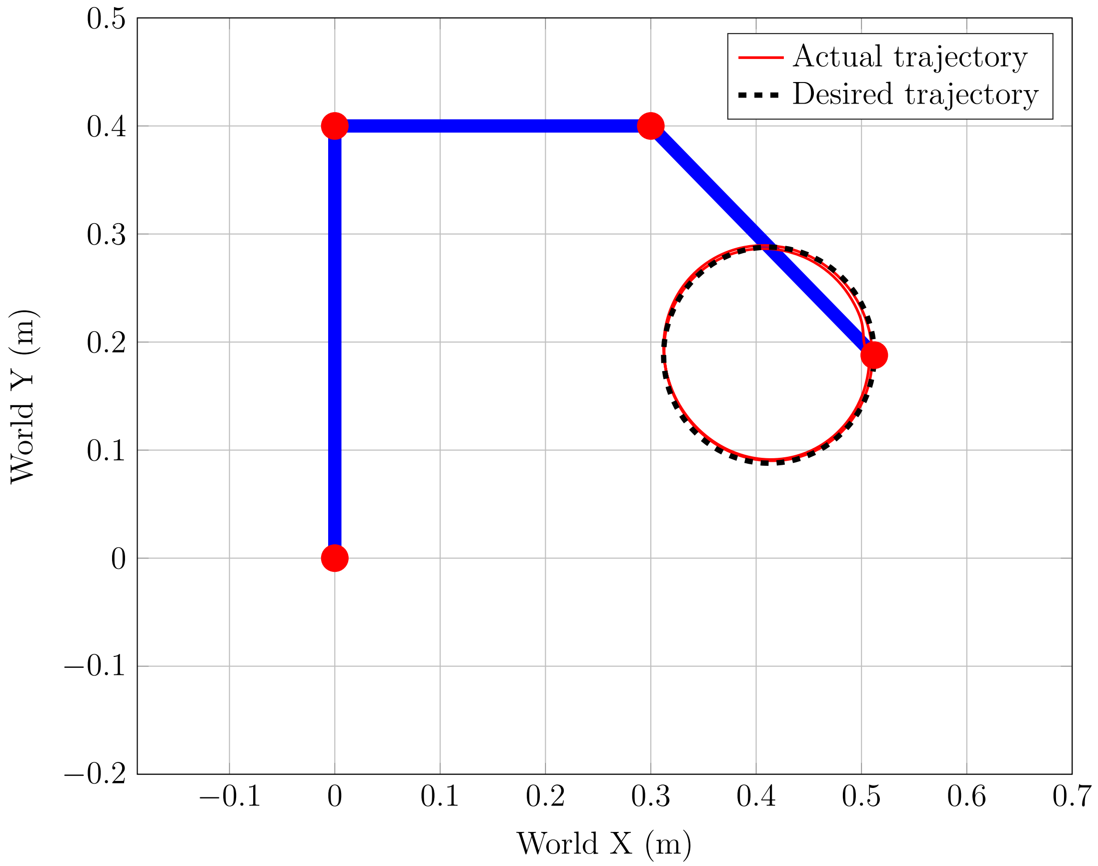


Figure 5.23: SMC circle tracking simulation (with significant disturbances and low γ value)

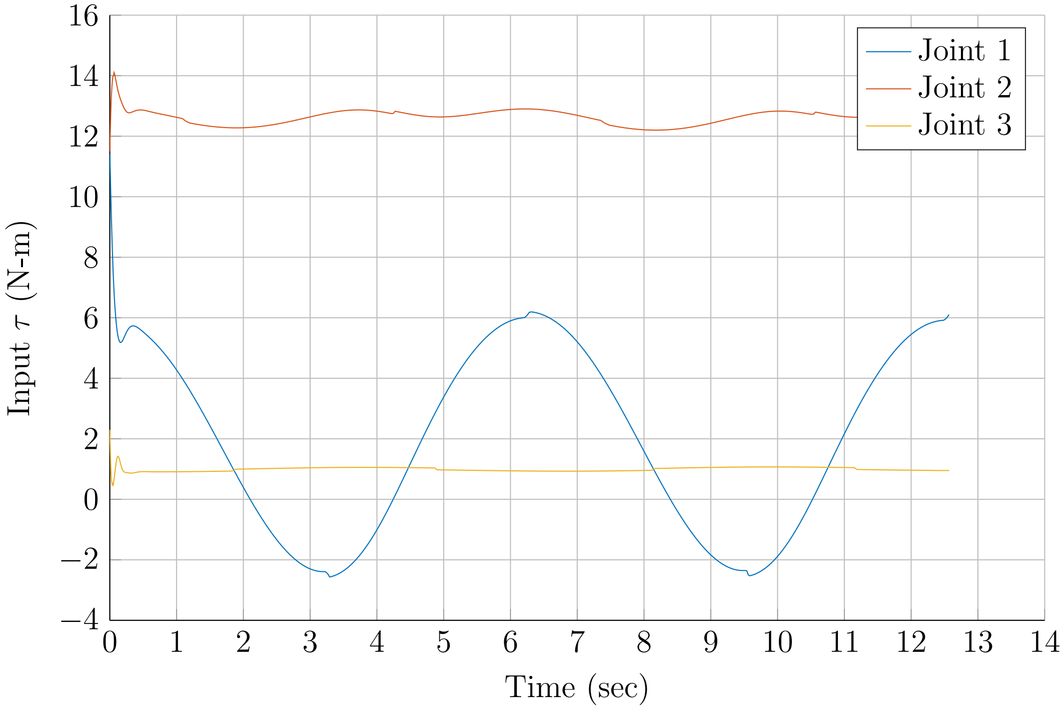


Figure 5.24: SMC input torque

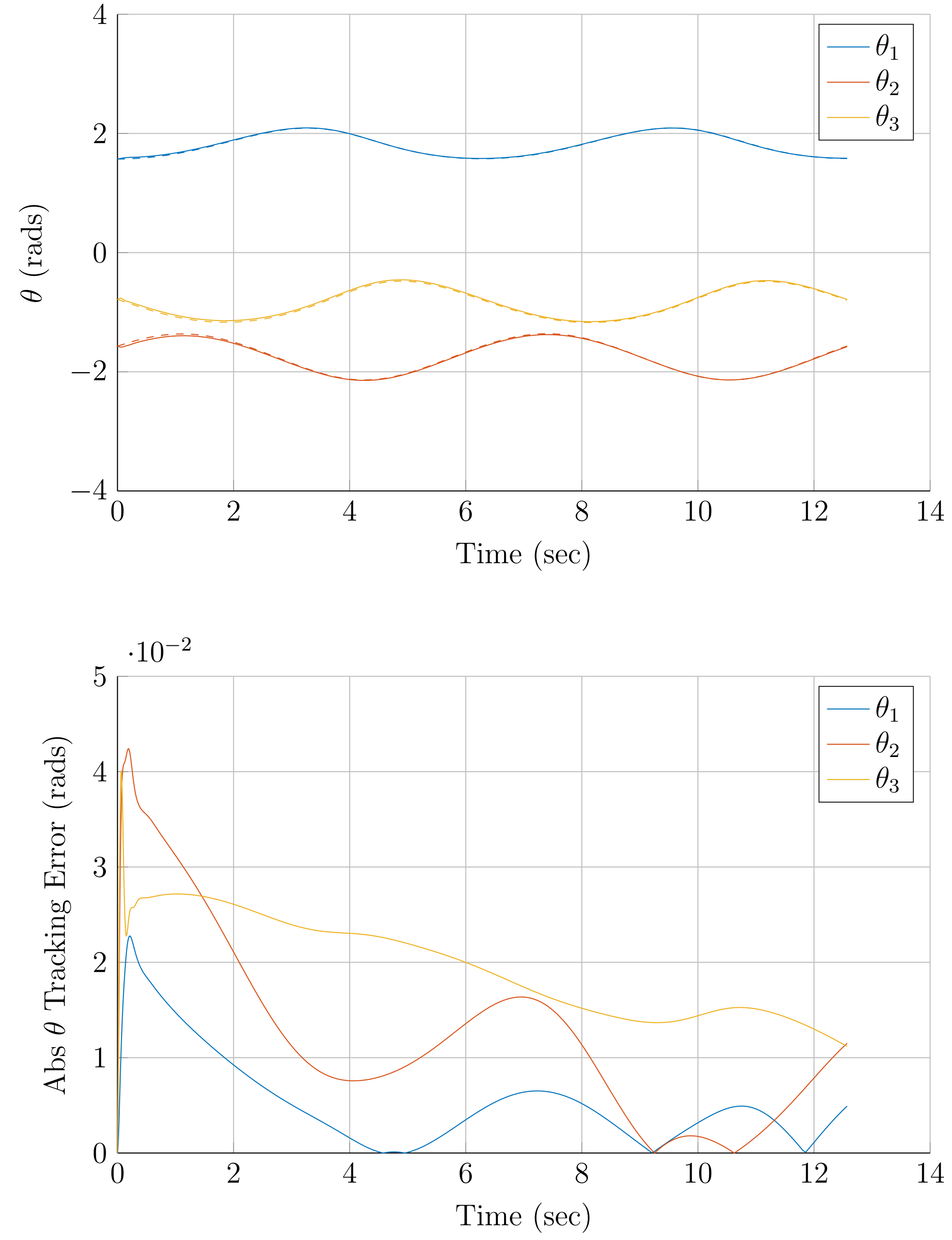


Figure 5.25: SMC joint angles (above) and joint angle tracking error (below)

5.6.2 Punch Force Control Simulation

This simulation set involves force control and tracking. In order to simulate the effect the punch has on the EEF of the robot arm, an approximate force profile was created that would act perpendicular to the EEF of the robot, mimicking a punch force. In reality, the punch would also exert a torque on the EEF due to the moment arm offset from punch location to EEF. However, this is discounted in the simulation as inducing a torque compliance would bend the surface of the object, resulting in a poor shear profile. Because the goal is for the object to track the punching tool, only the X and Y forces in the world frame are accounted for. A Cartesian EEF outer-loop control law dictates how the EEF will react during a punch. The force profile is given in the equation below and in Figure 5.26. This force profile is modeled as a square instantaneous force for two seconds. In reality, the force on the EEF would be gradually induced. However, the instantaneous force is modeled here as a worst-case scenario.

$$F(t) = \begin{cases} 20, & \text{if } 4 \leq t \leq 6 \\ 0, & \text{otherwise} \end{cases}$$

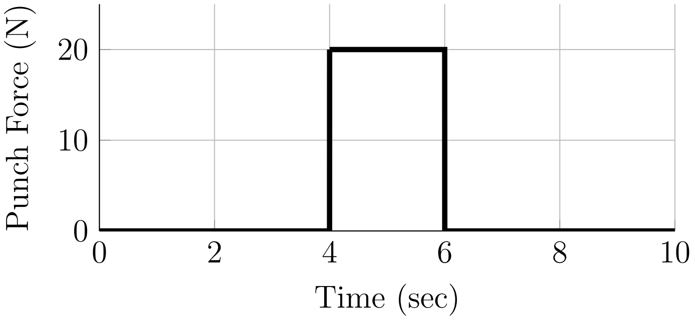


Figure 5.26: Punch force magnitude acting on EEF during simulation

The EEF commanded velocity to send to the controller is given by equation 5.26. The velocity command is dictated by a spring force that acts in opposition to the force on the EEF.

$$\begin{aligned} e_{eef} &= y - y_0 \\ \dot{y}_d &= K_f F + K_e e_{eef} \end{aligned} \tag{5.26}$$

The further the EEF distance is from its initial location, the greater its restoring force magnitude will be. The maximum distance the EEF will travel can be calculated by setting the commanded EEF velocity to 0. In this simulation, $K_f = \frac{1}{100}$, and $K_e = 2$.

$$|y_{max}| = \frac{K_f F}{K_e} \tag{5.27}$$

This ensures that the EEF will return to its initial state after the punch, while also exhibiting compliance during the punch to minimize shear forces on the object.

5.6.2.1 Computed Torque Method Results

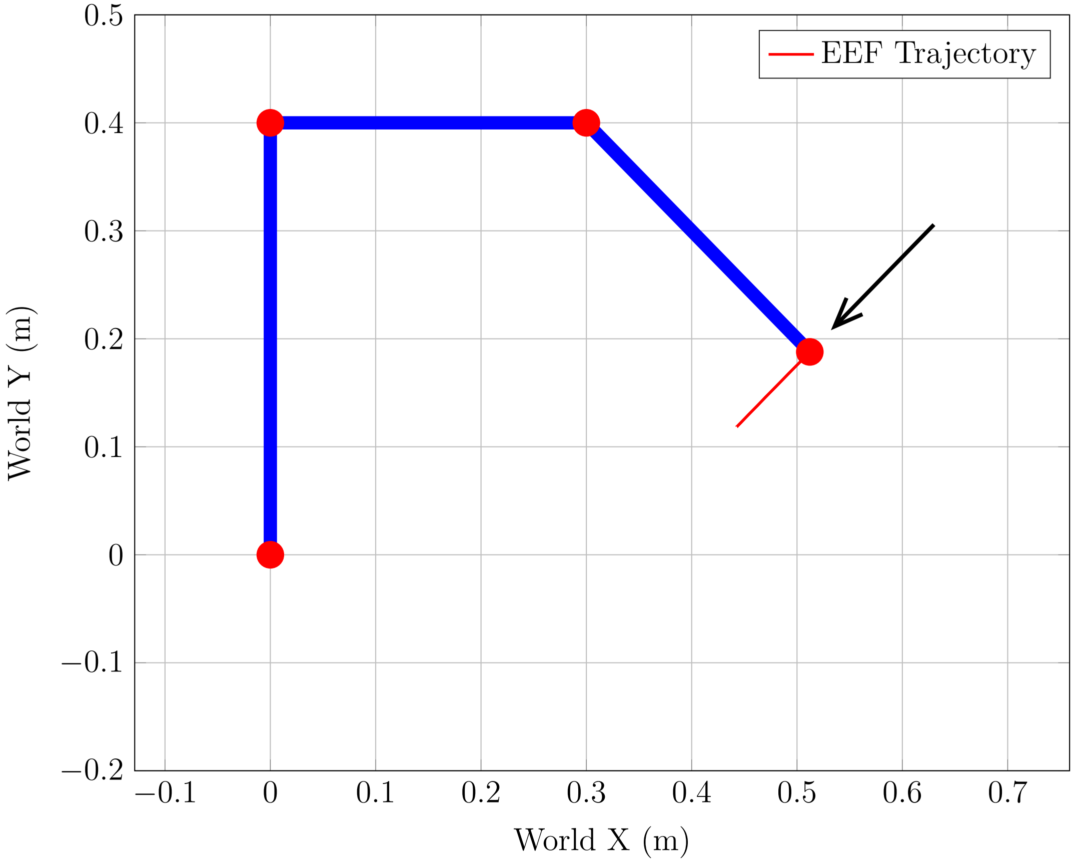


Figure 5.27: Computed Torque Method punch simulation (no disturbances). Arrow represents punch force direction

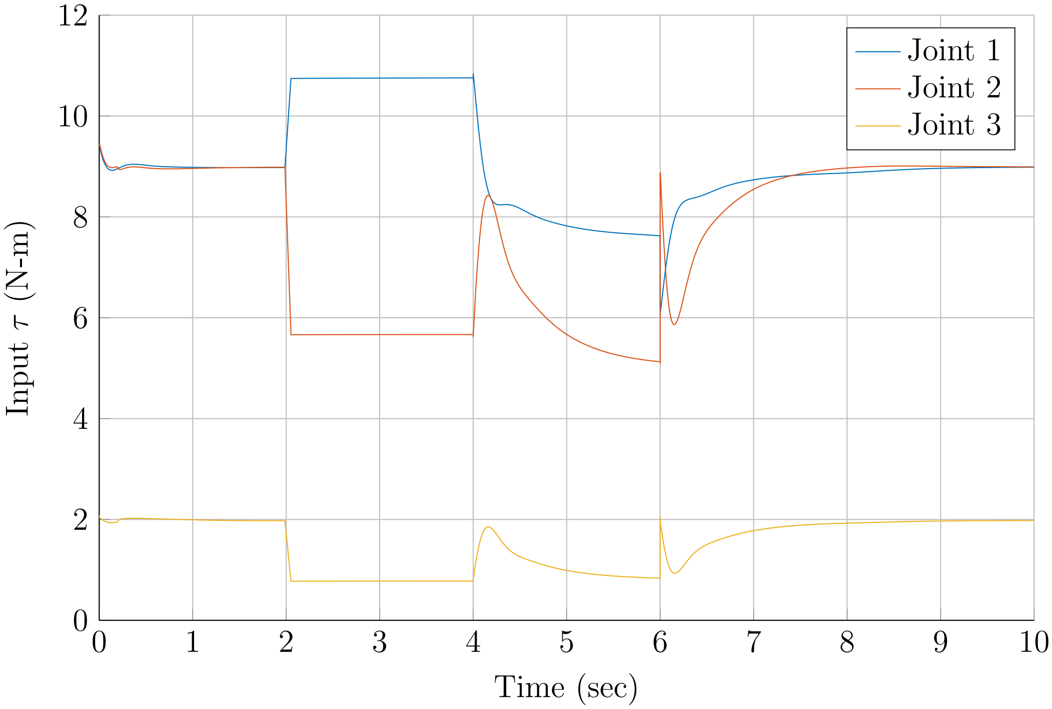


Figure 5.28: Computed Torque Method input torque

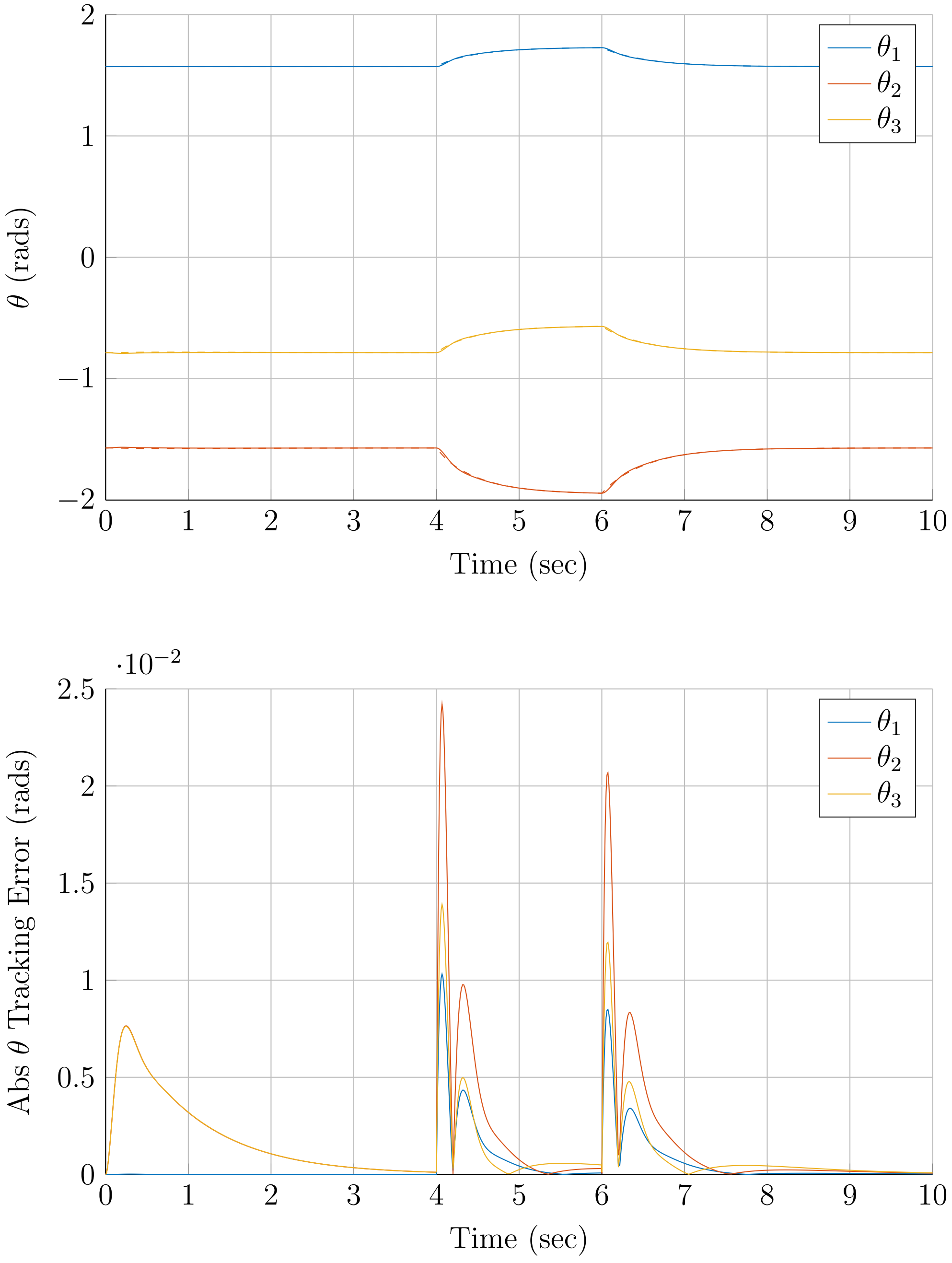


Figure 5.29: Computed Torque Method joint angles (above) and joint angle tracking error (below)

Figures 5.30 through 5.32 show the performance of the same computed torque control law during a punch with constant significant disturbance torques. The disturbance torque values are given in Table 5.1. The effect of adding unknown disturbances significantly change the expected trajectory.

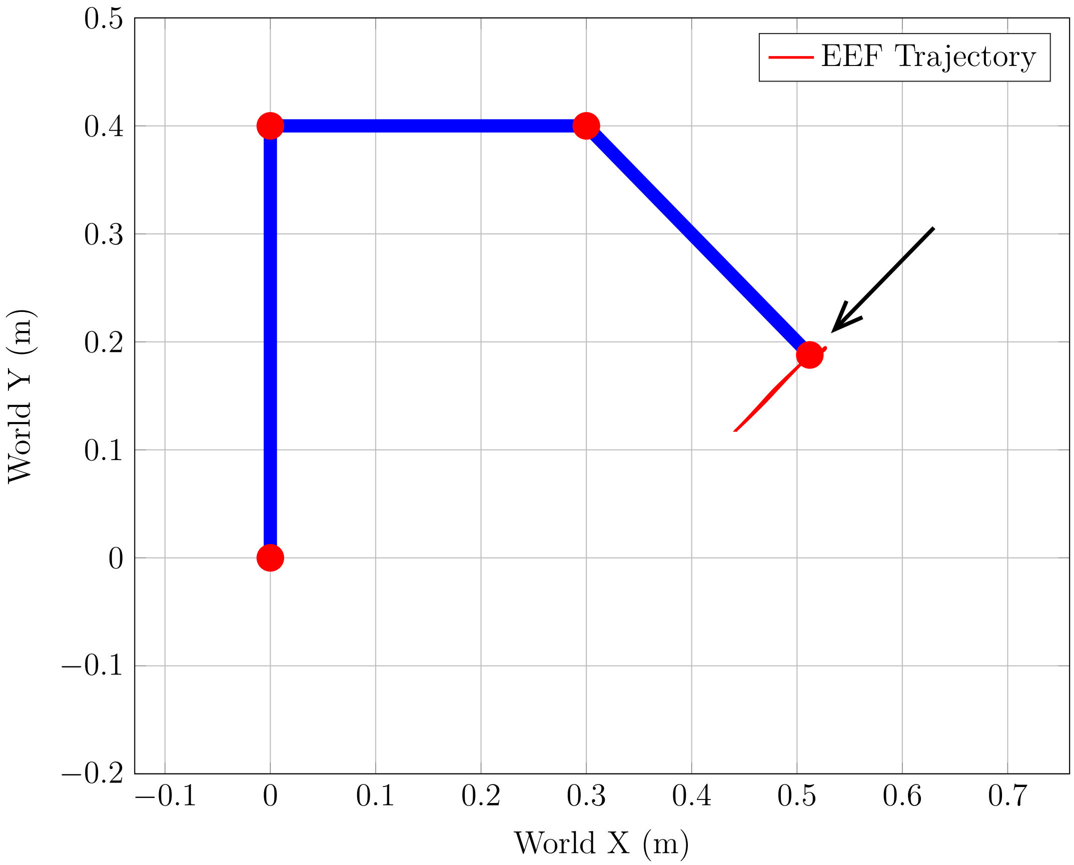


Figure 5.30: Computed Torque Method punch simulation (with significant disturbances). Arrow represents punch force direction

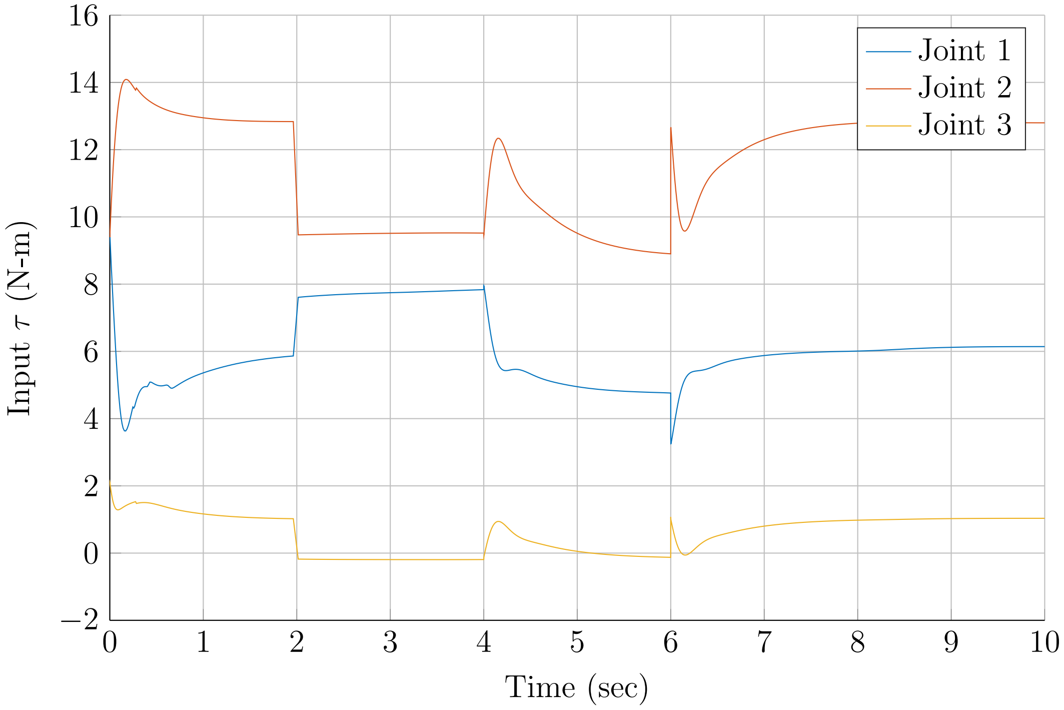
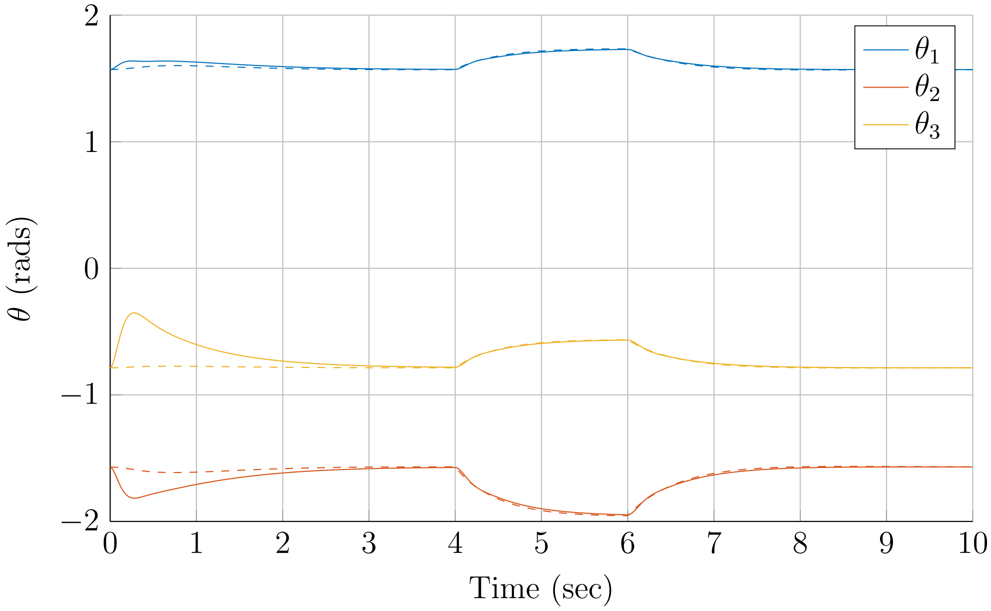


Figure 5.31: Computed Torque Method input torque



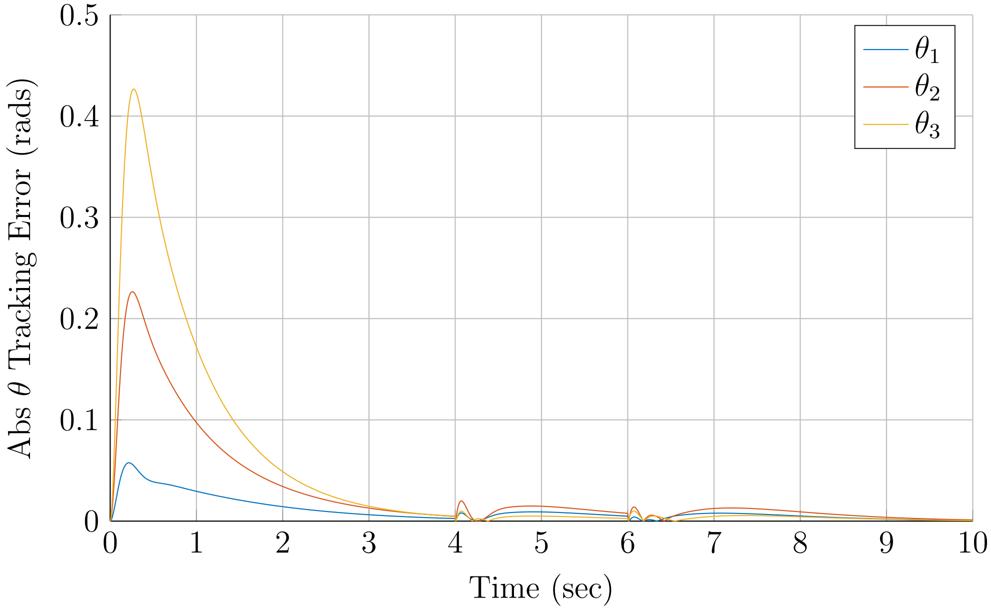


Figure 5.32: Computed Torque Method joint angles (above) and joint angle tracking error (below)

5.6.2.2 Sliding Mode Control Results

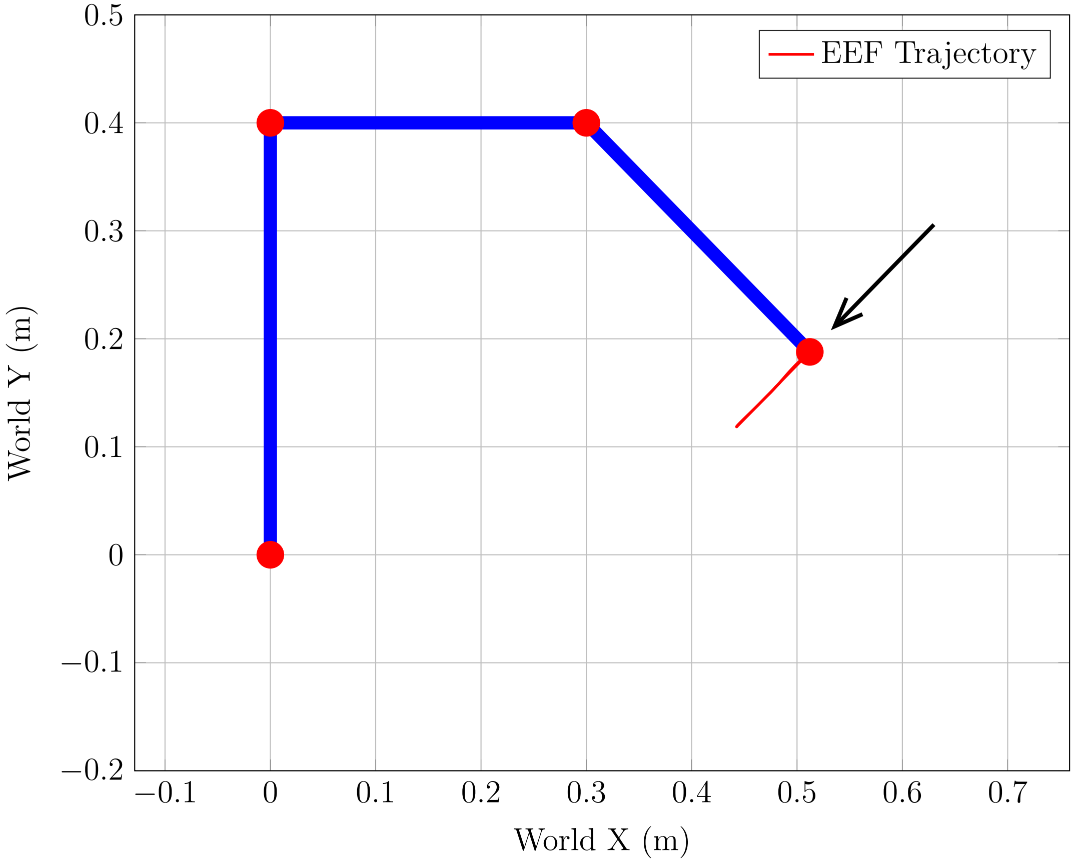


Figure 5.33: SMC punch simulation (with significant disturbances). Arrow represents punch force direction

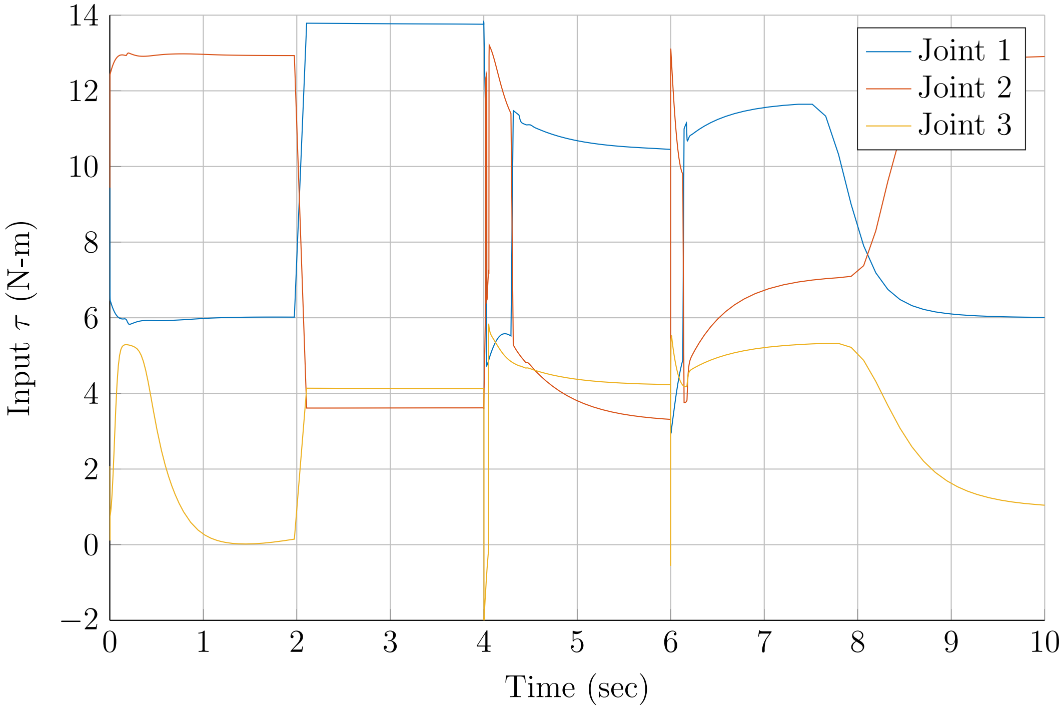
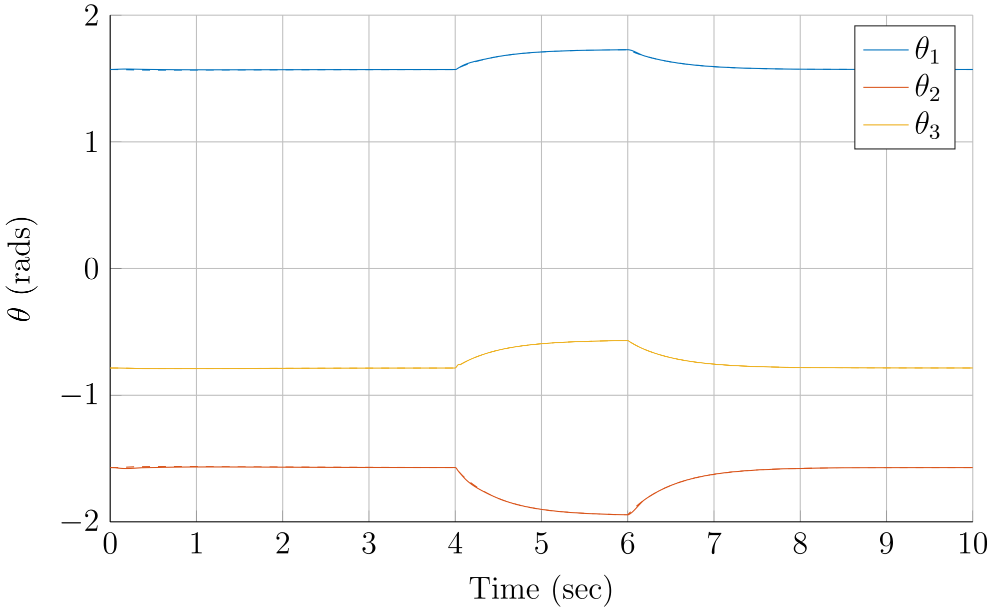


Figure 5.34: SMC input torque



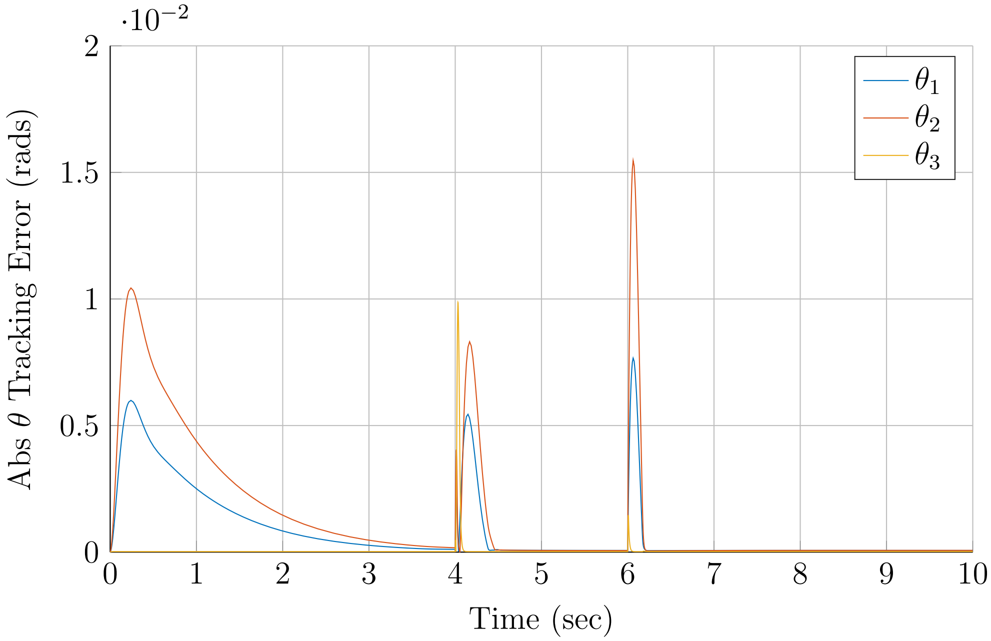


Figure 5.35: SMC joint angles (above) and joint angle tracking error (below)

5.7 Conclusion

The results of the circle tracker and punch simulations give insight as to how the robot arm hardware will behave if either of the two nonlinear control laws are implemented. The difference between the two control laws is quite apparent by looking at Figures 5.19, 5.22, 5.32, and 5.33. In the presence of significant bounded disturbance torques, the sliding mode controller guarantees a faster response toward equilibrium with a minimal error margin for both circular tracking and force control scenarios. These simulations show that a sliding mode controller is a feasible solution to EEF force control during handling tasks. However, the simulation is limited in scope and sliding mode control has hardware drawbacks.[4] For example, the PID gain parameters and sliding control parameters η and λ were not changed throughout this simulation, but have a huge effect on the behavior of the system. The main gain parameter η can cause significant overshoot if too high, and variation of λ will change the speed of the sliding surface. The outer-loop linear controller will ultimately determine successful tracking of the system.

The details concerning the drawbacks of sliding mode control, alternative control laws, simulation improvements, and real time hardware control are discussed in the next chapter. As simplistic as this simulation is, it is apparent from the results that the introduction of a sliding surface parameter improves the robustness of the controller. It also provides additional safety and operational efficiencies over the open loop approach shown in Chapter 4. It is recommended that

[4] Sliding mode drawbacks such as chattering and plant damage can be remedied with techniques such as adaptive gain tuning and time delay techniques. This is discussed further in Chapter 6.

further research be done to implement a robust controller on glovebox-deployed PDMs.

Chapter 6

Conclusions and Future Work

6.1 Research Summary

Industrial robots are seeing more use as they are slowly introduced into hazardous environments. This particular document details the automation of a Port-Deployed Manipulator (PDM) to replace a human worker to exposure in a hazardous radiological environment for one particular task. This task consists of the size reduction of a hemispherical plutonium shell using a micro punch. A successful material reduction algorithm was implemented on a PDM and demonstrated on a stainless steel surrogate. The algorithm was developed to reduce a hemispherical shell into pieces small enough to fit in a crucible without generating sharps or fines. The algorithm works for a variety of different hemishell radii by utilizing online motion planning and state-of-the-art sampling based motion planning. Although this algorithm was demonstrated using a particular robot, the software base is hardware-agnostic. This allows the algorithm to be implemented on a variety of platforms.

In addition to an open-loop hardware demonstration, a closed-loop simulation of the nonlinear system was developed in MATLAB using two different nonlinear controllers. A feedback linearizing controller and a sliding mode controller were used to introduce trajectory tracking on a 3-DOF simulated robot. Two simulation scenarios were conducted with each controller: a circular EEF trajectory tracking scenario and a force-feedback compliance scenario. Unmodeled disturbances were added to the dynamics of the system during each scenario to compare the robustness of each controller. The circular EEF trajectory tracking simulation demonstrated the tracking capabilities of both of the combined PID-nonlinear control schemes. The force-feedback compliance simulation demonstrated the behavior of the closed-loop system in response to a similar punch force that is induced during the material reduction process. The sliding mode controller outperformed the computed torque controller by tracking the desired trajectory with significantly smaller error bounds and a faster response time.

6.2 Future Work

The material reduction algorithm demonstrated the feasibility of an automated solution in a laboratory setting. The robot and supporting hardware itself may change in order to meet additional requirements for the hot workspace. The software base will likely need revisions if cold testing results deem it necessary. The implementation of closed-loop force feedback during motion planning needs to address issues such as discrete time control, input saturation, plant damage, and time delay. This section will outline future research topics that are of interest for material reduction task automation.

6.2.1 Material Reduction Hardware Selection

All research and development for the material reduction task was completed using an SIA5 robot arm, which has served as a hardware platform for much of the research done in the UT Nuclear Robotics Group. However, it may not be an ideal hardware platform for hot cell implementation. Further investigation should be done to identify other hardware platforms for glovebox automation that may improve certain automation qualities. The SIA10, for example, could also serve as a PDM. The SIA10 has a much more favorable geometric configuration, allowing for more IK solutions to be calculated in a given workspace by virtue of its longer link lengths. It is also capable of manipulating twice the payload of an SIA5.

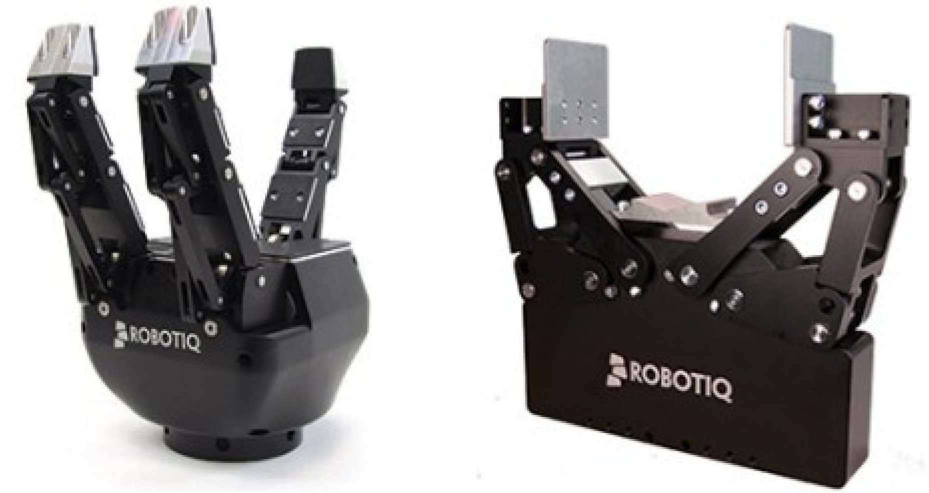


Figure 6.1: Robotiq 3-finger gripper (left) and 2-finger gripper (right) [47]

Another option that should be investigated is a dual-arm configuration. This would be useful if finger grippers (such as the Robotiq grippers shown in figure 6.1) are used as EEFs. This would allow one robot arm to transfer the hemishell to the other robot arm to provide a more favorable object/tool transformation during motion planning. Additionally, the robot arm in question need not be a PDM, but a robot that fits completely inside of a glovebox. This would protect the outside environment to contamination from the contents of a glovebox. Smaller 5-DOF Fanuc LR Mate 100i robots are used in the ARIES project that fit inside of a hot cell [38]. Although motion planning development on a geometrically smaller platform would be difficult, a smaller robot would likely be cheaper to replace should radiation compromise its integrity.

Similarly, the EEF used in the material reduction process does not need to be a vacuum gripper. Although vacuum grippers are commonly used in industrial manufacturing, they are not without their limitations. Running a pneumatic line through the inside of a hot glovebox could potentially be dangerous as it could spread contaminants. The compliance of the gripper suction cup may also interfere with motion planning, as the plutonium hemishell's weight could change its orientation with respect to the EEF during robot motion. A suction cup also takes up surface area on the hemishell, which will not be reduced by the punch. This means a human would have to manually reduce the last piece of the hemishell if it cannot fit inside of the crucible if the suction cup is too large. This could be remedied by researching an alternative method of rigidly attaching the hemishell in a desirable configuration onto the EEF without sacrificing surface area.

6.2.2 Material Collection

The material reduction process developed in this report does not have a method of collecting the slugs of material produced by the micro punch. Material accountability should be addressed before hot cell deployment. This would involve the creation a material collection apparatus that would hold pieces of the reduced hemishell in some sort of container. A container would need to be handled and its contents emptied into an induction crucible. The handling of the container could potentially be automated, as well as using capabilities already demonstrated by the NRG [24].

The collection of the hemishell pieces has been researched by an undergraduate design team at UT. This device was previously mentioned in Chapter 3.

The design team formed several ideas for a collection system, including a shaker table, feed ramp, and a pneumatic-assisted design. The team concluded that based on their research, the best material collection system is a funnel-fed tube system. The pieces of hemishell material would be gravity-fed down a tube into a container as they are produced from the tool. This design can be seen in figure 6.2.

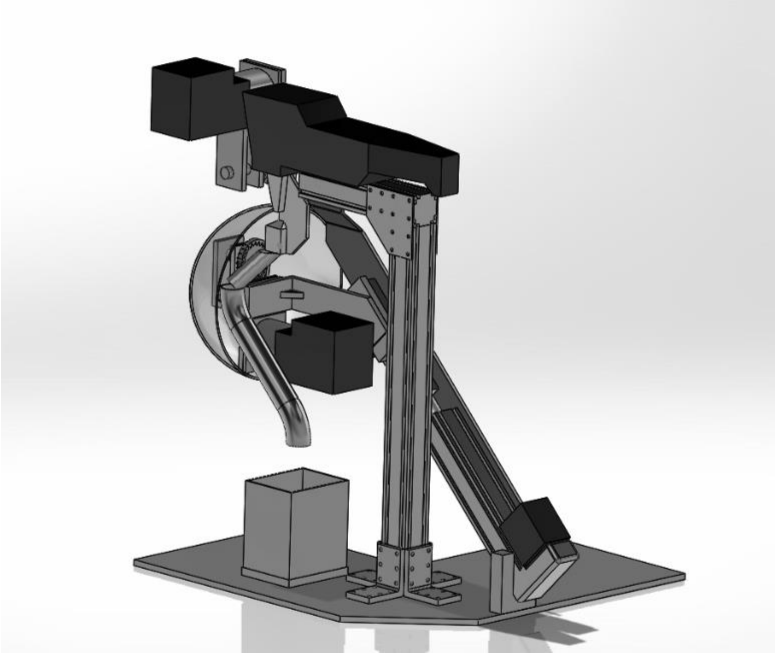


Figure 6.2: Undergraduate design team material collection prototype design

A similar material collection system could be developed for the micro punch that collects slugs of material in a standard DOE plutonium container.

6.2.3 Reactive Motion Planning

The motion planning algorithm developed for the material reduction process only allows the radius of the object to be specified prior to the start of the program. It is currently not suitable for objects to be reduced that are not perfectly hemispherical. However, the algorithm can be modified to include any number of objects that are revolute about an axis. Consider a perfect hemisphere as discussed in this report. This hemisphere can be defined by a Bézier curve revolved around a vertical axis. The algorithm developed for this report is only valid for one particular type of Bézier curve defined by its radius. Instead of only using a user-defined radius as the input for the material reduction algorithm, a user-defined Bézier curve and radius could be used as inputs. Such an algorithm would allow a variety of solid model configurations to be valid that are not perfectly hemispherical. A standard Bézier curve of order n is described by $n+1$ control points in 3-D space, expressed by P_i in equation 6.1.

$$\begin{aligned} B(t) &= \sum_{i=0}^{n} x_i b_{i,n}(t), t \in [0,1] \\ P_i &= \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} \end{aligned} \tag{6.1}$$

De Casteljau's algorithm is a recursive method of determining point locations along a known Bézier curve [18]. This is necessary to determine the location in space to place the pit in relation to the punch tool. The punch is then rotated by θ for a number of rotations based on the radius of the solid object. The next point of interest on the Bézier curve is calculated by varying the parameter t with De Casteljau's algorithm. Prautzsch et. al give a more detailed explanation of the algorithm [18].

This method of automated motion planning would assume that the user gave an input of a known curve to the program. If the user does not know how to describe a curve to fit the surface of any given pit, it may be possible to automate this step using a vision application. Visual feedback has been studied at the UT Nuclear Robotics Group and is occasionally used in industrial automation. O'Neil developed a vision algorithm that identified objects in a glovebox environment using visual point-cloud data [42]. This capability could be extended by developing an algorithm to match point cloud data of the curvature of a hemispherical object to a Bézier curve. Visual feedback would also be useful during the material reduction process to determine punch success or failure.

6.2.4 Closed-Loop Hardware Control

In order to implement the controllers presented in Chapter 5 to real world hardware, several factors must be addressed that were not included in the mathematical simulation. In the real world, joint servos have an output limit. This means that the input torques to the system are limited by a saturation function. Although the input torques in each of the simulations described in Chapter 5 are well within the range of standard industrial robots (on the order of several Newton-meters), handling heavier payloads will necessitate the use of a saturation function.

Real-world effects such as viscous friction and joint noise are unmodeled. The inertia matrix and other nonlinear terms in the nonlinear control law will not be known exactly and will have some kind of uncertainty associated with each term (ie. $M = \hat{M}, V = \hat{V}$, etc.). This is why more robust methods of nonlinear control should be researched further. However, implementing these controllers on hardware involves discretizing the control law as robots are commonly controlled digitally via a high-frequency control loop.

6.2.4.1 Discrete Control and Time Delays

A discrete control law must be used as opposed to the continuous controllers introduced in the previous chapter. Recall the sliding mode controller from the previous chapter. The sliding mode controller has its roots in continuous-time relay control, but sliding mode control theory assumes that switching can be achieved at an infinitely fast rate. In reality, this is not possible. There are several methods to compensate for this. Consider the block diagram given in figure 6.3. A hybrid closed-loop system is created by incorporating a digital controller with a continuous system.

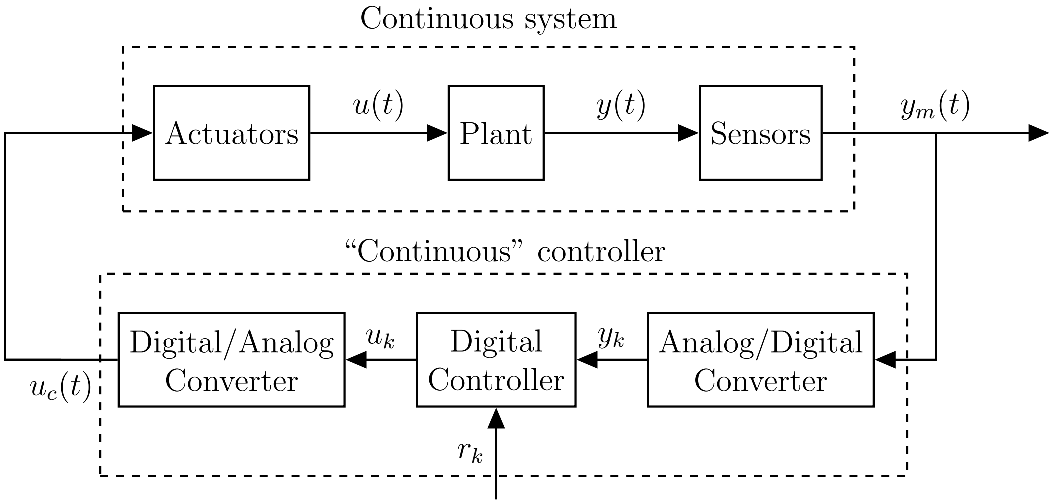


Figure 6.3: Block diagram of closed-loop discrete system [13]

The sliding mode controller needs to be discretized. One of the most common ways to discretize a continuous signal is by using the delta model. The control signal given by the delta model is similar to Euler's method of predicting the next time step function value and is given by equation 6.2 [13].

$$x_{k+1} = x_k + T f_\delta(x_k, u_k) \tag{6.2}$$

where

$$\lim_{T \to 0} f_\delta(x_k u_k) = f(x_k, u_k) = \lim_{T \to 0} \frac{x_{k+1} - x_k}{T} \tag{6.3}$$

Discrete sliding mode controller designs have been researched extensively [13] [40]. However, using a discrete controller creates challenges. Latency between the actuators and plant, between the plant and the sensors, and between the controller and the actuators cause a time delay. If the sampling frequency is high, this may not be a noticeable issue. However, a low sampling rate may necessitate the use of a compensator. One method of time delay compensation is the Smith Predictor Method, previously mentioned in Chapter 2 [54].

Digitally sampling a continuous waveform leads to discontinuities, even at a high control rate. In some of the worst cases, this may lead to *aliasing*, where the meaning of the waveform is lost due to poor sampling of the signal. This phenomenon is illustrated in figure 6.4.

This applies to sampling from sensors such as a force/torque sensor. This may not be an issue if the hardware is operating at a high control rate, but some cases may necessitate the use of a digital filter if data quantization is significant. Quantization of the input signal given by the discrete controller will also occur, but as long as the control rate is high and controller is robust, the system should remain stable. Force/torque sensor data filtering and quantization in robotic systems has been previously researched by Zelenak of the UT NRG [56].

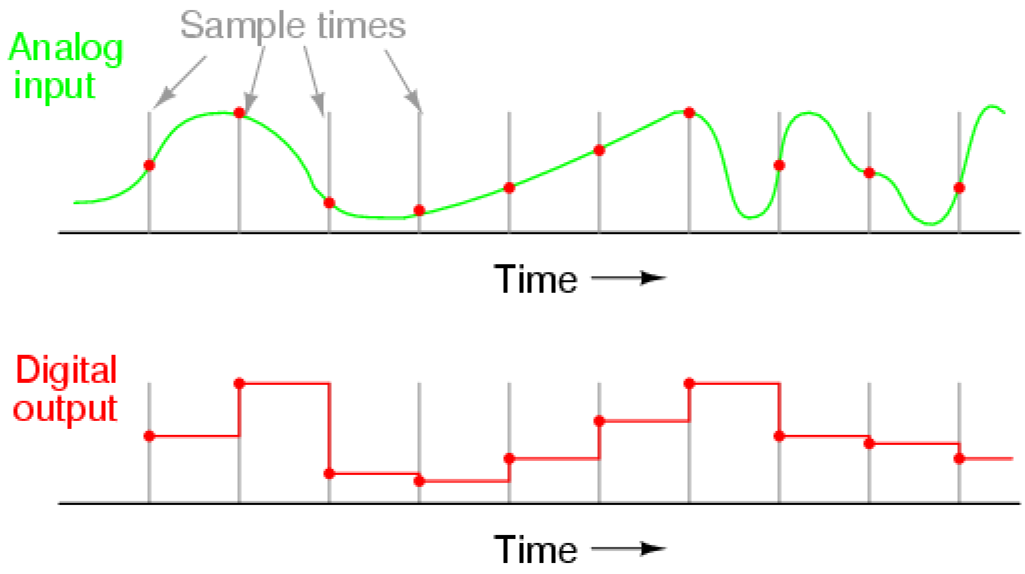


Figure 6.4: Digital sampling of a continuous signal [27]

6.2.4.2 Adaptive Control

The outer-loop PID gains were not optimized for this particular task but were based on user-intuition. A more complex method of gain tuning would allow the system to essentially "self-tune" based on the behavior of the system in previous time intervals [12]. This adaptive interaction algorithm could potentially optimize tracking for any control scheme that relied on a linear outer-loop controller.

Similarly, an adaptive sliding mode control law could be tested to improve the gains associated with the sliding surface term. A reasonable η gain value was chosen for the purposes of simulation in the previous chapter. However, too large of an η value would overshoot the desired sliding surface and cause "chattering" near the sliding surface. This could lead to plant servo/actuator damage. The use of an adaptive sliding mode control algorithm could alleviate some of the issues of sliding mode control. The basic algorithm to implement an adaptive sliding mode controller is given by Slotine et al [51].

6.2.4.3 Impedance/Admittance Control

The force-feedback control law introduced in the 3-DOF simulation was modeled after a linear spring. This is a simplified version of an impedance control law. An overview of Hogan's impedance control law was given in Chapter 2. It might prove useful to expand this linear relation of force and position change into a full impedance controller. This is accomplished by adding a mass and acceleration term, and a damping factor associated with the EEF velocity. Depending on hardware test results, a full impedance controller may improve performance by introducing human-like compliance. In any case, further research can be done to develop a more suitable force control law.

6.3 Concluding Remarks

The work presented in this report furthers the development of automated systems and control in hazardous radiological environments. Although there is work to be done before system deployment, the research presented in this report is applicable to a variety of automated glovebox systems. In particular, this report addresses:

- Collision detection and object recognition based on user-specified parameters.
- Joint torque monitoring
- Online motion planning for contact tasks

- *Object-in-hand* industrial manufacturing
- Grasping and handling of nuclear material
- Software compliance via robust nonlinear control methods

Although these topics are not new avenues of research, the integration of many different aspects of industrial automation is what makes the work presented in this report unique. Future development of material handling by robots in a nuclear environment will utilize many or all of these techniques for safe and efficient operation. It is recommended that the automation of glovebox tasks such as the one described in this report continue in order to increase safety across the DOE complex.

Appendices

Appendix A

Material Reduction Pseudocode

```
**Data**: User-specified radius $R$ and tool radius $r_{tool}$
**Result**: Size-reduced hemishell surrogate
ROS[workcell]→initialize();
$robot_{arm}$ →pick(bowl); // Pick up the bowl
$robot_{arm}$ →move(Pre-punch); // Move arm to prepunch position
R = *User-input*;
$r_{tool}$ = *User-input*;
$\theta = \frac{r_{tool}}{R}$;
$\phi = \theta$ ;
$N_\phi = \frac{R-d_{suction}}{r_{tool}}$;
**while** $N_{\phi_i} < N_\phi$ **do**
    $robot_{joint(t)}$ → move(-180 degrees); // Reset redundant joint
    **if** $N_{\phi_i} \neq 1$ **then**
        $robot_{arm}$ → moveCartesian(withdraw); // Withdraw arm
        $robot_{bowl}$ → rotate(XZ plane, $\phi$); // Rotate bowl to next ring
        $robot_{arm}$ → moveCartesian(punch); // Move arm into punch
    **end**
    **while** $\theta_i < 180$ **do**
        Punch();
        $robot_{joint(t)}$ → move($\theta$) ; // Rotate redundant joint
    **end**
    Punch();
    $N_{\phi_i} + = 1$; // Increment ring counter
**end**
$robot_{arm}$ →move(stow); // Move arm back to stow position
$robot_{gripper}$ → off; // Turn vacuum gripper off
Program→End(); // End the program
```

Algorithm 1: Material Reduction Algorithm

Appendix B

Dynamics of a Three-Link Planar Elbow Arm

The development of the Euler-Lagrange dynamic equations for a three-link planar arm are based on equations presented by Lewis et al. [14]. The free-body diagram model used in this section is the same one presented in Chapter 5. It is repeated here in figure B.1.

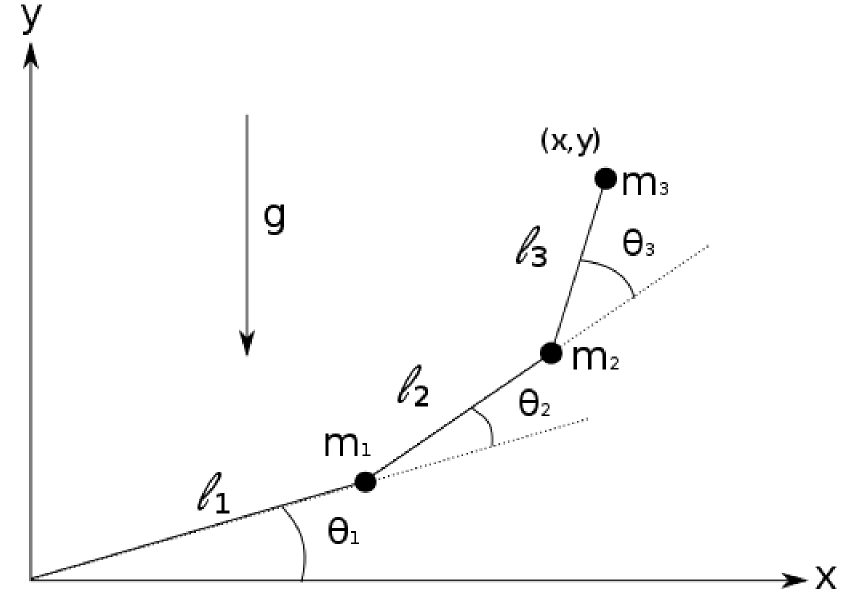


Figure B.1: Free body diagram of *3-DOF* planar robot arm

For link 1,

$$\begin{aligned} K_1 &= \frac{1}{2}m_1{l_1}^2\dot{\theta}_1^2 \\ P_1 &= m_1 g l_1 sin\theta_1 \end{aligned} \tag{B.1}$$

For link 2, it is necessary to find the end coordinates of the point mass m_2.

$$\begin{aligned} x_2 &= l_1 cos\theta_1 + l_2 cos(\theta_1 + \theta_2) \\ y_2 &= l_2 sin\theta_1 + l_2 sin(\theta_1 + \theta_2) \\ \dot{x}_2 &= -l_1\dot{\theta}_1 sin\theta_1 - l_2(\dot{\theta}_1 + \dot{\theta}_2) sin(\theta_1 + \theta_2) \\ \dot{y}_2 &= l_1\dot{\theta}_1 cos\theta_1 + l_2(\dot{\theta}_1 + \dot{\theta}_2) cos(\theta_1 + \theta_2) \end{aligned} \tag{B.2}$$

$${v_2}^2 = \dot{x}_2^2 + \dot{y}_2^2 = {l_1}^2\dot{\theta}_1^2 + {l_2}^2{(\dot{\theta}_1 + \dot{\theta}_2)}^2 + 2l_1 l_2\dot{\theta}_1(\dot{\theta}_1 + \dot{\theta}_2) cos\theta_2 \tag{B.3}$$

The kinetic and potential energies:

$$\begin{aligned} K_2 = \frac{1}{2}m_2{v_2}^2 = \frac{1}{2}m_2{l_1}^2\dot{\theta}_1^2 + \frac{1}{2}m_2{l_2}^2{(\dot{\theta}_1^2 + \dot{\theta}_2)}^2 + m_2 l_1 l_2\dot{\theta}_1(\dot{\theta}_1 + \dot{\theta}_2) cos\theta_2 \\ P_2 = m_2 g y_2 = m_2 g[l_1 sin\theta_1 + l_2 sin(\theta_1 + \theta_2)] \end{aligned} \tag{B.4}$$

The same process can be applied to link 3.

$$\begin{aligned} x_3 &= l_1 cos\theta_1 + l_2 cos(\theta_1 + \theta_2) + l_3 cos(\theta_1 + \theta_2 + \theta_3) \\ y_3 &= l_1 sin\theta_1 + l_2 sin(\theta_1 + \theta_2) + l_3 sin(\theta_1 + \theta_2 + \theta_3) \\ \dot{x}_3 &= -l_1\dot{\theta}_1 sin\theta_1 - l_2(\dot{\theta}_1 + \dot{\theta}_2) sin(\theta_1 + \theta_2) - \ldots \\ & l_3(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) sin(\theta_1 + \theta_2 + \theta_3) \\ \dot{y}_3 &= l_1\dot{\theta}_1 cos\theta_1 + l_2(\dot{\theta}_1 + \dot{\theta}_2) cos(\theta_1 + \theta_2) + \ldots \\ & l_3(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) cos(\theta_1 + \theta_2 + \theta_3) \end{aligned} \tag{B.5}$$

$$
\begin{aligned}
{v_3}^2 = \dot{x}_3^2 + \dot{y}_3^2 = {l_1}^2\dot{\theta}_1^2 + {l_2}^2{(\dot{\theta}_1 + \dot{\theta}_2)}^2 + {l_3}^2{(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)}^2 + \ldots \\
2l_1l_2\dot{\theta}_1(\dot{\theta}_1 + \dot{\theta}_2)cos\theta_2 + 2l_1l_3\dot{\theta}_1(\dot{\theta}_1\dot{\theta}_2\dot{\theta}_3)cos(\theta_2 + \theta_3) + \ldots \\
2cosl_2l_3(\dot{\theta}_1 + \dot{\theta}_2)(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)cos\theta_3
\end{aligned}
\tag{B.6}
$$

The kinetic energy of link 3 is

$$
\begin{aligned}
K_3 = \frac{1}{2}m_3{v_3}^2 = \frac{1}{2}m_3{l_1}^2\dot{\theta}_1^2 + \frac{1}{2}m_3{l_2}^2{(\dot{\theta}_1 + \dot{\theta}_2)}^2 + \ldots \\
\frac{1}{2}m_3{l_3}^2{(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)}^2 + m_3l_1l_2\dot{\theta}_1(\dot{\theta}_1 + \dot{\theta}_2)cos\theta_2 + \ldots \\
+m_3l_1l_3\dot{\theta}_1(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)cos(\theta_2 + \theta_3) + \ldots \\
m_3cosl_2l_3(\dot{\theta}_1 + \dot{\theta}_2)(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)cos\theta_3
\end{aligned}
\tag{B.7}
$$

The potential energy of link 3 is

$$
P_3 = m_3gy_3 = m_3g[l_1sin\theta_1 + l_2sin(\theta_1 + \theta_2) + l_3sin(\theta_1 + \theta_2 + \theta_3)]
\tag{B.8}
$$

With these equations, the Lagrangian L can be solved:

$$
\begin{aligned}
L = K - P = K_1 + K_2 + K_3 - P_1 - P_2 - P_3 = \\
\frac{1}{2}(m_1 + m_2 + m_3){l_1}^2\dot{\theta}_1^2 + \frac{1}{2}(m_2 + m_3){l_2}^2{(\dot{\theta}_1 + \dot{\theta}_2)}^2 + \ldots \\
\frac{1}{2}m_3{l_3}^2{(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)}^2 + (m_3 + m_2)l_1l_2\dot{\theta}_1(\dot{\theta}_1 + \dot{\theta}_2)cos\theta_2 + \ldots \\
m_3l_1l_3\dot{\theta}_1(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)cos(\theta_2 + \theta_3) + \ldots \\
m_3l_2l_3(\dot{\theta}_1 + \dot{\theta}_2)(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)cos\theta_3 - \ldots \\
(m_1 + m_2 + m_3)gl_1sin\theta_1 - (m_2 + m_3)gl_2sin(\theta_1 + \theta_2) - \ldots \\
m_3gl_3sin(\theta_1 + \theta_2 + \theta_3)
\end{aligned}
\tag{B.9}
$$

Solving for each joint torque:

$$\begin{aligned}\frac{\partial L}{\partial \dot{\theta}_1} = (m_1+m_2+m_3){l_1}^2\dot{\theta}_1 + (m_2+m_3){l_2}^2(\dot{\theta}_1+\dot{\theta}_2) + \dots \\ m_3{l_3}^2(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3) + (m_2+m_3)l_1l_2(2\dot{\theta}_1+\dot{\theta}_2)cos\theta_2 + \dots \\ m_3cos(\theta_2+\theta_3)l_1l_3(2\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3) + \dots \\ m_3cos\theta_3l_2l_3(2\dot{\theta}_1+2\dot{\theta}_2+\dot{\theta}_3)\end{aligned} \tag{B.10}$$

$$\begin{aligned}\frac{d}{dt}(\frac{\partial L}{\partial \dot{\theta}_1}) = (m_1+m_2+m_3){l_2}^2(\ddot{\theta}_1) + (m_2+m_3){l_2}^2(\ddot{\theta}_1+\ddot{\theta}_2) + \dots \\ m_3{l_3}^2(\ddot{\theta}_1+\ddot{\theta}_2+\ddot{\theta}_3) + (m_2+m_3)l_1l_2(2\ddot{\theta}_1+\ddot{\theta}_2)cos\theta_2 - \dots \\ (m_2+m_3)l_1l_2(2\dot{\theta}_1\dot{\theta}_2+\dot{\theta}_1^2)sin\theta_2 + m_3l_1l_3(2\ddot{\theta}_1\ddot{\theta}_2+\ddot{\theta}_1)cos(\theta_2+\theta_3) - \dots \\ m_3l_1l_3(2\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3)sin(\theta_2+\theta_3)(\dot{\theta}_2+\dot{\theta}_3) + \dots \\ m_3cos\theta_3l_2l_3(2\ddot{\theta}_1+2\ddot{\theta}_2+\ddot{\theta}_3) - m_3l_2l_3(2\dot{\theta}_1+2\dot{\theta}_2+\dot{\theta}_3)sin(\theta_3)\dot{\theta}_3\end{aligned} \tag{B.11}$$

$$\begin{aligned}\frac{\partial L}{\partial \theta_1} = -(m_1+m_2+m_3)gl_1cos\theta_1 - \dots \\ (m_2+m_3)gl_2cos(\theta_1+\theta_2) - m_3gl_3cos(\theta_1+\theta_2+\theta_3)\end{aligned} \tag{B.12}$$

$$\begin{aligned}\frac{\partial L}{\partial \dot{\theta}_2} = (m_2+m_3){l_2}^2(\dot{\theta}_1+\dot{\theta}_2) + m_3{l_3}^2(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3) + \dots \\ (m_2+m_3)l_1l_2\dot{\theta}_2cos\theta_2 + m_3cos(\theta_2+\theta_3)l_1l_3\dot{\theta}_1 + m_3cos\theta_3l_2l_3(2\dot{\theta}_1+2\dot{\theta}_2+\dot{\theta}_3)\end{aligned} \tag{B.13}$$

$$\begin{aligned}\frac{d}{dt}(\frac{\partial L}{\partial \dot{\theta}_2}) = (m_2+m_3){l_2}^2(\ddot{\theta}_1+\ddot{\theta}_2) + m_3{l_3}^2(\ddot{\theta}_1+\ddot{\theta}_2+\ddot{\theta}_3) + \dots \\ (m_2+m_3)l_1l_2\ddot{\theta}_1cos\theta_2 - (m_2+m_3)l_1l_2\dot{\theta}_1\dot{\theta}_2sin\theta_2 + \dots \\ m_3l_1l_3\ddot{\theta}_1cos(\theta_2+\theta_3) - m_3l_1l_3\dot{\theta}_1(\dot{\theta}_2+\dot{\theta}_3)sin(\theta_2+\theta_3) + \dots \\ m_3cos\theta_3l_2l_3(2\ddot{\theta}_1+2\ddot{\theta}_2+\ddot{\theta}_3) - m_3l_2l_3(2\dot{\theta}_1+2\dot{\theta}_2+\dot{\theta}_3)\dot{\theta}_3sin\theta_3\end{aligned} \tag{B.14}$$

$$\begin{aligned}\frac{\partial L}{\partial \theta_2} = -(m_2+m_3)l_1l_2\dot{\theta}_1(\dot{\theta}_1+\dot{\theta}_2)sin\theta_2 - \ldots \\ m_3l_1l_3\dot{\theta}_1(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3)sin(\theta_2+\theta_3) - (m_2+m_3)gl_2cos(\theta_1+\theta_2) - \ldots \\ m_3gl_3cos(\theta_1+\theta_2+\theta_3)\end{aligned} \tag{B.15}$$

$$\begin{aligned}\frac{\partial L}{\partial \dot{\theta}_3} = m_3{l_3}^2(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3)m_3cos(\theta_2+\theta_3)l_1l_3\dot{\theta}_1 + \ldots \\ m_3cos\theta_3l_2l_3(\dot{\theta}_1+\dot{\theta}_2)\end{aligned} \tag{B.16}$$

$$\begin{aligned}\frac{d}{dt}(\frac{\partial L}{\partial \dot{\theta}_3}) = m_3{l_3}^2(\ddot{\theta}_1+\ddot{\theta}_2+\ddot{\theta}_3) + m_3l_1l_3\ddot{\theta}_1cos(\theta_2+\theta_3) - \ldots \\ m_3l_1l_2\dot{\theta}_1(\dot{\theta}_2+\dot{\theta}_3)sin(\theta_2+\theta_3) + m_3l_2l_3(\ddot{\theta}_1+\ddot{\theta}_2)cos\theta_3 - \ldots \\ m_3l_2l_3(\dot{\theta}_1+\dot{\theta}_2)\dot{\theta}_3sin\theta_3\end{aligned} \tag{B.17}$$

$$\begin{aligned}\frac{\partial L}{\partial \theta_3} = -m_3l_1l_3\dot{\theta}_1(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3)sin(\theta_2+\theta_3) - \ldots \\ m_3l_2l_3(\dot{\theta}_1+\dot{\theta}_2)(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3)sin\theta_3 - m_3gl_3cos(\theta_1+\theta_2+\theta_3)\end{aligned} \tag{B.18}$$

Using *Lagrange's Equation of Motion*:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \tau \tag{B.19}$$

By calculating τ_i from Equations B.10 through B.18, the resulting torques can be put into the form

$$M(q)\ddot{q} + V(q,\dot{q}) + G(q) = \tau \tag{B.20}$$

The friction dynamics are represented by joint-localized dynamic friction F_d and viscous friction F_v terms:

$$M(q)\ddot{q} + V(q,\dot{q}) + G(q) + F_v(\dot{q}) + F_d(\dot{q}) = \tau \tag{B.21}$$

This equation represents the standard manipulator dynamics for a generic robot arm that includes friction, where:

$$\begin{aligned} F_v\dot{q} &= vec[f_{vi}\dot{q}_i] \\ F_d\dot{q} &= vec[f_{di}sgn(\dot{q}_i)] \end{aligned} \tag{B.22}$$

The terms f_v and f_d are scaling terms. The terms used for the arm simulation were chosen to closely match the friction model created by a previous UT NRG member for the SIA5 arm [49]. In his dissertation, Schroeder experimentally determined joint friction models based on current.

Calculating the nonlinear terms for this arm,

$$M = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \tag{B.23}$$

$$V = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} \tag{B.24}$$

$$G = \begin{bmatrix} G_1 \\ G_2 \\ G_3 \end{bmatrix} \tag{B.25}$$

$$
\begin{aligned}
M_{11} = (m_1 + m_2 + m_3){l_1}^2 + (m_2 + m_3){l_2}^2 + m_3{l_3}^2 + \ldots \\
2(m_2 + m_3)l_1l_2cos(\theta_2) + 2m_3l_1l_3cos(\theta_2 + \theta_3) + 2m_3l_2l_3cos(\theta_3) \\
M_{12} = M_{21} = (m_2 + m_3){l_2}^2 + m_3{l_3}^2 + (m_2 + m_3)l_1l_2cos(\theta_2) + \ldots \\
m_3l_1l_3cos(\theta_2 + \theta_3) + 2m_3l_2l_3cos(\theta_3) \\
M_{13} = M_{31} = m_3{l_2}^2 + m_3l_1l_3cos(\theta_2 + \theta_3) + m_3l_2l_3cos(\theta_3) \\
M_{22} = (m_2 + m_3){l_2}^2 + m_3{l_3}^2 + 2m_3l_2l_3cos(\theta_3) \\
M_{23} = M_{32} = m_3{l_3}^2 + m_3l_2l_3cos(\theta_3) \\
M_{33} = m_3{l_3}^2
\end{aligned}
\tag{B.26}
$$

$$
\begin{aligned}
V_1 = -(m_2 + m_3)l_1l_2(2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2 sin(\theta_2) - \ldots \\
m_3l_1l_3(2\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)(\dot{\theta}_2 + \dot{\theta}_3)sin(\theta_1 + \theta_3) - \ldots \\
m_3l_2l_3(2\dot{\theta}_1 + 2\dot{\theta}_2 + \dot{\theta}_3)\dot{\theta}_3sin(\theta_3) \\
V_2 = -(m_1 + m_3)l_1l_2\dot{\theta}_1\dot{\theta}_2sin(\theta_2) - m_3l_1l_3\dot{\theta}_1(\dot{\theta}_2 + \dot{\theta}_3)sin(\theta_2 + \theta_3) - \ldots \\
m_3l_2l_3(2\dot{\theta}_1 + 2\dot{\theta}_2 + \dot{\theta}_3)\dot{\theta}_3sin(\dot{\theta}_3) + (m_2 + m_3)l_1l_2\dot{\theta}_1(\dot{\theta}_1 + \dot{\theta}_2)sin(\theta_2) + \ldots \\
m_3l_1l_3\dot{\theta}_1(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)sin(\dot{\theta}_1 + \theta_3) \\
V_3 = -m_3l_1l_2\dot{\theta}_1(\dot{\theta}_2 + \dot{\theta}_3)sin(\theta_2 + \theta_3) - m_3l_2l_3(\dot{\theta}_1 + \dot{\theta}_2)\dot{\theta}_3sin(\theta_3) + \ldots \\
m_3l_1l_3\dot{\theta}_1(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)sin(\theta_2 + \theta_3) + m_3l_2l_3(\dot{\theta}_1 + \dot{\theta}_2)(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)sin(\theta_3)
\end{aligned}
\tag{B.27}
$$

$$
\begin{aligned}
G_1 = (m_1 + m_2 + m_3)gl_1cos(\theta_1) + (m_2 + m_3)gl_2cos(\theta_1 + \theta_2) + \ldots \\
m_3gl_3cos(\theta_1 + \theta_2 + \theta_3) \\
G_2 = (m_2 + m_3)gl_2cos(\theta_1 + \theta_2) + m_3gl_3cos(\theta_1 + \theta_2 + \theta_3) \\
G_3 = m_3gl_3cos(\theta_1 + \theta_2 + \theta_3)
\end{aligned}
\tag{B.28}
$$

$$
\begin{aligned}
Fr_1 &= fv_1\dot{\theta}_1 + fd_1 sign(\dot{\theta}_1) \\
Fr_2 &= fv_2\dot{\theta}_2 + fd_2 sign(\dot{\theta}_2) \\
Fr_3 &= fv_3\dot{\theta}_3 + fd_3 sign(\dot{\theta}_3)
\end{aligned}
\tag{B.29}
$$

References

[1] Johnny Carson's Tonight Show. Television Show, 1966.

[2] Technical Committee ISO/TC 184. *ISO 8373*. International Organization for Standardization, 2012.

[3] Pieter Abbeel. Sampling-based motion planning. Lecture at UC Berkeley EECS, images from Lavalle, Planning Algorithms.

[4] Fun-Woong Bae Benito Fernandez and Louis J. Everett. Control of robot manipulator through robust sliding linearization. In *IEEE International Conference on Robotics and Automation*. IEEE, may 1990.

[5] Cheryl Lynn Brabec. A shape primitive-based grasping strategy using visual object recognition in confined, hazardous environments. Master's thesis, University of Texas at Austin, dec 2013.

[6] US Congress. *Dismantling the Bomb and Managing the Nuclear Materials*. Office of Technology Assessment, sep 1993.

[7] John Craig. *Introduction to Robotics: Mechanics and Control*. Prentice Hall, second edition, aug 2004.

[8] AB Todtenkpf TR Kurfess DE Whitney, AC Edsall and AR Tate. Development and control of an automated robotic weld bead grinding system. *Journal of Dynamic Systems, Measurement, and Control*, jun 1990.

[9] Jacques Denavit and Richard Scheunemann Hartenberg. A kinematic notation for lower-pair mechanisms based on matrices. *Journal of Applied Mechanics: ASME DC*, 1955.

[10] Vincent Duchaine. General model of human-robot cooperation using a novel velocity based variable impedance control. In *EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, Tsukaba, mar 2007. Department of Mechanical Engineering, University of Laval, Quebec, IEEE.

[11] Gualtiero Fantoni et al. Grasping devices and methods in automated production processes. *CIRP Annals - Manufacturing Technology*, 63, 2014.

[12] Robert D. Brandt Feng Lin and George Saikalis. Self-tuning of PID controllers by adaptive interaction. In *American Controls Conference*. American Automatic Control Council, jun 2000.

[13] Benito Fernandez. Discrete SMC notes. Nonlinear controls lecture notes (UT Austin graduate mechanical engineering course), 2015.

[14] C.T. Abdallah F.L. Lewis and D.M. Dawson. *Control of Robot Manipulators*. Macmillan Publishing Company, 1993.

[15] Andras Kovacs Gabor Erdos, Zsolt Kemeny and Jozsef Vancza. Planning of remote laser welding processes. *Procedia CIRP*, 7, 2013.

[16] Lester Godwin. Controller interfaces for robotic surface finishing applications. Technical report, PushCorp. Inc, oct 1998. Presented at: The 4th Annual RIA Grinding, Deburring and Finishing Workshop.

[17] M. Gopal. *Control Systems: Principles and Design.* McGraw-Hill Education, 2002.

[18] Wolfgang Boehm Hartmut Prautzsch and Marco Paluszny. *Bezier and B-Spline Techniques.* Springer, 2002.

[19] Joseph Hashem. The requirements and implementation of dynamically-deployed robotic systems for use in confined, hazardous environments. Master's thesis, University of Texas at Austin, 2012.

[20] Neville Hogan. Impedance control: An approach to manipulation. *Journal of Dynamic Systems, Measurement, and Control*, 107, mar 1985.

[21] International Atomic Energy Agency. *The Role of Automation and Humans in Nuclear Power Plants*, oct 1992.

[22] Mark Moll Ioan Sucan and Lydia Kavraki. The Open Motion Planning Library. *IEEE Robotics and Automation Magazine*, dec 2012.

[23] Donald Johnson. A note on Dijkstra's shortest path algorithm. *Journal of the ACM*, 1973.

[24] Brian O'Neil Joseph Hashem and Mitch Pryor. Integrating fixed and flexible solutions for glovebox automation. In *ANS EPRRSD - 13th Robotics and Remote Systems for Hazardous Environments.* The University of Texas at Austin, aug 2011.

[25] Sheldon Landsberger Joshua Williams, Mitch Pryor and Louis D. Schulte. A complete approach to reduce operator dosage in hazardous environments. In *3rd International Joint Topical Meeting On Emergency Preparedness & Response And Robotics & Remote Systems.* American Nuclear Society, 2011.

[26] Alexander Kleiner and Bernhard Nebel. Introduction to multi-agent programming: Search algorithms and path-finding. Lecture Notes.

[27] Tony R. Kuphaldt. *Lessons in Electric Circuits (digital)*. Published under Design Science License, fourth edition, 2007.

[28] Mitch Pryor Kyle Schroeder and Troy Harden. On the use of joint torque sensors for collision detection in a confined environment. In *3rd ANS EPRRSD - 13th Robotics and Remote Systems for Hazardous Environments, 11th Emergency Preparedness and Response*. American Nuclear Society, 2011.

[29] Desai JP Lanfranco AR, Castellanos AE and Meyers WC. Robotic surgery a current perspective. *Annals of Surgery*, 2004.

[30] Steven LaValle. Rapidly-Exploring Random Trees: A new tool for path planning. Technical report, Computer Science Department, Iowa State University, 1998.

[31] Steven M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

[32] Cindy Lawton and Amanda Castro. Ergonomic efforts to reduce isolator worker injuries. In *ISPE Tampa Conference*, feb 2010.

[33] DJ Leith and WE Leithead. Survey of gain-scheduling analysis and design. *International Journal of Control*, 73, 2000.

[34] Jean-Claude Latombe Lydia Kavraki, Petr Svestka and Mark Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4), aug 1996.

[35] D. C. Haley M. W. Noakes and W. D. Willis. The selective equipment removal system dual arm work module. In *ANS Sixth Topical Meeting on Robotics and Remote Systems*. American Nuclear Society, 1997.

[36] John Markoff. Skilled work, without the worker. New York Times article, aug 2012.

[37] Brigitt Martin. The secret life of Louis Slotin. *Alumni Journal of the University of Manitoba*, 59, dec 1999.

[38] Steven McKee. Actinide research quarterly 1st/2nd quarters 2008. Technical report, Los Alamos National Laboratory, 2008.

[39] Cynthia M. Renner Michael E. Cournoyer and Cynthia L. Kowalczyk. Lean Six Sigma tools for a Glovebox Glove Integrity Program. *Journal of Chemical Health and Safety*, 2011.

[40] Govert Monsees. *Discrete-Time Sliding Mode Control*. PhD thesis, Delft University of Technology, 2002.

[41] Shimon Y. Nof. *Handbook of Industrial Robotics*. Wiley, second edition, 1999.

[42] Brian O'Neil. *Object recognition and pose estimation for nuclear manipulation in nuclear materials handling applications*. PhD thesis, University of Texas at Austin, may 2013.

[43] T.W. Parks and C. S. Burrus. *Digital Filter Design (Topics in Digital Signal Processing)*. Wiley-Interscience, 1987.

[44] NJ Nilsson PE Hart and B Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4, jul 1968.

[45] Dave Nelson Bill Santistevan Wendel Brown Pete Pittman, Torsten Staab. Automation of the LANL ARIES lathe glovebox. Technical report, Los Alamos National Laboratory, 2001. Submitted to The American Nuclear Society Ninth Topical Meeting on Robotics and Remote Systems.

[46] Conley Ken. Gerkey Brian P.. Faust Josh. Foote Tully. Leibs Jeremy. Wheeler Rob. Quigley, Morgan. and Andrew Y. Ng. ROS: an open-source Robot Operating System. In *ICRA Workshop on Open Source Software*, 2009.

[47] Robotiq. Robotiq official website. http://robotiq.com/.

[48] Marisa Sandoval. LANL meets plutonium pit production goal. Technical report, Los Alamos National Laboratory, 2011.

[49] Kyle Schroeder. *Requirements for Effective Collision Detection on Industrial Serial Manipulators*. PhD thesis, University of Texas at Austin, 2013.

[50] C.W. Sherman and R.S. Isenson. *First Interim Report On Project Hindsight (Summary)*. Department of Defense, Office of the Director of Defense Research and Engineering, Washington DC, 20301, jun 1966.

[51] Jean-Jacques E. Slotine and Weiping Li. *Applied Nonlinear Control*. Prentice Hall, 1991.

[52] Anthony Stentz. Optimal and efficient path planning for unknown and dynamic environments. *International Journal of Robotics and Automation*, 10, 1993.

[53] US Nuclear Regulatory Commission. *10 CFR 20.1003*, jul 2014.

[54] K. Warwick and D. Rees. *Industrial Digital Control Systems.* Peter Peregrinus Ltd on behalf of The Institution of Electrical Engineers, 1988.

[55] Linda K. Wood. Advanced Recover and Integrated Extraction System (ARIES) preconceptual design report. Technical report, Los Alamos National Laboratory, sep 1996.

[56] Andy Zelenak. Gaussian-based adaptive fuzzy control. *2014 IEEE Conference on Norbert Wiener in the 21st Century (21CW)*, 2014.

Vita

Clinton Peterson is a native Texan who grew up in Fort Worth. He moved to Austin and earned a Bachelor of Science in Mechanical Engineering from the University of Texas at Austin. He liked it so much he stayed for grad school. One of his favorite hobbies is speaking in third person. He will continue his robotics research full time at LANL.

E-mail Address: clinton.peterson@utexas.edu

This thesis was typeset with LaTeX[†] by the author.

[†] LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.