FLEXIBLE ADAPTATION OF ITERATIVE LEARNING CONTROL WITH APPLICATIONS TO SYNTHETIC BONE GRAFT MANUFACTURING

BY

DAVID J HOELZLE

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Mechanical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2011

Urbana, Illinois

Doctoral Committee:

Professor Andrew Alleyne, Chair and Co-Director of Research
Assistant Professor Amy Wagoner Johnson, Co-Director of Research
Professor Placid Ferreira
Professor Seth Hutchinson

Abstract

Additive manufacturing processes are powerful tools; they are capable of fabricating structures without expensive structure specific tooling – therefore structure designs can efficiently change from run-to-run – and they can integrate multiple distinct materials into a single structure. This work investigates one such additive manufacturing process, micro-Robotic Deposition (μRD), and its utility in fabricating advanced architecture synthetic bone grafts. These bone grafts, also known as synthetic bone scaffolds, are highly porous three-dimensional structures that provide a matrix to support the natural process of bone remodeling. Ideally, the synthetic scaffold will stimulate complete bone healing in a skeletal defect site and also resorb with time so that only natural tissue remains.

The objective of this research is to develop methods to integrate different regions with different porous microstructures into a single scaffold; there is evidence that scaffolds with designed regions of specific microstructures can be used to elicit a strong and directed bone ingrowth response that improves bone ingrowth rate and quality. The key contribution of this work is the development of a control algorithm that precisely places different build materials in specified locations, thereby the fabrication of advanced architecture scaffolds is feasible. Under previous control methods, designs were relegated to be composed of a single material. The control algorithm developed in this work is an adaptation of Iterative Learning Control (ILC), a control method that is typically best suited for mass manufacturing applications. This adaptation reorients the ILC framework such that it is more amenable to additive manufacturing

systems, such as μRD. Control efficacy is demonstrated by the fabrication of advanced architecture scaffolds. Scaffolds with contoured forms, multiple domains with distinct porous microstructures, and hollow cavities are feasible when the developed controller is used in conjunction with a novel manufacturing workflow in which scaffolds are filled within patterned molds that support overhanging features. An additional application demonstrates controller performance on the robot positioning problem; this work has implications for additive manufacturing in general.

To my parents

Acknowledgments

I would like to acknowledge my two advisers, Prof. Andrew Alleyne and Prof. Amy Wagoner Johnson, who have provided guidance throughout my doctoral training. They each provided a unique expertise that was fundamental to the success of this interdisciplinary research project. Most importantly, they were great examples of professionalism and their approach to research and education will guide me in my future academic career.

I would like to thank my family, who are a constant source of inspiration and have always been a support mechanism that was only a phone call away. To my parents, who have always stressed the importance of education and continue to be great examples of a diligent work ethic. To my two sisters, Angela and Melissa, both of whom continue to motivate me as they too work towards advanced degrees.

I would like to acknowledge the members of my examination committee who have guided my research and made it stronger. Many of them also wrote letters of recommendation for me and I thank them for their part in my success securing an academic position.

Many of my colleagues made direct contributions to this work. I would like to thank Samantha Polak, Shelby Svientek, Seongsu Yun, Heather August, and Kira Barton for their efforts.

I truly values the interactions I have had with fellow graduate students at the University of Illinois as our interactions have been a wonderful diversion from the rigorous path we have all chosen. Kira and Alex were the first people I met at Illinois

and have become close friends. It has been a great experience working with you, spending time with you, and even watching you two start a family together. Doug Bristow was an important mentor to me and he taught me much about Iterative Learning Control and the micro-Robotic Deposition system that I performed all the experimental results of this thesis on. He did an excellent job setting up the system and his efforts made it easy for me to make the system modifications necessary for my research. Sheeny Lan taught me how to prepare the hydroxyapatite build materials which enabled me to investigate interesting controls tools almost immediately. I have become really close friends with Neera, Deep, and Mike Sangid while at Illinois and I look forward to all of us being in proximity to each other for the next few years. I would like to thank Grace for teaching me how to enjoy life a little more. I would also like to thank Mike for being the best roommate I have ever had. To the rest of the individuals that have made my graduate time at Illinois at enjoyable experience – Mbikayi, Damena, Newton, Sandipan, Vikas, Tim, Bin, David Li, Nanjun, Yangmin, Erick, Shreyas, Louis, Enric, Brian, Scott, and Rich – thank you for always being a friend.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AM	Additive Manufacturing
β-TCP	β-Tricalcium Phosphate
BTILC	Basis Task Iterative Learning Control
CaP	Calcium Phosphate
FDM	Fused Deposition Modeling
HA	Hydroxyapatite
ILC	Iterative Learning Control
LTI	Linear Time-Invariant
m_1	Material 1; no induced microporosity
m_2	Material 2; includes $\approx$50 $vol\%$ microporosity
μCT	micro-Computed Tomography
μRD	micro-Robotic Deposition
MIMO	Multiple Input Multiple Output
P-D	Proportional-Derivative
RMS	Root Mean Square
SEM	Scanning Electron Microscopy
SISO	Single Input Single Output
SLA	Stereolithography Apparatus
SLS	Selective Laser Sintering
tbasis	Task basis
tspan	Task span

List of Symbols

$d(k)$	Disturbance signal
D	Number of configurations
D^O	Dynamic space
Δ	Time shift
$e(k)$	Error signal
$\dot{\gamma}$	Shear rate
H_d	System operator
$k = 0, \ldots K-1$	Discrete time index
$k_d(z)$	Feedback controller
K	Discrete time signal length
$L(q)$	Learning filter
m	Fluid consistency index
m_n	Multiplicity index
M_n	Number of multiplicities
n	Flow behavior index
N	Number of basis tasks
$\mathbf{O}$	Operation set adjacency matrix
$\bar{\sigma}$	Maximum singular value
P	Number of training routines
P_r	Syringe reservoir pressure

q	Forward time shift
$Q(q)$	Q-filter
Q_{in}	Plunger displacement rate
Q_{out}	Volumetric flowrate
$r(k)$	Reference signal or basis task
ρ	Spectral radius
R^O	Operation space
$\mathfrak{s}(k)$	Unit step function
τ	Shear stress
$\mathbf{T}$	Training set adjacency matrix
$u(k)$	Input signal or basis signal
$x_d(k)$	Configuration index
$x^j(k)$	Iteration index
$x^{L(j)}(k)$	Denotes conventional ILC
$x_n(k)$	Basis task index
$x^O(k)$	Denotes operation set
$x^T(k)$	Denotes training set
$x^{T_p}(k)$	Training routine index
$y(k)$	Output signal
z	Denotes discrete-time transfer function

Chapter 1

Introduction

This thesis investigates manufacturing and controls tools for the fabrication of advanced architecture synthetic bone grafts. In its broadest sense, this thesis is an investigation into a set of manufacturing tools termed Additive Manufacturing (AM) processes. Controls tools are developed to improve the performance of AM processes in general, modifying an established manufacturing control method to be more amenable to the flexible nature of AM. We have thoroughly investigated the manufacture of synthetic bone grafts by the AM technology micro-Robotic Deposition (μRD). Accordingly, the introduction provides a general overview of AM technologies in Section 1.1 – specifically those with utility for biomedical applications – and then provides a more rigorous review of synthetic bone graft design and manufacturing in Section 1.2. The thesis has a strong basis in control and automation and the chosen control methodology, Iterative Learning Control (ILC), has its own introductory chapter, Chapter 2, that details ILC and its tenets and aspects that we have modified for our purposes.

1.1 Additive Manufacturing (AM)

AM technologies are a set of manufacturing tools that employ the following work flow: design of structure in 3D modeling software package; translate the 3D model to AM machine language; directly create the three-dimensional structure in a layer-by-layer fashion [15]. There is no need for process planning or tooling and material preparation

as the work flow is nearly identical for every feasible design for a given system. The AM work flow is in direct contrast to more traditional manufacturing methods such as milling or injection molding. For instance, milling is a subtractive process that requires surface milling to prepare a material billet or purchasing of a specific size billet. Complex milled parts require multiple rotations of the billet to mill features in each side. Injection molding is additive, but the process occurs rapidly, leaving no opportunity to integrate multiple materials; additionally, injection molding requires expensive dies with long die development times.

The minimal preparation labor and lead-time make AM technologies extremely efficient in that the structural designs can change from run-to-run; manufacturing a diverse set of structures is almost as efficient as fabricating a set of identical structures. AM technologies have been commonly referred to as rapid prototyping [6]. Early uses were exactly that, prototypes. Architectural and engineering firms used the prototypes as inexpensive and rapidly generated tools to assess how a design fit and interacted with its surrounding environment before committing to a design to scale up for construction or production. Although AM is still useful for fabricating prototypes, the name rapid prototyping is becoming a misnomer. Many of the structures fabricated by AM are now the finished products. Now capable are structures composed of fused metals [15], tough polymers [5], and ceramics [16]. Of particular interest to this thesis are biomedical applications of AM. The following subsections will provide a cursory overview of successful applications of AM to a variety of biomedical needs.

1.1.1 Biomedical Applications

Conventionally, synthetic products for biomedical applications came in a one-size-fits-all format with built in adjustability. A common example are crutches that have an adjustable length to fit a range of patient heights. More sophisticated examples are

allogenic and synthetic bone substitutes which come in a range of stock sizes and are then fashioned into the defect shape in the surgical theater [17]. AM processes are well suited to bring a design and manufacturing cycle that is completely tailored to the patient, as opposed to the traditional one-size-fits-all design paradigm. We and others [18] envision design on more than just the product fit level. We believe that the real power of AM process lies in the ability to tailor the complete product to the patient's needs; this involves designs in which the product's mechanical strength, surface and internal features, materials composition, and local chemistry are all tailored to the specific patient anatomy and factors such as age, gender, race, and pathology.

The first application of AM processes to biomedicine was the fabrication of mock-ups to aid surgical procedures [15]. Just as architects have used AM to explore building designs, the medical community has used AM to build anatomical replicas for training and visualization. Although this is useful, our primary interest is synthetic tissue grafts, also termed synthetic tissue scaffolds. Synthetic tissue scaffolds have been defined by Drury et al. to be "synthetic extracellular matri(ces) to organize cells into a three-dimensional architecture and present stimuli, which direct the growth and formation of a desired tissue [19]." Fig. 1.1 demonstrates a variety of different synthetic scaffolds fabricated by AM methods. Fig. 1.1f demonstrates a prosthetic to show a diversity of applications, but it is not a scaffold. We demonstrate this particular application to highlight that AM and biomedicine have merged with fashion. This particular company (Bespoke Innovations) fabricates custom designed prosthesis, many with exotic shapes and surface finishes such as leather and corrugated stainless steel cladding [New York Times]. Fig. 1.1 is not an exhaustive compilation, but demonstrates a diversity of methods and scaffold uses.

Four of the AM processes in Fig. 1.1 are diagrammed in Fig. 1.2. This thesis details a specific form of micronozzle extrusion and the details of this process will be given in Chapter 4. Complete details of each process can be found in [6] and [15]. A

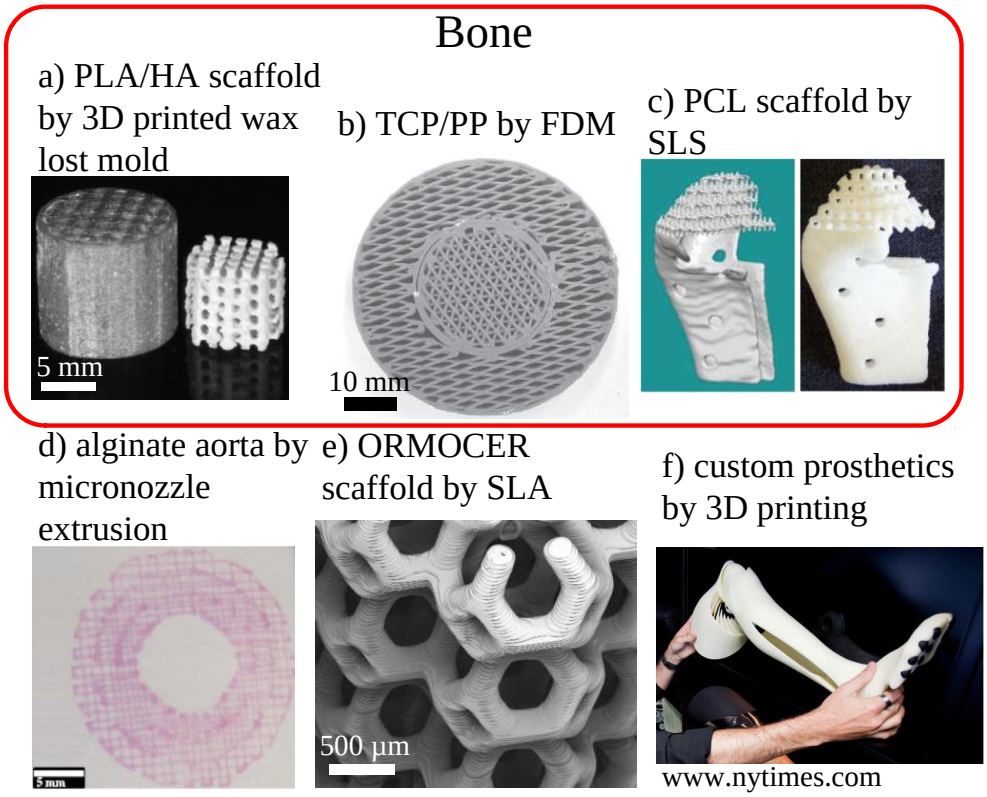


Figure 1.1: Representative examples of biomedical devices fabricated by AM. (a)-(e) are scaffolds. (f) is a custom prosthetic device. (a) Poly(L)lactide / Hydroxyapatite (PLA/HA) bone scaffold fabricated by casting a PLA/HA slurry into a lost wax mold fabricated by 3D printing [1]. (b) Tricalcium Phosphate/Polypropylene (TCP/PP) bone scaffold fabricated by Fused Deposition Modeling (FDM) where the polymer PP phase melts to flow through a heated FDM nozzle [2]. (c) Polycaprolactone (PCL) mandibular condyle scaffold fabricated by Selective Laser Sintering (SLS) [3]. (d) Alginate aorta cross section fabricated by micronozzle extrusion [4]. (e) Organically Modified Ceramics (ORMOCER) scaffold fabricated by a Stereolithography Apparatus (SLA) [5]. (f) Custom prosthetic fabricated by 3D printing; here, form and style are the central considerations as designs come in many exotic shapes and surface finishes.

brief description of each is given below:

- **Fused Deposition Modeling (FDM).** FDM uses heated nozzle to melt a polymeric feedstock material. The molten polymer is extruded through the nozzle and immediately sets once exposed to the lower temperature chamber

outside the nozzle. Sacrificial polymers (support material in Fig. 1.2) are integrated into the build routine to support overhanging features. Sacrificial materials are either water soluble or can be fractured away from the build material for post-process removal.

- **3D Printing.** A bed of ceramic or polymeric powder is rolled out in a thin even layer into a fabrication bay. A print head, similar to a common ink-jet print head, prints a binder to consolidate powders on a 2D plane. At each subsequent layer, the bed of powder is lowered and a new 2D plane is adhered to the previous layer with the binder to create 3D structures. The powder bed supports overhanging features during the build routine. Bound particles can be used as is, or heat treated to strengthen the structure.

- **Selective Laser Sintering (SLS).** Similar to 3D Printing, a thin layer of powder is applied to a fabrication bay. A laser scans a 2D shape to sinter powder particles together or adhere interstitial binder particles. At each subsequent layer, the powder bed lowers and a new layer is added. Excess powder supports overhanging features.

- **Stereolithography Apparatus (SLA).** At each layer, a sweeper smooths the surface of a vat of photopolymer and then a laser scans a 2D feature in the plane, curing the exposed surface. With each subsequent layer, the platform lowers deeper into the vat and a new layer is applied to build 3D structures. The photopolymer can be weakly adhered by reducing the laser intensity and exposure time to build a removable support structure.

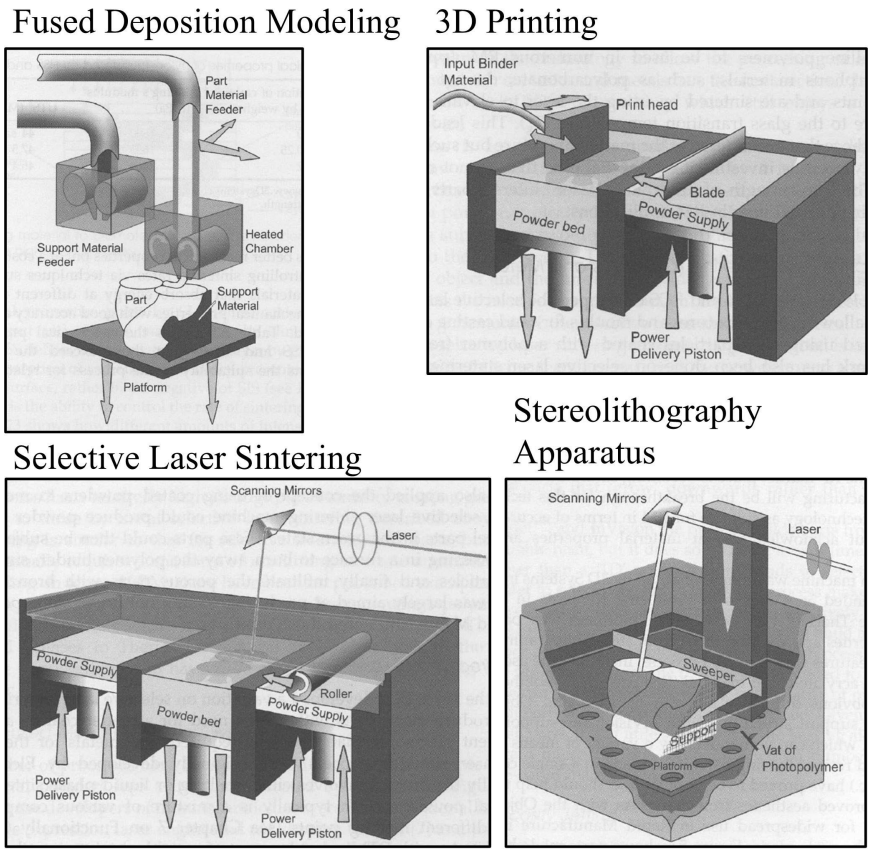


Figure 1.2: Diagrams of four AM processes that are commonly used for synthetic scaffold manufacturing. All images are from [6].

1.2 Manufacturing Synthetic Bone Grafts

1.2.1 Motivation

Patients with a large skeletal defects ($> 5\ mm$) typically receive a graft of autologous bone, most commonly harvested from the iliac crest of their hip or from their ribs [21]. This procedure is termed an autograft and the success rate for complete integration of the donor bone in the host site is high. However, complications at the donor site are high as well; complications ranging from nerve injury, infection, and chronic pain are reported in 8.5 – 20 % patients [20]. Additionally, autologous bone has a limited supply as there are only a select few candidate donors sites within the skeleton [21]. Ideally, doctors and patients would have viable graft alternatives that eliminate this

painful and expensive harvest surgery and be virtually unlimited in supply.

The impetus for creating an artificial bone graft that circumvents the harvest surgery is becoming stronger. At 1.5 million graft procedures per year in the United States [22], bone grafts are the second most transplanted tissue, second to blood transfusions [20]. The demographic that is in most in need of a successful artificial bone graft is the elderly [17] and the number of elderly individuals is expected to double over the next 25 years [23]. The market for synthetic bone grafts is expected to grow at an annual rate of 9.6% from 2006 to 2013; highlighting both the clinical need for bone graft substitutes and significant economic potential [17].

A bone scaffold must satisfy two requirements to be viable *in vivo* [20]:

- **Biocompatible.** Scaffold materials must be biocompatible to prevent an autoimmune response that would lead to scaffold rejection.

- **Osteoconductive.** Scaffolds must allow for, and even facilitate, the formation of new bone and the necessary vasculature for nutrient and waste transport. Within the umbrella of osteoconductivity is the need for scaffolds to be porous, to enhance permeability for cell, nutrient, and waste transport and for tissue ingrowth.

Both of these considerations guide scaffold material selection and structural design. Additionally, it is beneficial for the scaffold to be osteoinductive, meaning that the scaffold stimulates or activates local mesenchymal stem cells to differentiate into bone forming osteoblasts to grow bone within the synthetic structure. Scaffolds are either innately osteoinductive because the material precipitates a biological layer that drives the appropriate biological species inside the scaffold or are modified to be osteoinductive by using growth factors [20]. The design and manufacture of scaffold porosity is of critical importance to this work as it affects osteoconductivity and it may affect osteoinductivity [8].

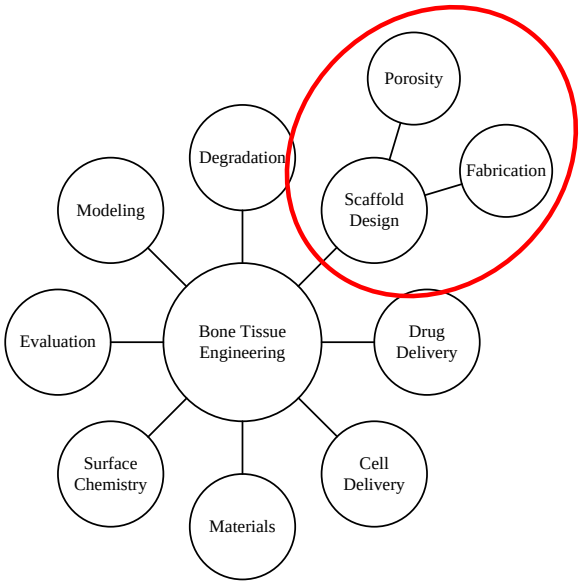


Figure 1.3: Incomplete map of the research directions in bone tissue engineering. This thesis focuses on scaffold design and manufacturing problems.

The need for synthetic bone scaffolds has been well articulated by many researchers [18, 20, 24]. Despite the recognized need, translation from the research environment to clinical application has been slow [18]. A very good reason why clinicians have not readily adopted artificial bone graft is that **there does not exist a synthetic bone graft that matches the osteointegrability of the autograft [25]**. A major 'bottleneck' in synthetic bone scaffold progress towards clinical translation is that tissue at the center of scaffolds implanted into large defects suffers from high necrosis rates because the inner scaffold regions are distant from the blood supply [26]. Bone tissue engineering researchers have investigated the development of a viable artificial implant from a variety of different approaches; an incomplete map of these approaches is shown in Fig. 1.3. We recognize the importance of the different research approaches, however this thesis investigates artificial bone scaffolds solely from the scaffold design and fabrication approach.

1.2.2 Scaffold Design

It is well known that mechanical properties and porosity are important considerations for scaffold design in tissue engineering [18, 24]. For bone tissue engineering, a few requisite designed properties have been established. Namely, the scaffold must be as stiff as native bone or slightly less stiff [3] and the scaffold must contain interconnected porosity throughout to facilitate blood vessel infiltration and bone cell migration. Pore interconnection size must be on the order of 100 to 1000 μm [24], termed macroporosity here. Within this range there is not a definitive optimal design point, provided that the pore volume percentage exceeds 60 $vol\%$ [24].

Outside of these basic requisite bone scaffold characteristics, there is both flexibility in design options and uncertainty as to what an optimal bone scaffold design is. Bone scaffold design can be broken down into three independent design problems, Fig. 1.4. We term the three-dimensional space that the bone scaffold occupies the scaffold envelope; the envelope typically will be specified by the patient defect shape and any mounting features needed to affix the scaffold to the defect site. Within the scaffold envelope, the scaffold must have interconnected macroporosity, introduced in the previous paragraph. Methods to manufacture scaffolds with a designed envelope and macroporosity are the most commonly reported bone scaffold design studies.

There is an additional design space that warrants a complete investigation. Interconnected porosity throughout the build material on the order of 2 to 20 μm, termed microporosity here, has recently received attention as an important design space to consider. See the images in Fig. 1.5 for examples of the micro- and macropore spaces in a basic scaffold with a cylindrical envelope. *In vivo* studies comparing relative levels of microporosity demonstrate that increasing the microporosity pore fraction positively influences bone ingrowth into the macropore space, significantly increasing bone growth [27]. Recent research [8,9] exhibits two interesting effects of microporos-

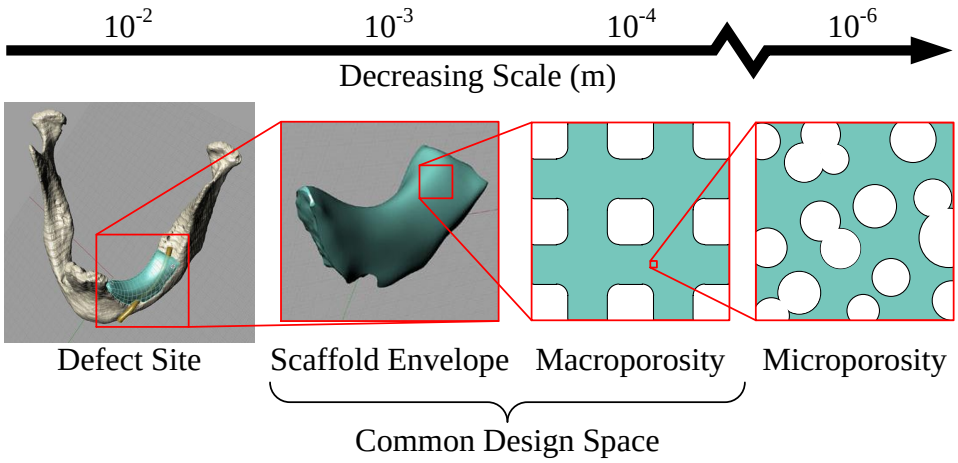


Figure 1.4: Schematic of the three independent design problems for bone scaffold design. Defect site and scaffold envelope images are from [7].

ity, Fig. 1.6. The first is that the micropore space, despite the pore interconnection size being on the order of osteoblasts, is able to support bone growth, as shown in Fig. 1.6a. Instead of a discrete interface between bone and scaffold, the bone infiltrates the interconnected network of microporosity. Even at three weeks after implantation, the scaffold is a fully integrated composite of natural and artificial tissue [8]. The second interesting effect of microporosity is that scaffolds with microporosity yield a more uniform distribution of bone in the macropores, as shown in Fig. 1.6b. In contrast, scaffolds without micropores preferentially form bone at the defect periphery, near the native bone [9]. This result is important when considering large defect repair; large defect grafts have a high failure rate because a non-uniform bone ingrowth profile leaves the central regions devoid of new bone and susceptible to failure [26].

1.2.3 Thesis Scope and Central Aim

In this thesis, we will make no attempt to claim what is the optimal macroporous and microporous design for human bone replacement. In fact, the scaffold parameter space is vast and the variability between patients is so great – patients vary by gender, age, race, and pathology – that it would be impossible to do so. Instead, we recognize

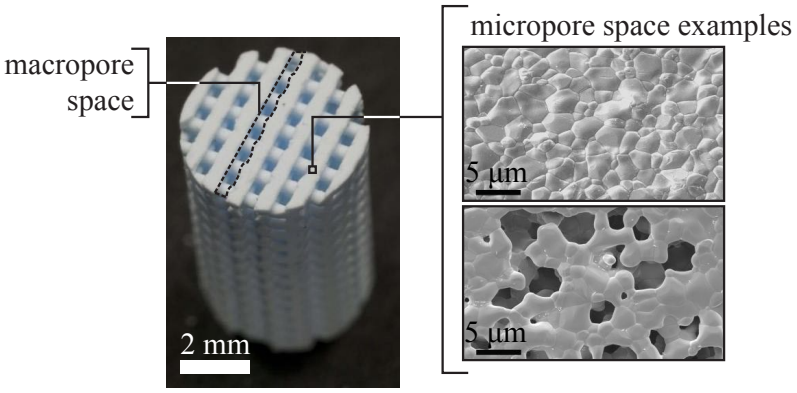


Figure 1.5: Scaffold design demonstrating the macropore and micropore space. This example was fabricated by a nozzle-based AM method.

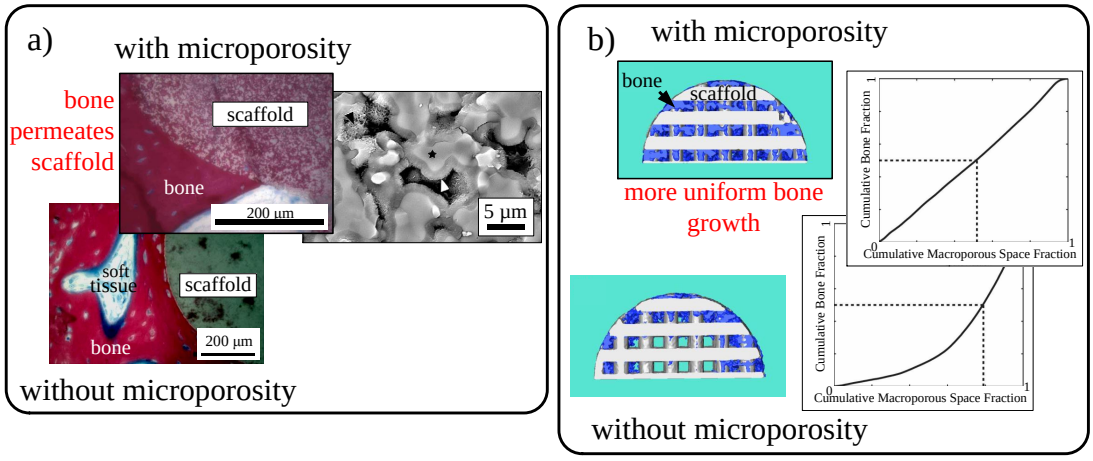


Figure 1.6: Recent results from the Wagoner Johnson group that highlight the importance of incorporating microporosity within a scaffold design. a) Scaffolds with microporosity induce natural bone to penetrate the scaffold structure, completely filling the tightly constricted and tortuous micropore space [8]. Histology slides demonstrate areas of new bone growth, synthetic scaffold, and soft tissue. The textured surface of new bone growth (white arrow) deposited on the scaffold micropore surface (black star) is clearly shown in the Scanning Electron Micrograph (SEM) of the fractured internal microstructure of a scaffold. b) Scaffolds with microporosity elicit a more uniform bone ingrowth profile [9]. New bone, represented in blue in the reconstructed image of micro-Computed Tomography (μCT) data, is more uniformly distributed in the scaffold with microporosity. Processed μCT data quantitatively demonstrates the more uniform bone ingrowth profile of microporous scaffolds; an ideal, perfectly uniform bone ingrowth profile will have a plot with a slope of 1 throughout the plot range.

that there are many researchers who would like to test macroporosity and microporosity treatments in *in vivo* studies and evaluate their effectiveness. However, there do not exist manufacturing methods that enable complete design flexibility. **We aim to develop a scaffold manufacturing method that enables the fabrication of scaffolds with a fully flexible design space at the envelope and macroporosity level, and enables multiple regions of different microporosities to be integrated within a single scaffold.**

We will, however, focus our investigation to Calcium Phosphate (CaP) scaffolds. Many polymeric scaffolds have toxic degradation products and uncertain degradation rates and therefore are arguably sub-optimal for bone tissue regeneration [28, 29]. Also, many polymers have a compressive strength that is significantly lower than that of native bone and consequently are not appropriate for load bearing sites [30]. CaP ceramics have properties that make them advantageous for bone replacement. CaP scaffolds can be fabricated to match the modulus of native bone, they have a slow degradation rate so there is a reduced risk of mechanical instability, and the degradation products create a mineral rich local environment that is conducive to bone remodeling [24].

1.2.4 Manufacturing System Rationale

Only a select few scaffold manufacturing platforms are capable of integrating more than one microstructure within a single structure. Lost-mold, electrospinning, and gas foaming methods are bulk processes that create randomly distributed pores throughout and are therefore unable to isolate porosity designs to distinct domains [21]. SLA, SLS, and 3D Printing methods can strictly define the envelope and macrostructure, however these method are incapable of interchanging build materials and are therefore limited to a single microporous composition [3]. To the best of the authors' knowledge, nozzle-based AM systems are the only manufacturing platforms that can

strictly control the macrostructure [4] and integrate multiple materials with different microstructures or chemistries. For scaffolds fabricated by nozzle-based AM methods, macroporosity is generally a function of *material placement* and microporosity a function of *material composition* where the microporosity is randomly distributed. Within the set of nozzle-based AM methods, polymeric systems such as FDM can readily integrate multiple materials within a single scaffold. FDM requires a build material phases transition; a few works have demonstrated the FDM fabrication of bone scaffolds where the polymeric build material is loaded with CaP ceramic particles [2]. However, these scaffolds contain a large quantity of polymeric material with little utility for scaffold integration.

The number of AM technologies available for ceramics is not as extensive as for polymers. Monolithic structures, defined here as structures with a uniform porosity distribution and composed of a single material, are readily achievable [31–34]. When considering CaP scaffolds, nozzle-based AM methods have not displayed the material placement dexterity needed to precisely define different regions of varying compositions. The most advanced CaP structures fabricated to date are demonstrated in [32] where multiple macroporous regions were defined with a single material composition.

An established nozzle-based AM technology that has demonstrated the ability to construct *monolithic* CaP scaffolds with low polymeric binder content ($\approx$1%) is micro-Robotic Deposition (μRD) [35]. This work elevates μRD above just *monolithic* capabilities by providing a method which enables precise control over material placement, where the ability to interchange material compositions with low material placement accuracy is already an established functionality.

Accurately extruding the build material is difficult because the material flowrate response is nonlinear with a time delay [36]. Current control methods either include extraneous lead-in lines to allow for a fully developed flow for monolithic scaffolds or *ad hoc* methods have been developed to gain some extrusion accuracy. Compound-

ing the extrusion performance problem is that no sensors are available for realtime flowrate feedback. Given a lack of process control and measurement, we have developed a control method that automatically learns the best input signal to achieve a desired extrusion flowrate response. This open-loop control method is termed the Basis Task Approach to Iterative Learning Control (BTILC) and it is integrated with a machine vision system to measure extrusion flowrate.

The ability to strictly place materials with different compositions considerably expands scaffold design options. Researchers can now explore structures with graded microstructures, integrated structures of different CaP species (e.g. integrated regions of hydroxyapatite (HA) and β-Tricalcium Phosphate (β-TCP) compositions), local composition anisotropies, near-net shape scaffold envelopes, and guided bone growth by exploiting the ability of highly-microporous compositions to drive new bone growth.

1.3 Thesis Organization

This thesis merges the two fairly disparate fields of tissue engineering and robotic control and automation. Consequently, the content and tone of presentation changes considerably depending on the chapter. Chapter 2 sets up the base Iterative Learning Control (ILC) method, providing important algorithm details and considerations and provides a brief review of ILC adaptations that have similarities to the method introduced here. From the ILC introduction, Chapter 3 introduces our adaptation of ILC, termed the Basis Task Approach to Iterative Learning Control (BTILC), and gives important controls based definitions and performance considerations. Once ILC and BTILC have been established, the thesis transitions to scaffold manufacturing. Chapter 4 details the scaffold fabrication systems, μRD and provides an overview of the general scaffold fabrication process. Chapter 5 merges Chapters 3 and 4, demon-

strating BTILC being directly applied to the μRD process. A few basic structures are fabricated to elucidate process details and capabilities. Chapter 6 culminates the scaffold manufacturing content, demonstrating the manufacture of two sets of advanced architecture scaffolds: 1) combinatorial test samples designed to efficiently evaluate multiple macroporosity and microporosity domains within a single structure and 2) near-net shape scaffolds that include contoured overhanging features, integrated heterogeneous materials, and hollow cavities. Bone scaffolds designs derived directly from skeletal defects will have all of these features, therefore the successful fabrication of these features demonstrates potential utility of our process for biomimetic design fabrication. Chapter 8 provides concluding statements and future work. The Appendices provide complete details of the processes, equipment, and computer code used in this work and are designed to aid researchers who are building on this work.

Chapter 2

Iterative Learning Control

2.1 System Setup

For all discussions of Iterative Learning Control (ILC), we will consider a physical plant represented by the single-input single-output (SISO) linear time-invariant (LTI) operator, H_d, in discrete-time state-space form:

$$H_d : \begin{cases} x_d(k+1) = \mathbf{A}_d x_d(k) + \mathbf{B}_d u_d(k) + w(k) \\ y_d(k) = \mathbf{C}_d x_d(k) + v(k) \end{cases} . \tag{2.1}$$

$y_d(k) \in \mathbb{R}^K$, $u_d(k) \in \mathbb{R}^K$, and $v(k) \in \mathbb{R}^K$ where K is the signal length and y, u, and v are the measured output, system input, and signal noise, respectively. $\mathbf{A}_d \in \mathbb{R}^{\rho \times \rho}$, $\mathbf{B}_d \in \mathbb{R}^{\rho \times 1}$, $\mathbf{C}_d \in \mathbb{R}^{1 \times \rho}$, and $\mathbf{D}_d \in \mathbb{R}^{1 \times 1}$ are appropriately sized state-space system matrices and $x_d(k) \in \mathbb{R}^{\rho \times K}$ is a vector of the operator states where ρ is the number of states, and likewise, $w(k) \in \mathbb{R}^{\rho \times K}$ is the noise on the input channel that is propagated to the states. ILC is extendable to multi-input multi-output (MIMO) systems [37], however this thesis addresses SISO systems solely to focus the discussion on the types of physical systems we will control. H_d can be either an open-loop stable plant or the plant sensitivity function of a stabilizable plant with stabilizing feedback; $\max_i |\lambda_i| < 1$ where λ_i is an eigenvalue of $\mathbf{A}_d$. k is the discrete-time index. d is the configuration index and indicates a specific physical system. Subscript d will sometimes be omitted

to de-emphasize the specificity of a particular physical system and the equation will hold true for a general system.

2.2 Introduction to Iterative Learning Control (ILC)

An excellent review of Iterative Learning Control (ILC) can be found in [10]. This section provides the salient information from the survey and elaborates on flexible adaptations of ILC. ILC is an appealing control algorithm because it does not require rigorous system modeling and identification to achieve high performance trajectory tracking [11]. Given a system with:

- a repeated reference trajectory
- discontinuous operation
- trial-invariant dynamics
- and trial-invariant initial conditions

ILC can be applied to exploit trajectory repetition to compensate for unmodeled dynamics, nonlinearities, and repeated disturbances [10]. The above system characteristics make ILC particularly appealing to mass manufacturing applications where repeated reference trajectories are inherent. ILC can be applied independently to open loop stable systems or appending feedback control in open loop stable and unstable systems, Fig. 2.1. In general, ILC achieves more accurate reference tracking than well designed feedforward and feedback controllers. However, specific constraints must be satisfied to apply ILC and therefore not all control objectives are appropriate.

ILC is a memory-based control algorithm. At each iteration, the input signal from the previous iteration, $u^j(k)$, and the error signal from the previous iteration, $e^j(k)$,

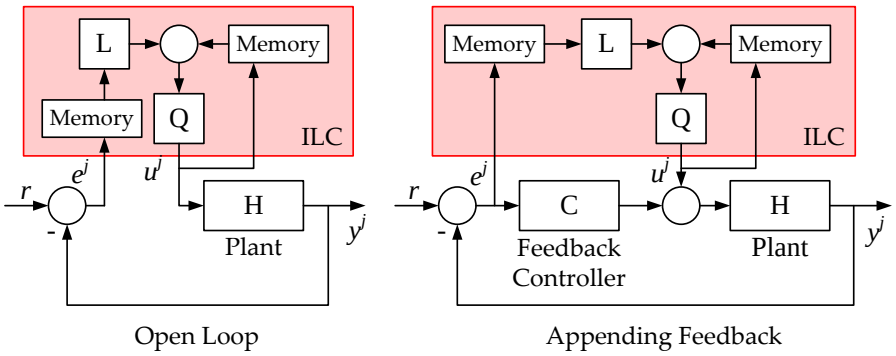


Figure 2.1: Block diagram of the two most commons configurations for ILC.

are used to calculate an updated input signal, $u^{j+1}(k)$, using the typical update law in (2.2).

$$u^{j+1}(k) = Q(q)\left[u^j(k) + L(q)e^j(k+1)\right] \tag{2.2}$$

Error is defined as the iteration-invariant desired reference trajectory, $r(k)$, minus the measured output, $y^j(k)$, $e^j(k) = r(k) - y^j(k)$, where j is the iteration number index. Please note that the chosen notation does deviate from ILC convention. Chapter 3 will introduce additional classifiers in the signal subscript and the notation $e^j(k)$ will be cleaner than the conventional $e_j(k)$. q is the typical forward time-shift operator, $qx(k) \equiv x(k+1)$, and $L(q)$ and $Q(q)$ are termed the learning filter and Q-filter, respectively. Given appropriately designed $L(q)$ and $Q(q)$ filters, trial error will decrease with each subsequent iteration. The inputs to the ILC update law, and a characterization of the iteration-to-iteration performance improvement, is given schematically in Fig. 2.2. It should be noted that the update law in (2.2) is the most commonly used update law, however there are alternatives [38]. All update law options use the premise that previous input signals and the resultant error signals can be used to iteratively improve performance. Common options for the learning filter, Q-filter, and update law can be found in [10].

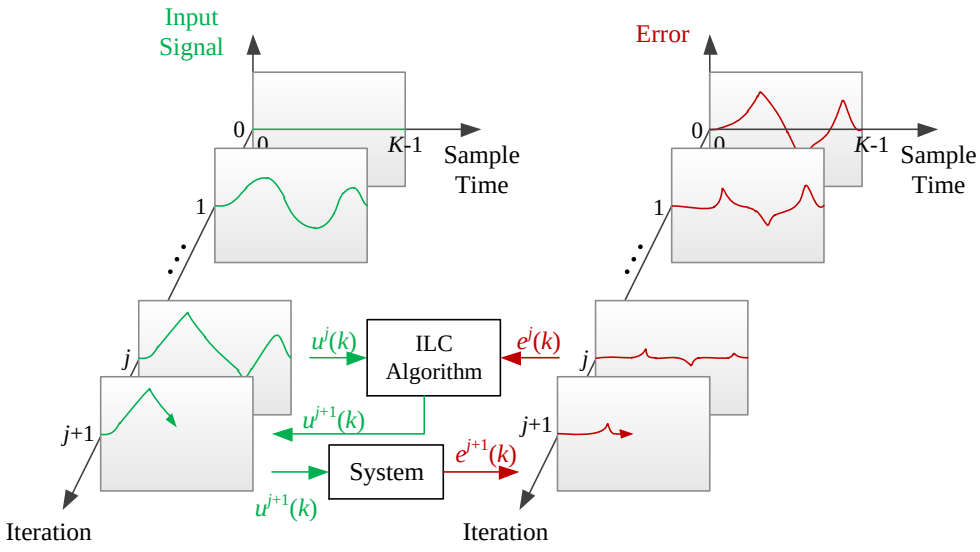


Figure 2.2: Schematic of the ILC update law. Given a stable system, at each iteration the previous input signal, $u^j(k)$, and resultant error signal, $e^j(k)$, is used to calculate an updated input signal, $u^{j+1}(k)$, to apply at the next iteration. With proper design, the error signal will decrease with each subsequent iteration. Adapted from [10] with permission.

There are two different system representations for ILC design; time-domain in the lifted-system framework and in the frequency domain. In the lifted-system framework, each K sample signal is represented in vector form, e.g.:

$$\mathbf{y}^j = \begin{bmatrix} y^j(1) & y^j(2) & \cdots & y^j(K) \end{bmatrix}^T. \tag{2.3}$$

H_d, (2.1), can likewise be represented in lifted form:

$$\underbrace{\begin{bmatrix} y^j(1) \\ y^j(2) \\ \vdots \\ y^j(K) \end{bmatrix}}_{\mathbf{y}^j} = \underbrace{\begin{bmatrix} h(1) & 0 & \cdots & 0 \\ h(2) & h(1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ h(K) & h(K-1) & \cdots & h(1) \end{bmatrix}}_{\mathbf{H}} \underbrace{\begin{bmatrix} u^j(0) \\ u^j(1) \\ \vdots \\ u^j(K-1) \end{bmatrix}}_{\mathbf{u}^j} + \underbrace{\begin{bmatrix} d(1) \\ d(2) \\ \vdots \\ d(K) \end{bmatrix}}_{\mathbf{d}} \tag{2.4}$$

where $h(i) = \mathbf{CA}^{i-1}\mathbf{B}$ and $w(k)$ and $v(k)$ have been lumped into a single disturbance vector, $\mathbf{d}$. The update law in (2.2) is written in the lifted-system framework using similar matrix manipulations. An LTI H_d, $L(q)$, and $Q(q)$ with yield matrices $\mathbf{H}$, $\mathbf{L}$, and $\mathbf{Q}$ that have Toeplitz structures, meaning that the entries along the diagonal are the same. $\mathbf{H}$ will be lower-triangular because only causal plants are considered. $\mathbf{L}$ and $\mathbf{Q}$ need not be causal and therefore can have upper-triangular entries. The frequency domain representations of the update law, (2.2), and plant, (2.1), are constructed using basic frequency domain operations:

$$\begin{aligned} Y^j(z) &= H(z)U^j(z) + D(z) \\ U^{j+1}(z) &= Q(z)\left[U^j(z) + zL(z)E^j(z)\right] \end{aligned}. \tag{2.5}$$

The two most important criteria for ILC design are stability and monotonic convergence. The update law / plant system in (2.2) / (2.4) is asymptotically stable if there exists a bounded $\bar{u} \in \mathbb{R}$ such that:

$$\begin{aligned} &|u^j(k)| \leq \bar{u} \text{ for all } k = \{0, \ldots, K-1\} \text{ and } j = \{0, 1, \ldots\} \\ &\text{and for all } k \in \{0, \ldots, K-1\} \lim_{j \to \infty} u^j(k) \text{ exists} \end{aligned}. \tag{2.6}$$

Stability criteria (2.6) is assessed by the writing the recursions:

$$\mathbf{u}^{j+1} = \mathbf{Q}\left(\mathbf{I} - \mathbf{LH}\right)\mathbf{u}^{j} + \mathbf{QL}\left(\mathbf{r} - \mathbf{d}\right)$$

or (2.7)

$$U^{j+1}(z) = Q(z)\left[1 - zL(z)H(z)\right]U^{j}(z) + zQ(z)L(z)\left[R(z) - D(z)\right]$$

for the lifted-system or frequency domain, respectively. The right most portion of (2.7) is bounded for a stable learning and Q-filter. The recursive portion is stable for:

$$\rho\left(\mathbf{Q}\left(\mathbf{I} - \mathbf{LH}\right)\right) < 1. \tag{2.8}$$

for the lifted-system analysis. For the frequency-domain analysis, we only have an approximate stability criterion:

$$\left\|Q(z)\left[1 - zL(z)H(z)\right]\right\|_{\infty} < 1 \tag{2.9}$$

where stability is guaranteed for infinite duration processes, $K = \infty$, which is impossible for finite duration manufacturing processes. Because the frequency-domain analysis is not exact, control designers must be conservative with their designs to prevent an unstable response from either unmodeled dynamics or an unstable plant / update law combined system that is not identifiable from the stability criteria in (2.8)

Stable ILC systems can result in a behavior in which the error transiently increases before converging to a small error and this behavior is generally considered to be unfavorable. An ILC system is defined as monotonically convergent if:

$$\left\|\mathbf{e}^{\infty} - \mathbf{e}^{j+1}\right\| \leq \gamma \left\|\mathbf{e}^{\infty} - \mathbf{e}^{j}\right\| \tag{2.10}$$

for a chosen norm. That is, the error signal gets closer to the infinite iteration error, in terms of a norm, with each increasing iteration. This condition is guaranteed in the lifted-system framework for the L_2-norm and in the frequency-domain with $K = \infty$ if:

$$\begin{aligned} &\bar{\sigma}\left(\mathbf{HQ}\left(\mathbf{I}-\mathbf{LH}\right)\mathbf{H}^{-1}\right) < 1 \\ &\text{or} \qquad\qquad , \\ &\left\|Q(z)\left[1-zL(z)H(z)\right]\right\|_{\infty} < 1 \end{aligned} \tag{2.11}$$

respectively. The fact that an exact stability criterion, (2.8), and monotonicity criterion, (2.11), can be evaluated in the lifted-system analysis is a compelling advantage. However, the dimension of the matrices scale quadratically with the signal length and the calculation of the matrix inverse in (2.11) becomes computationally intractable in long duration processes [39]. Therefore there is a tradeoff between the utility of analysis methods; frequency domain methods provide ease of use but there are not exact rules for stability and monotonicity while the lifted-system format is precise yet unwieldy.

Given the ideal situation in which: 1) the disturbance, $d(k)$, is iteration-invariant; 2) a Q-filter is chosen in such that $Q(q) = 1$; and 3) $L(q)$ is designed such that (2.8) holds, there will be an infinite iteration error that is identically zero. That is, $e^{\infty}(k) = 0$ for all $k \in \{1, \ldots, K\}$. This ideal situation never happens in practice and it is impossible to achieve zero-error tracking. In practice the disturbance is iteration-varying and can be decomposed into a repeating part and a non-repeating part:

$$d^j(k) = \bar{d}(k) + \tilde{d}^j(k). \tag{2.12}$$

In general, the repeating disturbance, $\bar{d}(k)$, contains mostly low-frequency signal content and the non-repeating disturbance, $\tilde{d}^j(k)$, contains mostly high-frequency sig-

nal content. The non-repeating disturbance will propagate from iteration-to-iteration for designs when $Q(q) = 1$, therefore many practical applications choose $Q(q)$ to be a low-pass filter that attenuates (thereby ignoring) non-repeating disturbances and passes (thereby learning) repeated disturbances. $Q(q)$ is designed by essentially balancing converged error performance with robustness. Performance is improved by increasing Q-filter bandwidth, however the propagation of non-repeating disturbances, which causes undesirable transients, is attenuated by lowering Q-filter bandwidth. Interesting new tools are being developed to analyze and design the Q-filter, but conventional practice has been to lower the Q-filter bandwidth whenever there is an indication that the system is not robust [40]. $L(q)$ is designed by a number of different methods, with the most popular being proportional-derivative designs (analogous to proportional-derivative feedback controller designs), model-inversion designs (analogous to dead-beat controller designs), and norm optimal designs (analogous to linear-quadratic-regulator designs) [10].

2.3 Trajectory Flexibility

ILC is particularly appealing to mass manufacturing applications where repeated reference trajectories and consistent dynamics are inherent. However, the application of ILC is limited in that if the desired reference trajectory or the system dynamics vary, the input signal from the previous iteration becomes invalid and the ILC algorithm must be reinitiated. Section 1.1 provided examples of flexible AM systems where the process and therefore the process trajectories change from run-to-run. As cost-saving industrial technologies such as flexible [41] and AM [6] systems become more prevalent in industry, trajectory and system specific ILC algorithms become less powerful. Instead, flexible adaptations of ILC need to be explored.

The need for trajectory flexibility in the ILC algorithm has been stressed previ-

ously [10, 11]. Many of the previous attempts to weaken the trajectory invariance constraint have deemphasized the time specificity of ILC, instead focusing on the trajectory dynamics and the input signals to achieve an output signal that accurately tracks the reference trajectory. This section highlights a few important examples in the literature. All flexible ILC methods use a two-step process; first a database of information is developed in a training routine and then second the database is used in an intelligent manner to track a new trajectory that is not constrained to be identical to one of the training trajectories.

2.3.1 Inverse Dynamics Methods

Integral Transformations

Messner and Horowitz published a series of papers in the early 1990's that detailed a method for adding flexibility to ILC using what were termed integral transforms [11, 42]. The integral transform method in essence maps a set of dynamics to a control action. Given a new reference trajectory with a similar set of dynamics, a new control action can be generated by an estimate of an influence function. The influence function is the mapping from a signal to dynamic space:

$$\mathbf{c}(\cdot) : [0, K] \rightarrow \mathbb{R}^n \tag{2.13}$$

where $\mathbf{c}(\cdot)$ is the exact influence function and the dimension of the dynamic space is size n. An estimate for the correct control action, $\hat{w}(\cdot)$, is given by the integral transformation:

$$\hat{w}(\cdot) = \int_{\Gamma} K(\cdot, \gamma)\hat{\mathbf{c}}(k, \gamma)d\gamma \tag{2.14}$$

where Γ is the complete set of identified dynamics, $K(\cdot)$ is a kernel function, and

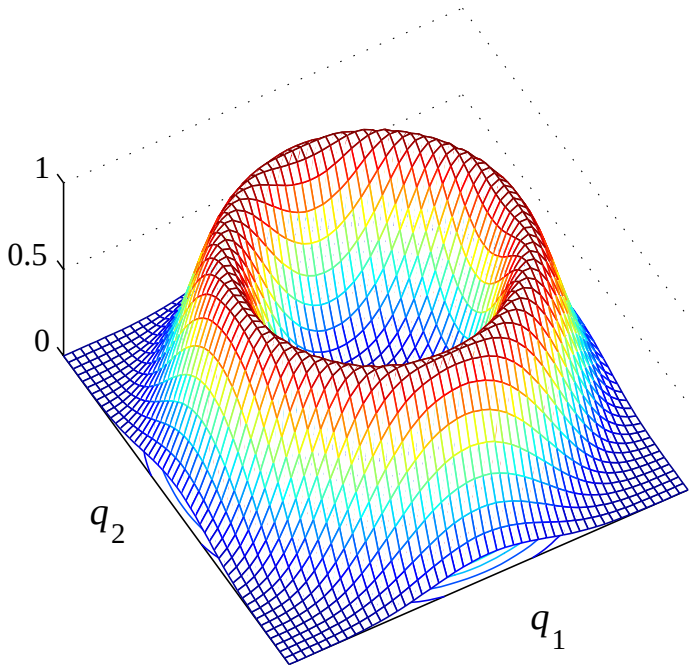


Figure 2.3: Demonstration of a Gaussian kernel about a circular trajectory in a 2D dynamic space, q_1 and q_2. The kernel has the most power about the circular trajectory and tails off as the new trajectory deviates from the circular trajectory. Figure motivated by the content of [11].

$\hat{\mathbf{c}}(\cdot)$ is an estimate of the influence function from a training routine. The authors select $K(\cdot)$ to be a standard Gaussian kernel function in their experiments. Fig. 2.3 demonstrates the principle of the integral transform method using a Gaussian kernel in 2D dynamic space. A given training routine will identify an influence function for the trajectory given by the manifold tracing the peak value of the plot. The Gaussian kernel extends the control action calculation to trajectories with similar dynamics. The further away from the training trajectory the new trajectory is, the less power the influence function will have on the control action.

The key experimental result displayed the ability to track a circular trajectory based on learning a spiral trajectory in a training routine, effectively demonstrating that integral transforms can be used to project the control action from one dynamic

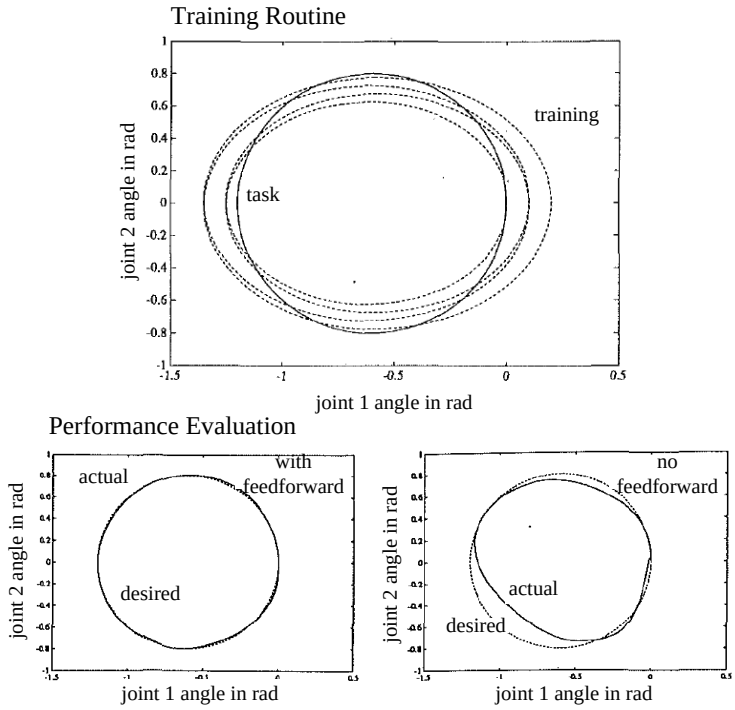


Figure 2.4: Experimental results from [11]. By using a spiral training trajectory and the integral transform method, the authors were able to track a circular trajectory with higher accuracy than if they did not use a feedforward signal. Figure has been modified from [11] for clarity of presentation.

set onto a new, but similar, dynamic set, Fig. 2.4.

Time Scale Transformations

Temporal flexibility has been researched by Sekimoto and colleagues [12, 43] using time-scale transformations in a framework that constrains the function that defines the reference trajectory, but permits tracking at arbitrary speeds; that is, a reference trajectory $\eta(t)$ can be scaled by a time-scale transformation, $\eta(f(t))$, but the base function must remain the same [12]. This method effectively allows the process to be either sped up or slowed down, however the trajectory must trace the same path each time. Time scale transformations use the system inverse dynamics to calculate a new input signal for a new tracking speed. The researchers demonstrate perfect tracking

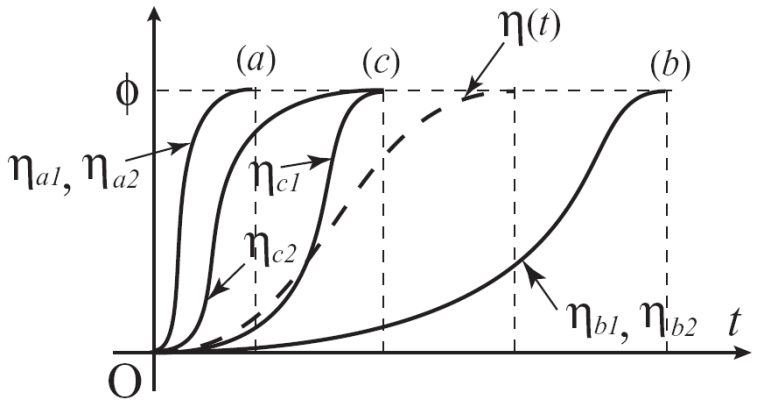


Figure 2.5: Scaled reference trajectories $\eta(t)$ using linear and non-linear time-scale transformation. Figure has been copied from [12].

of a new, time-scaled, reference trajectory in simulation.

2.3.2 Database Methods

Cataloged databases of input signals generated by ILC for a set of reference trajectories can be used in a flexible manner to be applied to a new reference trajectory with similar characteristics.

Best Fit Methods

Arif et al. used a simple algorithm to take a new reference trajectory and find the best fit of that trajectory in a database of reference trajectories [44]. The ILC input signal from the most similar trajectory is then used as iteration 1 for ILC on the new reference trajectory and they demonstrate that the system will converge more quickly than by not using the database, in simulation. Alsubaie et al. expanded upon this idea [45]. Instead of identifying the best fit trajectory within a database, they identified a linear combination of trajectories within a database, $\mathbf{R} = [r_1, r_2, \ldots, r_Q]$, that best fits a new trajectory, r_{new}, using least squares.

$$\begin{aligned} \min_{\mathbf{a}} \|e\|^2 &= \min_{\mathbf{a}} \|r_{new} - a_1 r_1 - a_2 r_2 - \cdots - a_Q r_Q\|^2 \\ \mathbf{a} &= \left(\mathbf{R}^T \mathbf{R}\right)^{-1} \mathbf{R}^T r_{new} \end{aligned} \tag{2.15}$$

The new input signal is then a linear combination of all the stored input signals:

$$u_{new} = [u_1, u_2, \ldots, u_Q]\, \mathbf{a}. \tag{2.16}$$

Experimental results demonstrate that using the database improves initial performance and that the more trajectories that are stored in the database, the more accurate the initial results, over the span of database sizes tested.

Point-to-point Task Based Methods

Gorinevsky et al. used a task-based approach to construct a new trajectory from simple point-to-point movements [46]. The two-link planar robotic system operating orthogonal to the gravitational force was assumed to be position invariant; that is a task was parameterized by the distance traversed in radians, not its initial and final position. Very little detail is provided on the designed form of the tasks or the validating experiments. Their approach essentially treated every task as its own trajectory tracking problem in which ILC could be used to learn a trajectory and then a sequence of trajectories with dwells in between could be tracked accurately. The ability to concatenate a series of tasks without dwells in between to construct more dynamic trajectories was not considered.

2.3.3 Discussion

Our goal is to develop a flexible adaptation of ILC that is directly in-line with common manufacturing practice. Many automatic manufacturing processes use an instruction based machine language, such as the pervasive G-code [47], in which a machine steps

sequentially through a list of tasks to complete a manufacturing operation. Our approach, introduced in the next chapter, is most similar Gorinevsky et al. [46] in that it decomposes complex trajectories into individual tasks. We term these *basis tasks*. What we will show is that the *basis tasks* can be used dynamically, directly transitioning from one *basis task* to another, and still achieve considerable improvement over systems operating solely under feedback. Section 3 will introduce our method, termed the Basis Task Approach to ILC (BTILC), and provide details on the *basis task* transitioning and performance. Chapter 7 will investigate the transitions in more detail. BTILC performance was evaluated extensively and the experimental results will be demonstrated in Chapters 5 – 7.

Chapter 3

Basis Task Approach to ILC

3.1 Introduction

Chapter 2 presents ILC and its tenets and weaknesses in depth. Of particular interests to AM systems are adaptations of ILC that can reap the benefits of learning based control while still maintaining the inherent flexibility of the manufacturing platforms. Section 2.3 provides important examples of previous efforts to enhance the flexibility of ILC, weakening the trajectory invariance constraint. The control scheme proposed here has a similar motivation as these previous attempts, however the method differs. While time specificity is deemphasized in many of the previous works, here we maintain the time specificity of ILC around a local event, or task, and use the superposition principle of linear systems to shift task occurrences and therefore alleviate trajectory constraints. This task centered framework gives a control scheme that is directly in-line with common manufacturing practice.

We term the proposed framework the *basis task* approach to ILC (BTILC). In a given manufacturing operation, we assume that there is a finite set of tasks desired, termed here as *basis tasks*, and a finite set of possible system dynamics, termed here as *configurations*. The set of *basis tasks* is termed the *operation space* and the set of *configurations* the *dynamic space*. Instead of learning the correct input signal by ILC each time the reference trajectory or system dynamics change, here we learn training trajectories that encompass each *basis task* in the *operation* and *dynamic space*. Therefore, the correct input signal for any manufacturing operation in the

operation and *dynamic space* is a concatenation of the *basis task* and *configuration* specific input signals appropriately shifted in time. This task oriented approach allows the operation trajectory to be arbitrarily chosen as dictated by task-oriented machine instructions or motion primitives, e.g. G-Code [47]. The compatibility with standard machine languages makes BTILC very flexible.

An additional benefit of this approach is that it alleviates trajectory length limitations encountered with computationally intensive ILC algorithms such as those using lifted system analysis [48] and data-rich sensors such as machine vision [36]. Improved computational efficiency is particularly important to AM processes which are typically on the order of minutes in duration, whereas many computationally intensive ILC algorithms are relegated to trajectories on the order of seconds in duration. For instance, a 30 *min* process would require the $\mathbf{H}$ matrix in (2.4) to be size $\mathbf{H} \in \mathbb{R}^{1.8mil \times 1.8mil}$ if norm-optimal ILC is utilized. Matrix computation at this size is computationally intractable on a conventional computer that would be interfaced with a manufacturing system. However, if the set of *basis tasks* could be identified with a *training set* reference signal on the order of seconds, more powerful algorithms such as the norm-optimal framework are feasible. ILC is applied to the shorter-in-duration training trajectories and the identified *basis signals* are coordinated appropriately for arbitrarily long operations.

Details of the approach are in following sections. Section 3.2 gives important definitions and system assumptions. A general application of BTILC is presented in Section 3.3. Importantly, this section provides considerations for logic design for *basis signal* extraction from the *basis signal library*. Section 3.4.2 concludes the chapter with stability and performance considerations.

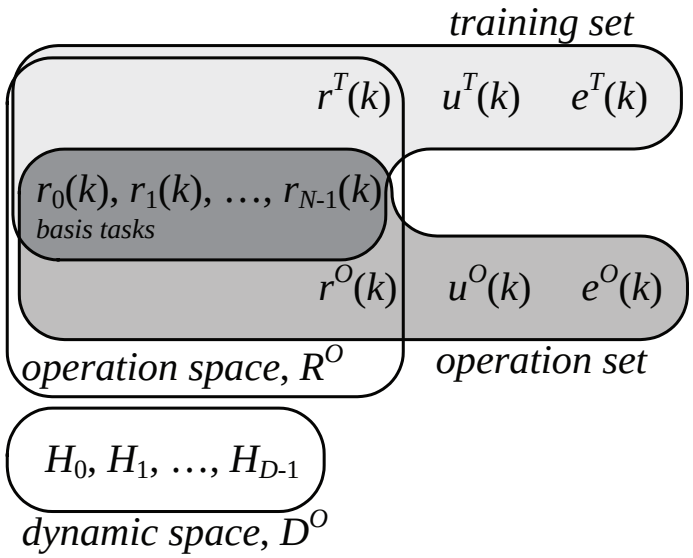


Figure 3.1: Signals and *configurations* contained in the *operation space*, *dynamic space*, *training set*, and *operation set*.

3.2 Problem Framework

A schematic of the spaces, sets, and signals in *Definitions 1 – 8* is given in Fig. 3.1.

3.2.1 Definitions

Definition 1 *Training and Operation Sets*

All signals belong to a specific set. The training set corresponds to signals applied, measured, or calculated during the training routine; denoted by the superscript T_p where p is the training routine index. In general, there are P training routines; $\{T_0, T_1, \ldots, T_{P-1}\}$. Iterations within the training set are denoted by adding the iteration index argument; e.g. $T(j)$. Likewise, the operation set corresponds to signals specific to the manufacturing operation; denoted by the superscript O. Superscripts are omitted for generic signals.

Definition 2 *Basis Task* $\triangleq r_n(k)$

Each basis task is defined by the reference signal designed to complete the given task, $r_n(k)$, where subscript n is the task index. In general, a given basis task, $r_n(k)$, is

defined on the domain $k \in [0, K_n - 1]$, where K_n is the discrete-time signal length of the n^{th} basis task. In the instance that the task is the regulation of a steady reference, $r_n(k)$ is a constant value, $\bar{r}_n$, in which $\bar{r}_n$ is repeated at the sampling interval for the duration of the regulation task.

Definition 3 $tspan\{r_n\} \triangleq \sum_{m_n=0}^{M_n-1} r_n(k - \Delta_{m_n}) \left(\mathfrak{s}(k - \Delta_{m_n}) - \mathfrak{s}(k - \Delta_{m_n} - K_n)\right)$

The conventional definition of span has been modified here for our purposes. The tspan, or task span, of a basis task, r_n, is the sum of M_n multiplicities of the basis task $r_n(k)$ where each multiplicity is shifted in time by Δ_{m_n}. Each Δ_{m_n} is unique based on its multiplicity index, m_n. $\mathfrak{s}(k)$ is the unit step function.

Definition 4 $R^O \triangleq tspan\left\{r_0(k), r_1(k), \ldots r_{N-1}(k)\right\}$

The operation space, R^O, is defined as the set of all basis tasks which constitute an operation where a given operation would be comprised of N basis tasks.

Definition 5
$$tbasis(R^O) \triangleq \{r_0(k), r_1(k), \ldots, r_{N-1}(k)\} \text{ if}$$
$$tspan\left\{r_0(k), r_1(k), \ldots, r_{N-1}(k)\right\} = R^O$$

Similar to Definition 4, the set of unique basis tasks $\{r_0(k), r_1(k), \ldots, r_{N-1}(k)\}$ is said to be a tbasis, or task basis, of R^O if R^O is comprised of basis tasks $\{r_0(k), r_1(k), \ldots, r_{N-1}(k)\}$.

Definition 6 $D^O \triangleq \{H_0, H_1, \ldots, H_{D-1}\}$

The dynamic space, D^O, is defined as the set of all configurations which are utilized in a manufacturing operation where the system dynamics for the d^{th} configuration is defined by the operator H_d, (2.1), and there are D configurations. Configurations of a different index need not be completely different systems, but possibly the same system operating under different conditions, e.g. a pick-and-place robot operating in either no-payload or carrying-payload conditions.

Definition 7 *Adjacency Matrix*

The sequence of basis tasks in either the training set or operation set is given by its binary adjacency matrix, $\mathbf{T} \in \mathbb{Z}_2^{N\times N}$ *or* $\mathbf{O} \in \mathbb{Z}_2^{N\times N}$*, respectively. Given an entire trajectory* $r(k) \in \mathbb{R}^K$*, a matrix entry* $c_{i,j} = 1$ *if* $\exists$ *a* $r(\Delta_{m_j} + K_j - 1)$ *and* $r(\Delta_{m_i})$ *such that:*

$$r(k) = \{\ldots, \underbrace{r_j(K_j - 1)}_{r(\Delta_{m_j}+K_j-1)}, \underbrace{r_i(0)}_{r(\Delta_{m_i})}, \ldots\} \tag{3.1}$$

where Δ_{m_n} *is the basis task transition time index at a given multiplicity of that task* m_n*. Else,* $c_{i,j} = 0$*. In words,* $c_{i,j} = 1$ *if there exists a transition from basis task* j *to* i*. An example of transitions in a generic set is given in the directed graph in Fig. 3.2 where each path will occur with probability 1 in the given set; that is the automaton gives a planned manufacturing operation where each instructed task will occur in sequence. The corresponding adjacency matrix for this example set is given in* (3.2)*.*

$$\mathbf{T} = \begin{bmatrix} 0 & 0 & 1 & \cdots & 0 \\ 1 & 0 & 0 & & 0 \\ 1 & 1 & 0 & & 0 \\ \vdots & & & \ddots & \\ 1 & 0 & 0 & & 0 \end{bmatrix}; \mathbf{O} = \begin{bmatrix} 0 & 0 & 1 & \cdots & 1 \\ 1 & 0 & 0 & & 0 \\ 0 & 1 & 0 & & 1 \\ \vdots & & & \ddots & \\ 0 & 0 & 1 & & 0 \end{bmatrix} \tag{3.2}$$

Definition 8 *Set Equivalence*

The training set and the operation set are equivalent if $\mathbf{T} + \mathbf{O} = \mathbf{T}$*, where* $+$ *is the logical OR operator for matrices;* $a_{i,j} + b_{i,j} = c_{i,j}$ *for all* $i = \{0, \ldots, N-1\}$ *and* $j = \{0, \ldots, N-1\}$*. In words, every basis task transition in the operation set has been learned in the training set. A training and operation set that have set equivalence is shown schematically, Fig. 3.3, where every task transition that occurs in the operation*

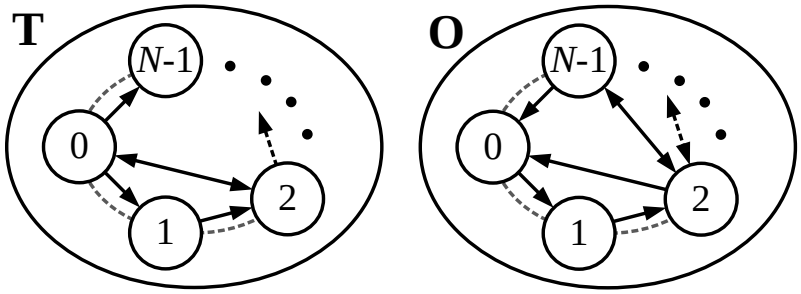


Figure 3.2: Directed graph of transitions between basis tasks. Each path has a probability of 1 of occurring since each transition is specified by the manufacturing instructions. This training and operation set graphs can be written in matrix form, **T** and **O** respectively, given in Eq. (3.2)

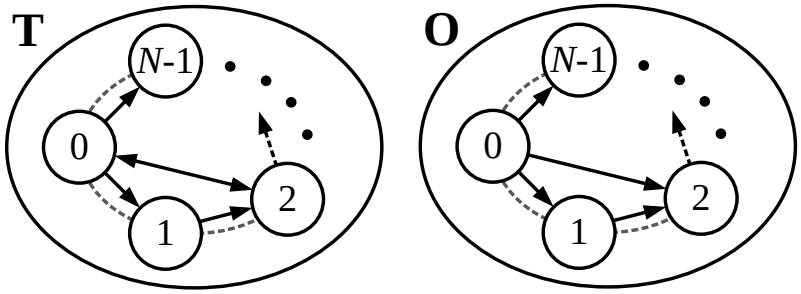


Figure 3.3: Example of a *training* and *operation set* that satisfy *set equivalence*.

set was previously identified in the training set. In contrast, the training and operation set in Fig. 3.2 would not satisfy set equivalence.

3.2.2 Assumptions

Assumption 1: *Training Set and Operation Set in Operation and Dynamic Spaces* The reference signals for the *training set* and *operation set*, $r^T(k)$ and $r^O(k)$, are comprised of *basis tasks*:

$$\begin{aligned} &tbasis\left(r^{T_0}(k), r^{T_1}(k), \ldots, r^{T_{P-1}}(k)\right) = \{r_0(k), r_1(k), \ldots, r_{N-1}(k)\} \\ &tbasis\left(r^{O}(k)\right) = \{r_0(k), r_1(k), \ldots, r_{N-1}(k)\} \end{aligned} \tag{3.3}$$

and the set of *configurations* utilized in the *training set* and *operation set* are in the

dynamic space.

Assumption 2: *Training Set Appropriate for ILC*

Systems appropriate for traditional applications of ILC must track a repeated trajectory, be time-invariant, have discontinuous operation, and have identical initial conditions, Chapter 2. Here, these constraints are weakened and only apply to the *training set*. The *operation set* trajectories can be arbitrarily chosen provided that *Assumption 1* is satisfied.

3.3 Design Application

This section provides the BTILC procedure for general systems. Subsections 3.3.1 and 3.3.2 correspond to the *training set* and subsections 3.3.3 – 3.3.3 correspond to the *operation set*. A specific application is given in Chapter 5. A schematic of BTILC provides a visualization of the manufacturing operations performed in the *Training* and *Operation Sets*, Fig. 3.4.

3.3.1 Training Routine

A set of training reference trajectories, $\left\{r^{T_0}(k), r^{T_1}(k), \ldots, r^{T_{P-1}}(k)\right\}$, is selected such that the set encompasses all the *basis tasks* in the *operation space*,

$$tbasis\left(r^{T_0}(k), r^{T_1}(k), \ldots, r^{T_{P-1}}(k)\right) = \{r_0(k), r_1(k), \ldots, r_{N-1}(k)\} \tag{3.4}$$

where each $r^{T_p}(k)$ will have the form:

$$r^{T_p}(k) = \sum_{i=a_p}^{b_p} \sum_{m_i=0}^{M_i-1} r_i(k - \Delta_{m_i}) \left(\mathfrak{s}(k - \Delta_{m_i}) - \mathfrak{s}(k - \Delta_{m_i} - K_i)\right). \tag{3.5}$$

Here, the set of *basis tasks* $[a_p, b_p]$ is a subset of the *basis tasks* in the *operation space*, $[0, N-1]$, and each *basis task* may have M_i multiplicities in the *training set*. The

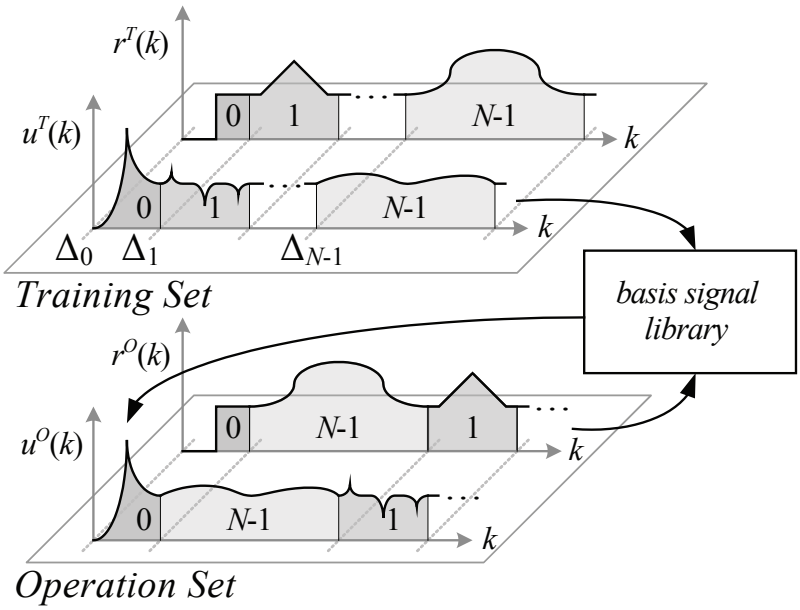


Figure 3.4: Visual depiction of BTILC. *Basis tasks* are learned in a training routine and the corresponding *basis signal* information is stored in the *basis signal library*. In the *operation set*, *basis signals* are applied as specified by logic applied to a schedule of *basis tasks*.

constraint:

$$\Delta_{i+1} \geq \Delta_i + K_i \tag{3.6}$$

is applied so that the *basis tasks* do not overlap in time. ILC is applied with a given $r^{T_p}(k)$ to the *configuration*, H_d. A typical ILC algorithm is shown in (3.7) [10]:

$$u^{T_p(j+1)}(k) = Q(q)\left(u^{T_p(j)}(k) + L(q)e^{T_p(j)}(k+1)\right). \tag{3.7}$$

Details on ILC can be found in Chapter 2. Assuming an $L(q)$ and $Q(q)$ such that the combination of (2.1) and (3.7) is stable, (2.8), and sufficient performance is achieved, the resultant $u_d^{T_p}(k)$ is considered to be the best input signal for reference tracking. This procedure is applied to all D *configurations* and P training reference trajectories, yielding $D \cdot P$ unique $u_d^{T_p}(k)$ signals.

3.3.2 Signal Segmentation

Given (3.5) and (3.6) each *basis task* is separated in time. Therefore, the input signal identified in the ILC training sequence can be decomposed into individual *basis signals*:

$$u^{T_p}(k) = \sum_{i=a_p}^{b_p} \sum_{m_i=0}^{M_i-1} u_i(k-\Delta_i)\left(\mathfrak{s}(k-\Delta_i) - \mathfrak{s}(k-\Delta_i-K_i)\right). \tag{3.8}$$

These individual *basis signals* are segmented at the transition times, Δ_i, and are stored in memory to be accessed during machine operation. A general data structure for *basis signal* storage is given in (3.9), however the data structure will be application dependent and may take on different forms. The authors term the collection of *basis signal* information the *basis signal library* where there will be D libraries for D *configurations*.

$$\text{library}_d = \left\{ \begin{array}{cccc} lead_0 & lead_1 & \cdots & lead_{N-1} \\ K_0 & K_1 & & K_{N-1} \\ u_{0,d}(0) & u_{1,d}(0) & & u_{N-1,d}(0) \\ \vdots & & \ddots & \\ u_{0,d}(K_0-1) & u_{1,d}(K_1-1) & & u_{N-1,d}(K_{N-1}-1) \end{array} \right\} \tag{3.9}$$

In general, ILC is noncausal, with the start of a *basis signal* preceding the occurrence of a *basis task* prompt in the machine commands. The first row of (3.9) stores the lead distance, $lead_n$, between the beginning of a *basis signal* and the intended *basis task* occurrence. $lead_n$ can be specified in either time or another system state variable. K_n is the signal length, given in *Definition 2*, and the $u_{n,d}(k)$ terms are the discrete-time *basis signals* indexed by *basis signal*, n, and discrete task time, k.

3.3.3 Basis Signal Extraction

Basis signals are extracted from the *basis signal library* based on logic applied to the set of instructions dictating manufacturing operation. *Basis signal* extraction is shown schematically in Fig. 3.4 where the *basis signals* are concatenated to construct the entire input signal. Given an instructed *basis task*, the corresponding *basis signal* is applied to the plant with the appropriate time shift.

$$u^O(k) = \sum_{i=0} \mathbf{u_i}(k - \mathbf{\Delta_i}) \left(\mathfrak{s}(k - \mathbf{\Delta_i}) - \mathfrak{s}(k - \mathbf{\Delta_i} - \mathbf{K_i})\right) \tag{3.10}$$

where $\mathbf{u} = \{u_0(k), u_1(k), \ldots\}$, $\mathbf{\Delta} = \{\Delta_0, \Delta_1, \ldots\}$, and $\mathbf{K} = \{K_0, K_1, \ldots\}$ are appropriately designed ordered vectors of *basis signals*, time shifts, and *basis signal* lengths, respectively.

The following subsections outline a general logic structure that extracts the correct *basis signal* as interpreted from the sequence of machine commands. The authors believe the following logic to be relevant to most applications.

Current and Future Tasks

The *basis signals* are generally non-causal, preceding the prompt for a *basis tasks*. Therefore the logic algorithm must access the command list in advance to determine and process future *basis task* information. Accordingly, the *basis signal* for the next *basis task* will begin as the current *basis task* is being completed.

Hierarchy of Tasks

For a given application, some *basis tasks* will be prioritized and therefore there is a task hierarchy to consider in *basis signal* extraction. An example of a hierarchical decision would be a pick-and-place robot in which the 'pick' task would take highest priority to prevent the robot from missing items on an assembly line. Fig. 3.5 provides

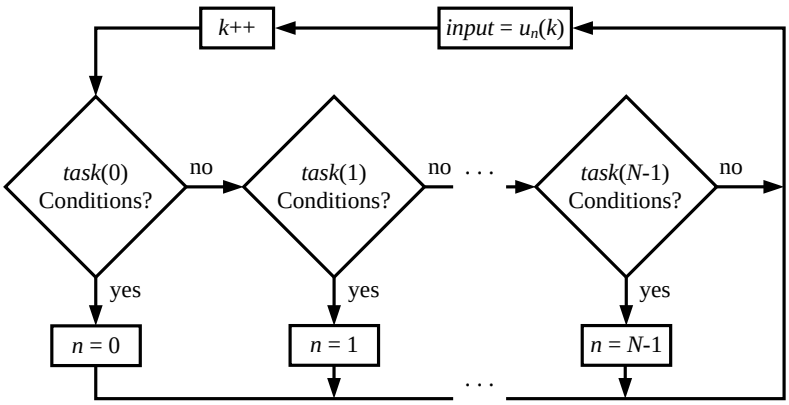


Figure 3.5: Generic logic structure to enforce a *basis task* hierarchy during *basis signal* extraction.

a generic logic structure to enforce a *basis task* hierarchy.

Allowable Task Sequence

It is advantageous to ensure that there is *set equivalence*, Definition 8. Therefore, a set of allowable *basis task* sequences may be written into the decision logic to allow only transitions that are *set equivalent*. A general automaton of allowable task sequences is displayed in Fig. 3.6. The allowable task sequence enforces task sequencing rules on two levels: at the intra-*configuration* and inter-*configuration* level.

3.4 Performance Considerations

3.4.1 Stability

Given that every *configuration*, H_d, in the *dynamic space* is stable or stabilizable and that the *basis signals* are bounded, by (2.8), the *operation set* output $y^O(k)$ is bounded.

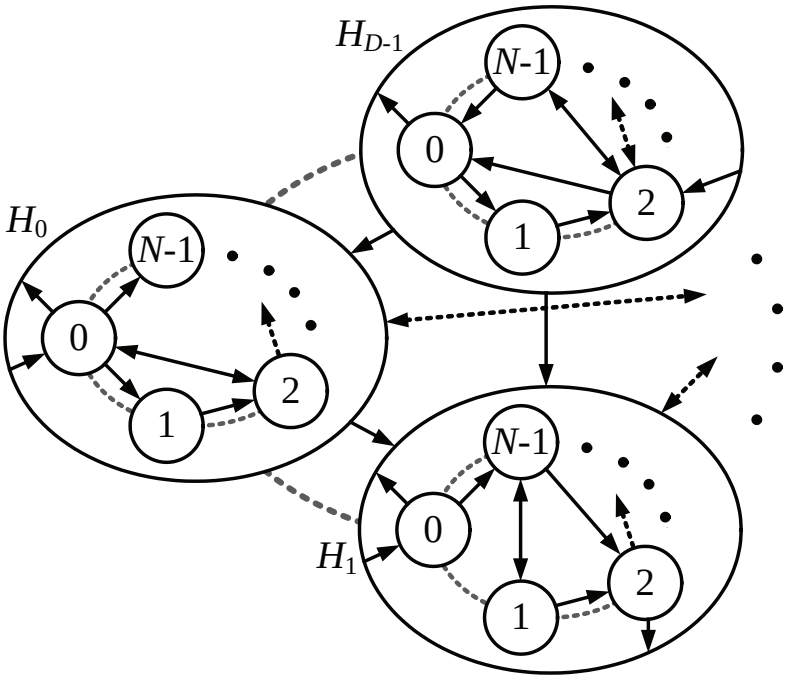


Figure 3.6: General automaton enforcing allowable sequences of *basis tasks*. *Basis tasks* within a *configuration* are represented by circular nodes where each *configuration* has N *basis tasks*. *Configurations* are represented by ellipse nodes where there are D *configurations*.

3.4.2 Performance

Consider the *training set* and operator H_d in (2.1). The training reference, $r^T(k)$, is comprised of temporally sequenced *basis tasks*, (3.5), where a *basis task* $r_n(k)$ is active over an interval starting at a task-transition time index, Δ_i. ILC is applied with this training reference, converging to a *basis signal*, $u^T(k)$, that gives the performance:

$$\begin{aligned} y^T(k) &= H_d\left(u^T(k)\right) \\ e^T(k) &= r^T(k) - y^T(k) \end{aligned} \tag{3.11}$$

where $u^T(k)$, $y^T(k)$, and $e^T(k)$ can be decomposed temporally into their *basis task* specific signals. At the task-transition indices, $\left\{\Delta_{a_p}, \ldots, \Delta_{b_p}\right\}$, operator H_d in (2.1) has the following states, $\left\{x_{a_p}, \ldots, x_{b_p}\right\}$, that effectively give the initial conditions for each task, Fig. 3.7

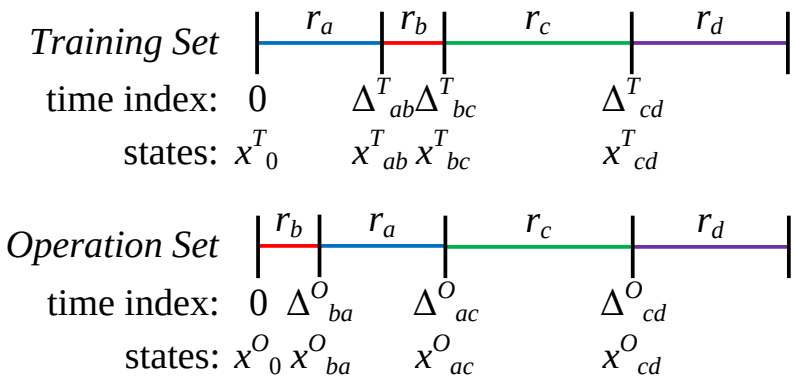


Figure 3.7: Conceptual schematic of the states at the transition indices between adjacent *basis tasks* in the *training* and *operation sets*. A different sequence of *basis tasks* will yield different states because of either historical differences or because *set equivalence* is not satisfied.

In the *operation set*, however, the sequence of tasks given by $r^O(k)$ is different than the sequence in $r^T(k)$. The states at the transition between *basis tasks*, or initial conditions, will be different because of historical differences or because *set equivalence*, Definition 8, was not enforced:

$$x_n^O = x_n^T + \delta x_n \tag{3.12}$$

where δx_n is the difference in states between the *training* and *operation sets* at each task-transition. Therefore, there will be a performance decrease because the *basis signals*, $u_n^T(k)$, are identified for the *training set* transition states, x_n^T, not the *operation set* transition states, x_n^O. The resultant error signal will be:

$$e_n^O(k) = e_n^T(k) - \mathbf{C}_d \mathbf{A}_d^k \delta x_n, \tag{3.13}$$

which can be shown by properties of linear systems:

$$
\begin{aligned}
e_n^O(k) &= r_n(k) - \mathbf{C}_d\mathbf{A}_d^k x_n^O - \mathbf{C}_d \sum_{i=o}^{k-1} \mathbf{A}_d^i \mathbf{B}_d u_n(i) \\
e_n^T(k) &= r_n(k) - \mathbf{C}_d\mathbf{A}_d^k x_n^T - \mathbf{C}_d \sum_{i=o}^{k-1} \mathbf{A}_d^i \mathbf{B}_d u_n(i) \\
e_n^O(k) &= e_n^T(k) - \mathbf{C}_d\mathbf{A}_d^k \left(x_n^O - x_n^T\right)
\end{aligned}
\tag{3.14}
$$

The additional error introduced by BTILC, $e_n^O(k) - e_n^T(k)$, is equivalent to the free-response of the system to non-zero initial conditions, where the difference in states between the *training* and *operation sets*, δx_n, is in place of the initial conditions.

Remark 1:

The 2-norm of the additional error introduced by BTILC at each *basis task* is bounded by a function of δx_n and the observability Grammian, $\mathbf{W}_d$,

$$
\left\| e_n^O(k) - e_n^T(k) \right\|_2 < \sqrt{\delta x_n^T \mathbf{W}_d \delta x_n} \tag{3.15}
$$

where $\mathbf{W}_d$ is the solution of the Lyapunov equation:

$$
\mathbf{W}_d - \mathbf{A}_d^T \mathbf{W}_d \mathbf{A}_d = \mathbf{C}_d^T \mathbf{C}_d. \tag{3.16}
$$

Proof:

$$
\begin{aligned}
e_n^O(k) - e_n^T(k) &= -\mathbf{C}_d\mathbf{A}_d^k \delta x_n \\
\left\| e_n^O(k) - e_n^T(k) \right\|_2 &= \sqrt{\delta x_n^T \left(\sum_{k=0}^{K_n-1} (\mathbf{A}_d^T)^k \mathbf{C}_d^T \mathbf{C}_d \mathbf{A}_d^k \right) \delta x_n}
\end{aligned}
\tag{3.17}
$$

where:

$$
\sum_{k=0}^{K_n-1} (\mathbf{A}_d^T)^k \mathbf{C}_d^T \mathbf{C}_d \mathbf{A}_d^k < \sum_{k=0}^{\infty} (\mathbf{A}_d^T)^k \mathbf{C}_d^T \mathbf{C}_d \mathbf{A}_d^k = \mathbf{W}_d \tag{3.18}
$$

for a finite dimensional *basis task*. $\mathbf{W}_d$ is the solution of the Lyapunov equation in (3.16) for an $\mathbf{A}_d$ with eigenvalues less than 1 in magnitude. Therefore $\left\|e_n^O(k) - e_n^T(k)\right\|_2 < \sqrt{\delta x_n^T \mathbf{W}_d \delta x_n}$.

□

Remark 2:

δx_n can be thought of as a metric of similarity between the *training* and *operation sets*. In general, the system states will not be known and therefore δx_n will not be calculable. However, this analysis does provide insight and recommendations for *basis task* selection and *basis signal* application. The main insight is that the error from the dissimilarity between the *training* and *operation sets* will build and decay at a rate given by the eigenvalues and order of matrix $\mathbf{A}_d$. The primary design recommendation is that efforts should be made to ensure that δx_n is as small as possible. In order of importance, this includes:

1. not allowing *basis tasks* to overlap in time in the *operation set*; enforcing a similar constraint in the *operation set* as in (3.6).

2. ensuring that there is *set equivalence*, Definition 8, between *training* and *operation sets*. This consideration is discussed in Section 3.3.3

3. choosing *basis tasks* to be long enough in duration such that transitions occur at regions of relatively little error fluctuation; that is, the first derivative of the error should be small. This consideration introduces a trade-off between *basis task* brevity, and hence flexibility, and accuracy.

Chapter 4

Micro-Robotic Deposition

One system that can be setup to satisfy the assumptions for BTILC is micro-Robotic Deposition (μRD). μRD, also termed robocasting and direct ink writing, was invented at Sandia National Laboratory by Cesarano and co-workers [35]. μRD is a nozzle-based AM process in which a colloidal build material is extruded through a nozzle in a predefined trajectory to build three-dimensional structures with micro-scale features. Structures are built in a layer-by-layer fashion, similar to the more established technology FDM [6] except using a ceramic or polymeric material at room temperature instead of a polymer melt. The colloidal material has carefully tailored viscoelastic properties to facilitate material flow through a nozzle while maintaining a stiffness appropriate for spanning structural gaps up to 2 mm [49]. These properties allow the fabrication of porous structures without the use of lost molds, making μRD well suited for applications such as artificial bone scaffolds [13,31,50], piezoelectric actuators and sensors [51, 52], microfluidic networks [53], and photonic bandgap structures [54].

4.1 System Components

This section provides μRD system components. The proposed BTILC will be applied to the material system in Chapter 5 and to the positioning system in Chapter 7.

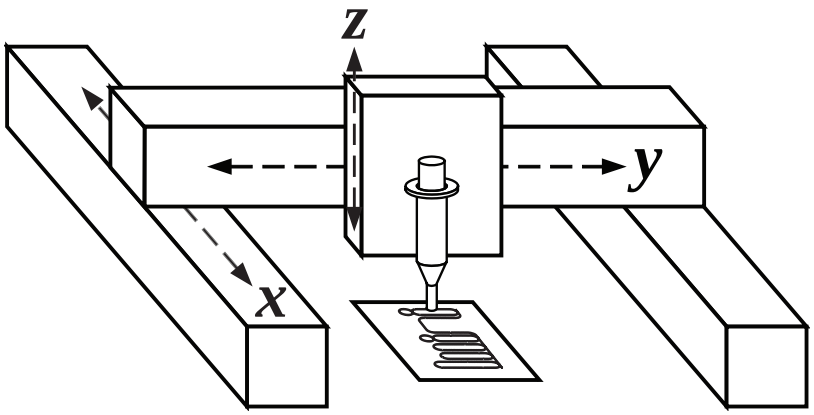


Figure 4.1: Diagram of a serial positioning system tracking an arbitrary trajectory.

4.1.1 XYZ Positioning System

The position of the deposition head is controlled by an XYZ serial positioning system with position feedback as described in [55]. In this specific design, the axes are stacked serially where the x-axis carries the y and z-axes and the y-axis carries the z-axis, shown schematically in Fig. 4.1. The positioning system has an accuracy and resolution that is significantly smaller than the length scales pertinent to structures typically fabricated by μRD. For all work in this manuscript pertaining to the manufacture of structures, Chapters 5 and 6, the positioning error is assumed to be approximately zero.

Other axes orientations are possible. A popular option utilizes stacked x and y axes that are mounted below an independent z axis which incrementally raises the deposition head height [56]. The deposition system is stationary during the deposition of each layer and the substrate attached to the x and y-axes moves.

4.1.2 Multi-Material Deposition Head

The deposition system mounted to the XYZ positioning system is a prototype multi-material deposition head. The deposition head contains four individual extrusion systems oriented in a circular array, see Fig. 4.2. However, the design here is extend-

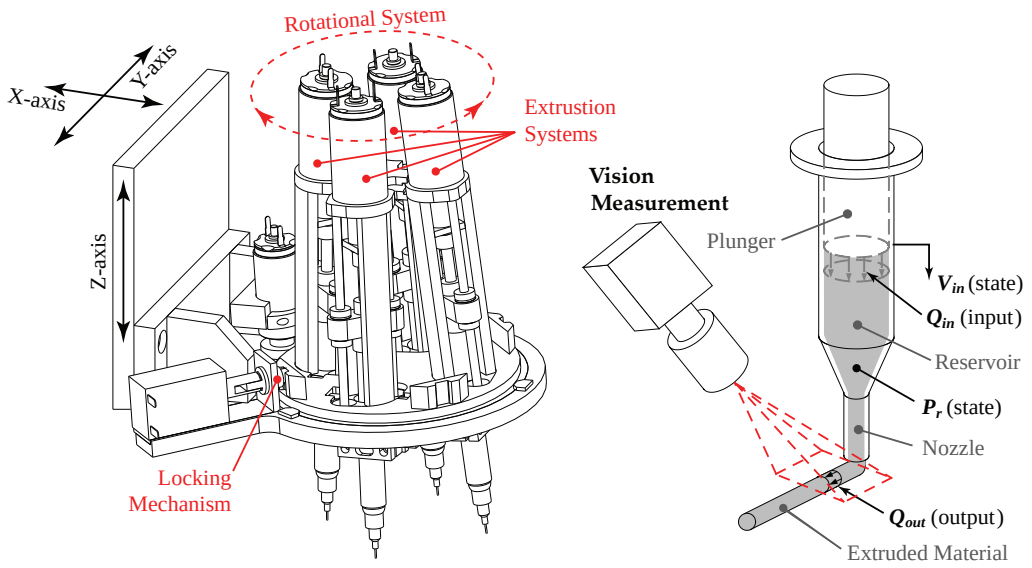


Figure 4.2: Deposition system displaying the array of four extrusion systems. Rotational system indexes between extrusion systems and therefore materials. Build material is extruded, Q_{out}, by the displacement of a plunger, Q_{in}. The two state variables are the displaced volume, V_{in}, and the reservoir pressure, P_r. A machine vision system measures extruded material volumetric flowrate.

able to more than four extrusion systems. Each individual extrusion system consists of a motor and lead screw assembly, which displaces a plunger and in turn applies pressure to the build material reservoir to extrude the material. An individual extrusion system is selected by rotating that system into the 'active' position. The actual hardware used is shown in Fig. 4.3. Extrusion and rotational system positioning are controlled by feedback control. Complete engineering drawings of the multi-material deposition head and a bill of materials are provided in Appendix E.

4.1.3 Machine Vision System

Material flowrate is measured by a machine vision system. A CCD video camera is mounted to the XYZ positioning system and focused at the exit of the syringe nozzle. Video of each extrusion iteration is recorded, saved, and input to a machine vision processing software package. A general overview of the processing algorithm is given

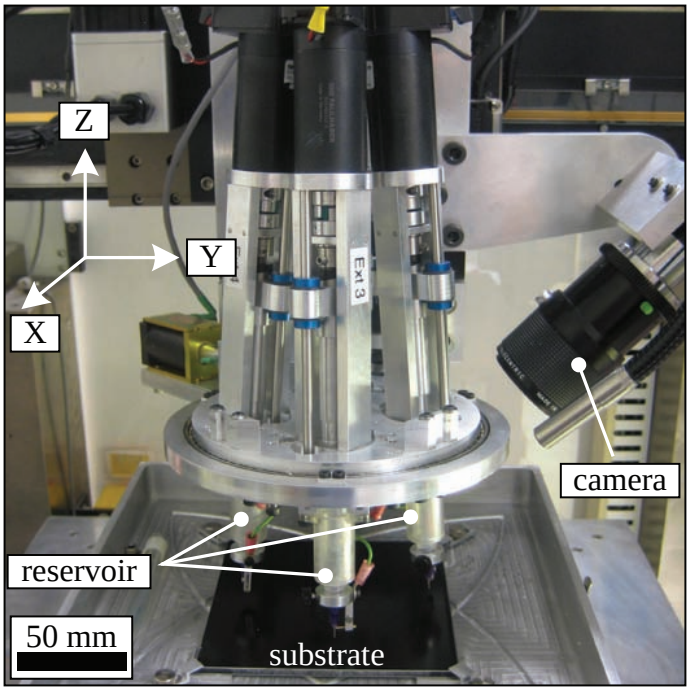


Figure 4.3: Image of the μRD hardware displaying the positioning system, extrusion system, and material flowrate measurement system which uses machine vision.

in Fig. 4.4 and the exact software is given in Appendix F. Briefly, the first relevant video frame is identified by an embedded motion sensing script, `Movement.m`. Image data is then loaded into memory and then a region of interest near the nozzle exit is specified and individual frames are spliced together based off the region of interest and calibrations. Spliced together frames are converted from an RGB image to a binary (black and white) image with a simple threshold operation. The width of the extruded filament of material at each point along the trajectory is measured and then correlated to a volumetric flowrate with the transformation equation:

$$Q_{out} = \begin{cases} \frac{\pi}{4} RW^2 & \text{for } 0 \leq RW \leq h \\ \frac{1}{2}\theta RW^2 + \frac{1}{2}h^2 \frac{1}{\tan\theta} & \text{for } RW > h \end{cases} \tag{4.1}$$

where RW is the measured rod width, h is the standoff height between the nozzle

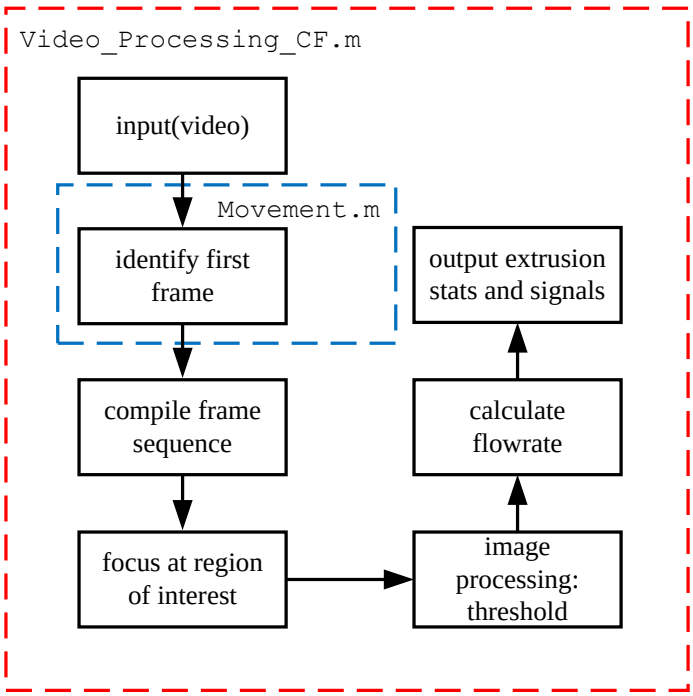


Figure 4.4: Outline of the machine vision processing algorithm; code is given in Appendix F.

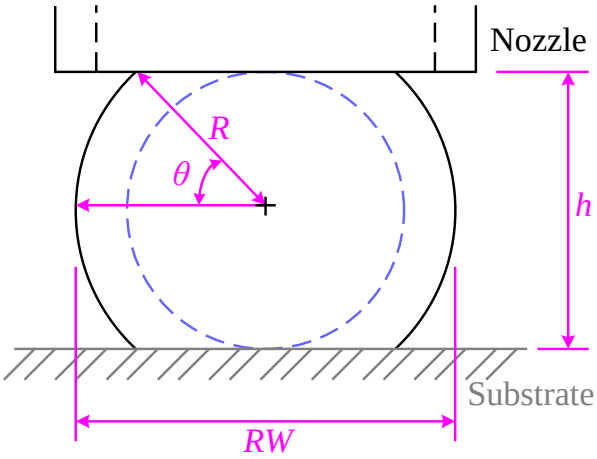


Figure 4.5: Assumed cross-section of extruded material. Geometry is used in (4.1) to correlate a measured rod width, RW, to a volumetric flowrate, Q_{out}.

and the substrate, and $\theta = \sin^{-1}\left(\frac{h}{RW}\right)$. (4.1) assumes the extruded cross-section given in Fig. 4.5. Cross-section images, not shown [57], and the μRD literature [35] support this assumption. After volumetric flowrate correlation, the data is converted to a time-domain signal using calibration information.

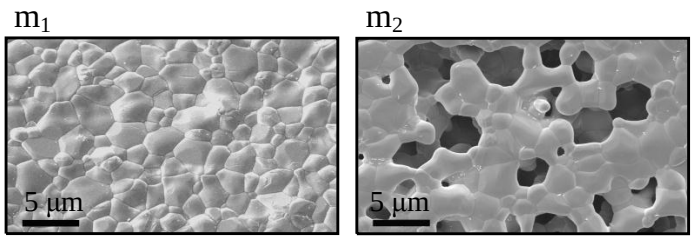


Figure 4.6: Microstructures of the two different materials that constitute the material system. m_1 has a dense microstructure where the individual hydroxyapatite grains are almost completely fused together. m_2 has a porous microstructure where there is interconnected void space permeating the scaffold. Images have been modified from [13]; used with permission.

4.1.4 Material System

This manuscript focuses on a material system consisting of two variations of the hydroxyapatite (HA) colloidal material developed in [31]. Appendix A provides explicit instructions for material synthesis. Many more material systems are conceivable, but to focus on manufacturing problems we limit the set of materials considered to these two; one is a dense microstructure material (0 $vol\%$ nominal void fraction), m_1, and the other is a porous microstructure material (50 $vol\%$), m_2. Material microporosity is tailored during materials synthesis by the inclusion of polymer microspheres in the appropriate volume ratio in the colloidal material. Microspheres volatilize during the post-process sintering stage, Section 4.3.2, leaving void space. Fig. 4.6 demonstrates the microstructure of each material. m_1 does not have induced porosity, however it has trace amounts of nano-scale porosity from incomplete sintering, $< 1\ \mu m$ diameter pores comprising $< 2\%$ of the volume [31]. m_2 has a mean pore size of 4.86 μm with porosity comprising 46% of the material volume [58], by the formulation given in [13]. Material characteristics are given in [58].

4.2 μRD System Model Development

4.2.1 Positioning System Model

Dynamic models of the x and y-axes were identified in [55]. Parameter values for the transfer functions of the axes plant models, $H_d(z)$, and stabilizing feedback controllers, k_d, with the forms given in (4.2) and (4.3) are given in Tb. 4.1. Frequency response plots for the open-loop transfer function, $H_d(z)$, and the complementary sensitivity function, $T_d(z) = \frac{k_d(z)H_d(z)}{1+k_d(z)H_d(z)}$, are given in Figs. 4.7 and 4.8, respectively. All results were acquired with a sampling rate, $1/t_s$, of 1 kHz.

$$H_d(z) = \frac{K(z+\alpha_1)(z^2+\alpha_2 z+\alpha_3)(z^2+\alpha_4 z+\alpha_5)}{(z+\beta_1)(z-1)(z^2+\beta_2 z+\beta_3)(z^2+\beta_4 z+\beta_5)} \tag{4.2}$$

$$k_d(z) = \frac{K(z^2+\alpha_1 z+\alpha_2)}{(z-1)(z+\beta_1)} \tag{4.3}$$

Table 4.1: System Parameters and Controller Design

Plant	Parameter				
Den	β_1	β_2	β_3	β_4	β_5
H_x	-0.9994	-1.978	0.9894	-1.738	0.8672
H_y	-0.9994	-1.983	0.9911	-1.87	0.9539
k_x	-0.7408				
k_y	-0.7408				
Num	α_1	α_2	α_3	α_4	α_5
H_x	0.9604	-1.981	0.9918	-1.874	0.9747
H_y	1	-1.983	0.9912	-1.873	0.9547
k_x	-1.941	0.9423			
k_y	-1.949	0.9506			
Gain	K				
H_x	8.3315×10^{-4}				
H_y	1.8506×10^{-3}				
k_x	38				
k_y	27.375				

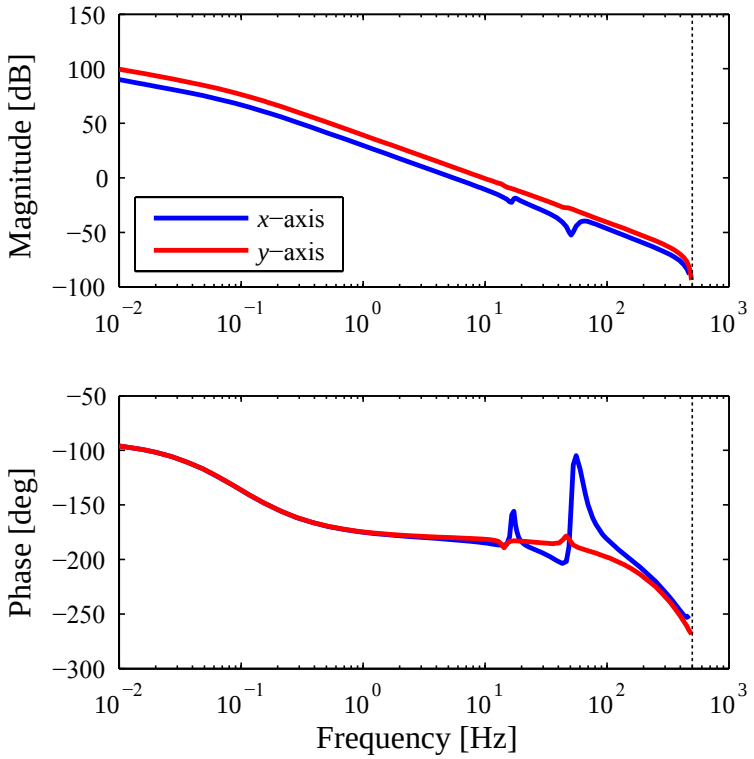


Figure 4.7: Frequency responses of the open-loop responses, $H_d(z)$, for the x and y-axes, Eq. (4.2).

4.2.2 Material Extrusion Model Development

In all work except Chapter 7, the material system dynamics are assumed to be the dominant time scale, as compared to positioning and extrusion system dynamics; 7.9x10^{-2} Hz bandwidth for the material system as compared to 35 Hz for the positioning system [55] and 1.9 Hz for the extrusion system. The material system dynamics are derived from non-Newtonian fluid dynamics theory. The colloidal material of interest here is characteristic of a yield-pseudoplastic fluid [14]. Yield-pseudoplastic fluids exhibit a nonlinear response in which the fluid does not deform unless a yield-stress is exceeded. Above the yield stress, the flow is shear-thinning, meaning that the fluid viscosity decreases with increasing shear rate [59]. This behavior is captured by the Herschel-Bulkley model:

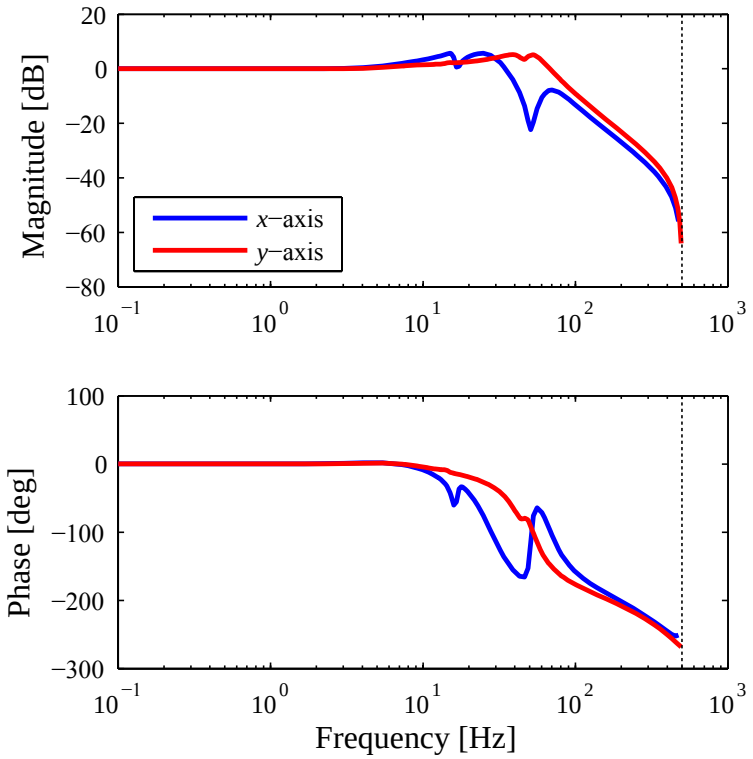


Figure 4.8: Frequency responses of the complementary sensitivity function, $T_d(z) = \frac{k_d(z)H_d(z)}{1+k_d(z)H_d(z)}$, for the x and y-axes.

$$\tau = \tau_0 + m\dot{\gamma}^n \tag{4.4}$$

where the shear stress, τ, is a function of the material yield stress, τ_0, shear rate, $\dot{\gamma}$, fluid consistency index, m, and the flow behavior index, n [14].

A model for the relationship between Q_{in} and Q_{out} is developed in two parts. First considering a control volume of fluid in the reservoir, the pressure change in the reservoir, $\dot{P}_r$, can be calculated by a rate balance:

$$\dot{P}_r = \frac{\beta}{V_0 - V_{in}} (Q_{in} - Q_{out}) \tag{4.5}$$

where β is the fluid compliance, V_0 is the original syringe volume, and V_{in} is the displaced fluid volume. V_{in} is contained in the set $V_{in} \in [0, V_0]$. That is, the total

displaced volume is bounded by the total reservoir volume. The second part of the model considers a yield-pseudoplastic material flowing through a nozzle in a laminar flow regime. Governed by the Herschel-Bulkley model, yield-pseudoplastic fluids have the flowrate profile given in Fig. 4.9, where there is an unyielding fluid core in the nozzle center. At the radius, r, at which the shear-stress exceeds the yield-stress, the material is shear-thinning, and the fluid velocity profile as a function of radius, $V(r)$, is super-parabolic until it reaches zero at the nozzle wall, assuming zero wall slip. The equation governing this response is complex and not well-suited for integration with controls tools:

$$Q_{out} = \begin{cases} \pi R^3 n \left(\frac{\tau_w}{m}\right)^{1/n} (1-\phi)^{(n+1)/n} \left\{ \frac{(1-\phi)^2}{3n+1} + \frac{2\phi(1-\phi)}{2n+1} + \frac{\phi^2}{n+1} \right\} & \text{for } \phi \leq 1 \\ 0 & \text{for } \phi > 1 \end{cases} \quad (4.6)$$

where $\phi = \frac{\tau_0}{\tau_w}$ and $\tau_w = \left(-\frac{P_r}{L}\right)\frac{R}{2}$. τ_w is the shear-stress at the nozzle wall and L is the nozzle length. Assuming that material is pseudoplastic instead of yield-pseudoplastic, the equation simplifies from the piece-wise continuous nonlinear function in Eq. (4.6) to the nonlinear function in Eq. (4.7).

$$Q_{out} = \pi \frac{n}{3n+1} \left(\frac{P_r}{2mL}\right)^{1/n} R^{(3n+1)/n} \quad (4.7)$$

The two-part model is combined to give a two-state, V_{in} and P_r, nonlinear model:

$$\begin{aligned} \dot{V}_{in} &= Q_{in} \\ \dot{P}_r &= \frac{\beta}{V_0 - V_{in}} Q_{in} - \frac{\beta}{V_0 - V_{in}} \pi \frac{n}{3n+1} \left(\frac{P_r}{2mL}\right)^{1/n} R^{(3n+1)/n} \end{aligned} \quad (4.8)$$

with an output that is a nonlinear function of the the state P_r, Eq. (4.7). Linearizing

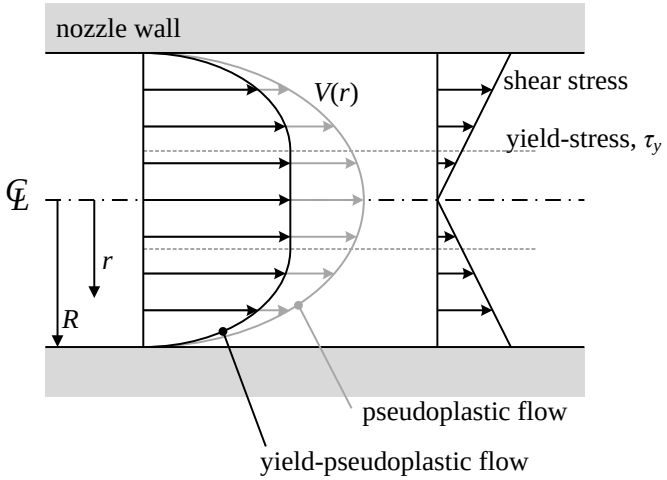


Figure 4.9: Velocity profiles for a yield-pseudoplastic and a pseudoplastic fluid flowing through a nozzle. Yield-pseudoplastic fluids have an unyielding fluid core at stresses below the yield-stress. Adapted from [14].

about steady operating points, V_{in}^* and P_r^*, gives the two-state state-space Wiener model [60] in which there are linear state relations and a nonlinear dependence of the output on the states:

$$\begin{bmatrix} \frac{d\delta V_{in}}{dt} \\ \frac{d\delta P_r}{dt} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 \\ A_{2,1} & A_{2,2} \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} \delta V_{in} \\ \delta P_r \end{bmatrix} + \underbrace{\begin{bmatrix} 1 \\ \frac{\beta}{V_0 - V_{in}^*} \end{bmatrix}}_{\mathbf{B}} \delta Q_{in}$$

$$A_{2,1} = \frac{\beta}{(V_0 - V_{in}^*)^2} Q_{in}^* - \frac{\beta}{(V_0 - V_{in}^*)^2} \pi \frac{n}{3n+1} \left(\frac{P_r^*}{2mL} \right)^{1/n} R^{(3n+1)/n}. \tag{4.9}$$

$$A_{2,2} = -\frac{\beta}{V_0 - V_{in}^*} \frac{\pi}{2mL} \frac{R^{(3n+1)/n}}{3n+1} \left(\frac{P_r^*}{2mL} \right)^{(1-n)/n}$$

$$Q_{out} = \pi \frac{n}{3n+1} \left(\frac{P_r}{2mL} \right)^{1/n} R^{(3n+1)/n}$$

It is obvious that (4.9) is not asymptotically stable since the **A**-matrix has a zero eigenvalue. However, we are restricting the input, Q_{in} to be bounded with bounded derivatives and limiting the state $V_{in} \in [0, V_0]$; therefore the boundedness of V_{in} is not a concern. In μRD, flowrates are typically very small, $\ll 1\ mm^3/mm$, and we

can assume that V_{in} does not change appreciably over short time periods. With this realization, (4.9) can be further simplified by choosing the state V_{in} to be constant and linearizing the nonlinear function about a steady output. The resulting discrete-time transfer function has the form in (4.10), relating the plunger displacement rate input (Q_{in}) to the volumetric flowrate through the nozzle (Q_{out}).

$$H_d(z) = \frac{Q_{out}(z)}{Q_{in}(z)} = \frac{b_1}{z + a_1}. \tag{4.10}$$

Although the simple model in (4.10) is a stark deviation from the original model developed in (4.5) and (4.6), (4.10) has been validated many times with over a dozen different batches of build material. Section 4.2.3 will demonstrate that (4.10) captures the dominant dynamics of material extrusion in μRD.

4.2.3 Material Extrusion Model Validation

Fig. 4.10 is a representative validation of the model in (4.10). To elucidate the testing procedure, at a steady positioning system velocity of 5 mm/sec material is extruded in a line on a substrate for a sufficient period of time to allow the flowrate to reach steady-state. Flowrate is then turned off, denoted by the step drop in reference flowrate trajectory from the nominal 0.132 mm^3/mm to 0 mm^3/mm. This test is repeated ten times to give an average response. All flowrate units are expressed in mm^3/mm because spatial information is more important to building an accurate structure than temporal information, e.g. mm^3/sec. All data is sampled at 1 kHz and then correlated to the spatial dimension with a known positioning system velocity map. The responses of *configuration* H_1 and H_2 are first-order linear systems as predicted by (4.10). Model (4.10) is fit to each data set, yielding the system parameters given in Tb. 4.2. Material m_1 is more viscous than Material m_2, a property that is realized in the parameter a_1 of the discrete-time response of H_1 and H_2.

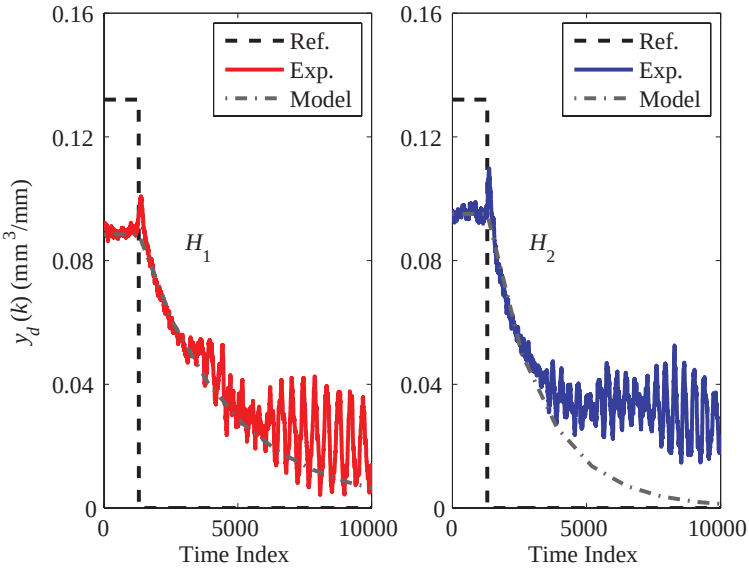


Figure 4.10: Validation of model (4.10). Each data set is the mean of 10 trials.

The responses deviate from model (4.10) at low flowrates because of model linearization and because the assumption that the yield-stress can be omitted is less valid at low flowrates. A consequence of the yield-stress omission is seen in the oscillatory flowrate measurement at low-flowrates that is from an intermittent, flow / no-flow, nonlinearity at stresses near the yield stress. The spike in flowrate measurement at the downward step in flowrate, $k = 1300$ in Fig. 4.10, is a consequence of the machine language interpolator software used. All flowrate commands are specified at a certain position and the interpolator software decelerates the positioning system into that position and accelerates it out of that position. At this junction, the material flowrate is essentially unchanged and the positioning system speed is decreased, leading to a swelling of material and therefore a spike in measured flowrate. In spite of the slight time-domain mismatch, Fig. 4.10 clearly demonstrates that (4.10) captures the major dynamics of the material system.

Table 4.2: First-Order System Parameters for model (4.10)

Parameter	H_1	H_2
b_1	2.030x10^{-4}	3.599x10^{-4}
a_1	-0.9997	-0.9995

4.3 Scaffold Fabrication

4.3.1 General Fabrication

A general scaffold manufacturing protocol is given in detail in Appendix B. In brief, materials are loaded into 5 mL syringes (EFD 5110LL-B), which are centrifuged (Eppendorf 5702) at 3000 rpm for 3 min to remove suspended air bubbles. Reconstructed Micro-Computed Tomography (μCT, SkyScan 1172) images and Scanning Electron Microscope (SEM, Philips XL30 ESEM-FEG) images of centrifuged material displayed no evidence of a density gradient formed by the centrifugal bubble removal process [57]. μCT is an imaging modality that measures x-ray attenuation to build three-dimensional density maps with micro-meter resolutions [66]. The syringes are then fitted with the desired diameter nozzle (EFD, inc.) to direct material flow and a piston (EFD 5110PDP-B) to apply the deposition pressure. Syringes are loaded into the μRD system detailed in Section 4.1. In general, the first layer has a 'stand-off' height above the substrate of $0.80d$ and each subsequent layer is translated upwards by 0.77Ø, where Ø is the nozzle diameter. A general rule for continuous material deposition is that the plunger volumetric flowrate should match the desired extruded volumetric flowrate, $Q_{in} = Q_{out} = \frac{\pi}{4}\text{Ø}^2 v$, where v is the positioning system velocity. Structures are deposited while submerged in a non-wetting oil in order to prevent non-uniform drying. Scaffolds are built layer-by-layer with each layer having one *Start* and one *Stop basis task* for each material, m_1 and m_2. At each material switch, the nozzle for the material to be used next is run through a cleaning station made of

Table 4.3: Temperature Profile for Organic Burnout and Sintering

Ramp [°C/hr]	Temperature [°C]	Hold [hr]
180	100	1
60	250	4
60	350	0
180	900	2
600	1300	2
600	400	0

a coarse brush to dislodge any debris.

4.3.2 Scaffold Post Processing and Evaluation Tools

Fabricated scaffolds are dried in air for 24 hours then sintered (Carbolite CWF 13/13/3216P1) with the programmed temperature profile given in Tb. 4.3. Sintered scaffolds are evaluated with a suite of visualization tools: optical imaging (Canon EOS-1Ds Mark III SLR Digital Camera), SEM, and μCT. Image processing and evaluation are performed with Adobe® Photoshop CS 8.0 and Amira® 5.3.0.

4.4 Conclusion

This chapter detailed all fundamental aspects of the μRD system except the control of the material flowrate. Chapter 5 will demonstrate the application of BTILC to the control of material flow for the μRD system given here. Importantly, it will demonstrate how a system that satisfies certain criteria, Assumptions 1 and 2 from Section 3.2.2, can be reframed in the context of BTILC and accurately controlled. Many of the developments of this chapter will be directly used in the process control; the machine vision algorithm (Section 4.1) is fundamental to the measurement of the flowrate response and the dynamic models developed in Section 4.2.2 will be directly

used in the control formulation.

Chapter 5

BTILC Application to μRD

5.1 Introduction

In this section, we apply the general BTILC algorithm in Chapter 3 specifically to the μRD system given in Chapter 4. The objective here is to accurately modulate material flowrate such that the fabrication of advanced architecture structures is enabled. Specifically, this means the ability to precisely start and stop material flow and maintain a consistent flowrate. The standard material flowrate control scheme for μRD is to fix the input Q_{in} as a static proportion of the desired output Q_{out} [49]; performance is poor and is analogous to the flowrate response in Fig. 4.10. A small experiment demonstrates inadequate flowrate modulation when attempting to fabricate a simple structure with on/off type control, Fig. 5.1. *Ad hoc* tuning methods have been attempted on similar systems [61, 62]. However, none of these control methods provide the material placement accuracy required to construct advanced architecture structures by μRD.

The μRD system is reframed in the context of BTILC, Chapter 3, to achieve the control objective. Fig. 5.2 demonstrates the reframing of the system. Compare Fig. 5.2 with Fig. 3.4 of Chapter 3. All structures are built with a material flowrate profile that can be decomposed into a coordinated sequence of *basis tasks*, described here as *Start*, *Steady-State*, *Corner*, *Stop*, and *No-Flow*. Additionally, a fully flexible μRD system accommodates multiple materials and nozzle sizes, each having distinct dynamics and therefore different *configurations*. Here we consider two materials, m_1

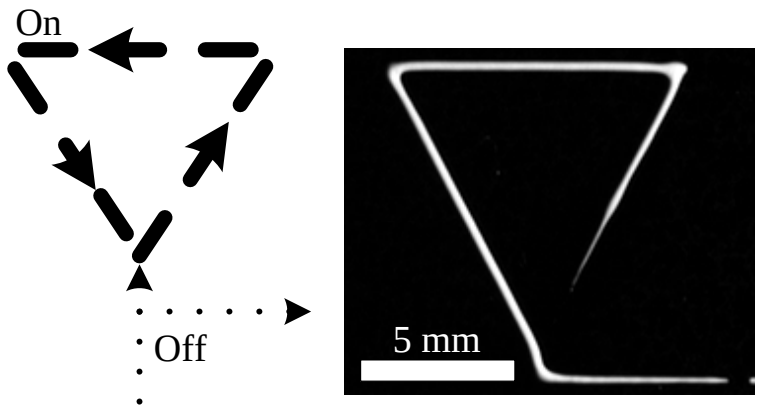


Figure 5.1: Extrusion performance under typical on/off control schemes. The long time delay and slow time constant of the flowrate response precludes the system from accurately fabricating a closed triangle.

and m_2, detailed in Chapter 4, and one nozzle size, corresponding to *configurations* H_1 and H_2. This gives the *operation space* and *dynamic space* shown descriptively in (5.1). Signal and systems representations of these descriptions are given in Fig. 5.3 and Eq. (4.10), respectively.

$$\begin{aligned} R^O &= tspan\,\{Start, Steady{-}State, Corner, Stop, No{-}Flow\} \\ D^O &= \{H_1, H_2\} \end{aligned} \tag{5.1}$$

BTILC is applied entirely in open-loop because a real-time volumetric flowrate sensor at this length scale does not exist. Instead, volumetric flowrate is calculated offline by a machine vision system [36] in-between ILC signal updates in the *training set*. All assumptions required for BTILC are satisfied, Section 3.2.2.

5.2 Training Set

1. *Training Reference Signal Selection*

As specified by (3.4), a set of training references, $r^T(k)$, is selected which encompasses all tasks in R^O. Here we have chosen a single reference trajectory that is a

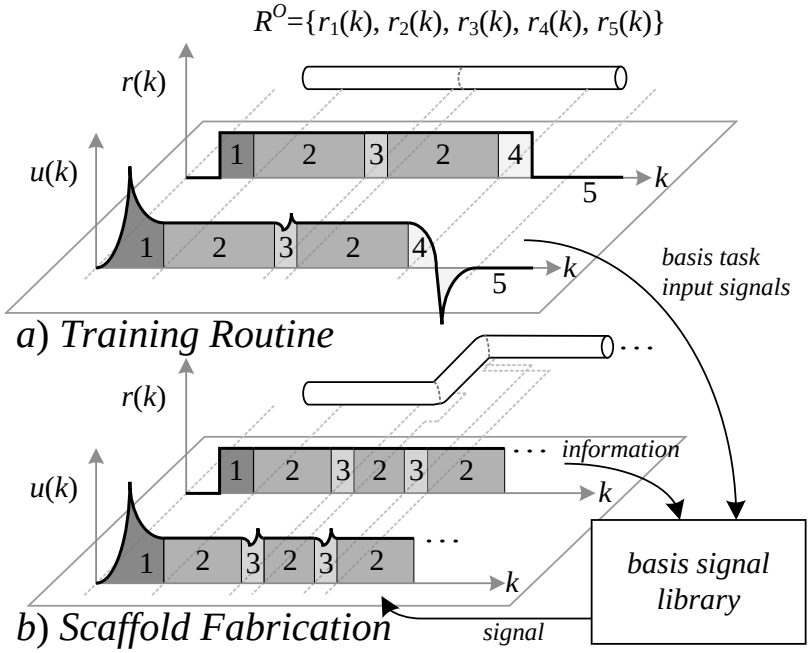


Figure 5.2: BTILC reframed in the context of the μRD system. Compare this figure to Fig. 3.4 of Chapter 3.

pulse input with a momentary dwell in the positioning system velocity in the middle of the pulse, $r^T(k)$ in Fig. 5.3. Material is deposited on a substrate at a constant positioning system velocity, $v(k) = 5\ mm/sec$, for most of the training routine. The upward step in the volumetric flowrate reference, $r^T(k)$, is representative of a *Start* task and likewise the downward step is representative of a *Stop* task. The momentary dwell in positioning system velocity, $v(k)$ in Fig. 5.3, at the $k = 7000$ time index is representative of the positioning system velocity deceleration and acceleration profile for a *Corner* task. Instead of making a corner, the momentary dwell follows the exact velocity profile of a corner but proceeds in a straight line to make the flowrate measurement with machine vision easier. The nominal flowrate section of $r^T(k)$ is representative of the *Steady-State* task and the zero flowrate is representative of the *No-Flow* task. The ideal fabricated shape is a thin cylinder.

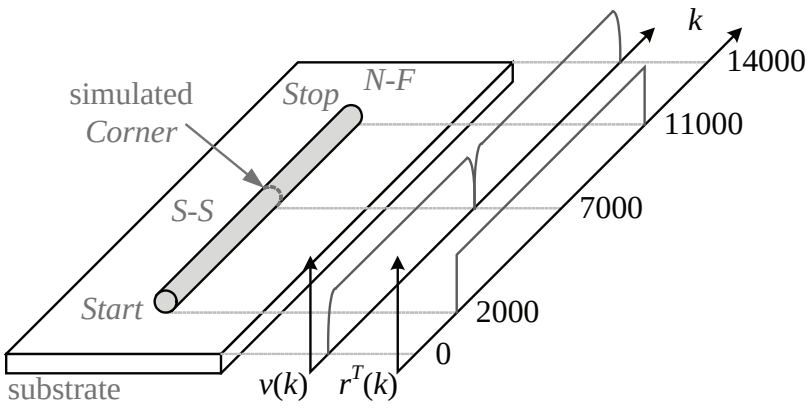


Figure 5.3: Training routine schematic. (*S-S* = *Steady-State*, *N-F* = *No-Flow*).

5.2.1 ILC Application

ILC algorithm (3.7) with a plant inversion learning filter is applied to the μRD system for two different *configurations*, H_1 or H_2, with the training routine in Fig. 5.3. The plant inversion learning filter [63] has the form $L(q) = k_p \hat{H}_d^{-1}(q)$ where k_p is a proportional gain and $\hat{H}_d^{-1}(q)$ is an inverted and modified version of H_d in (4.10). H_d is modified by adding a fast zero to the numerator to make the inversion proper. k_p is selected to be 0.3 and $Q(q)$ is selected to be a second-order low-pass Butterworth filter with a bandwidth of 4 Hz. $L(q)$ and $Q(q)$ are selected based on previous work [36], however, alternative filters are viable options.

5.2.2 Results

The results from applying ILC algorithm (3.7) to $r^T(k)$ are shown in Figs. 5.4 – 5.7. Iteration 0, with both *configurations*, displays the poor performance achieved when using the reference signal as the input. System performance improves considerably over the course of 25 iterations, with the algorithm able to achieve tracking within 4 - 5 times the resolution of the sensor, Fig. 5.4 and Fig. 5.5, for both *configurations*. As shown in Fig. 5.6, both systems converge to a root mean squared (RMS) error that is less than 20% of the RMS error at iteration 0. RMS is defined as:

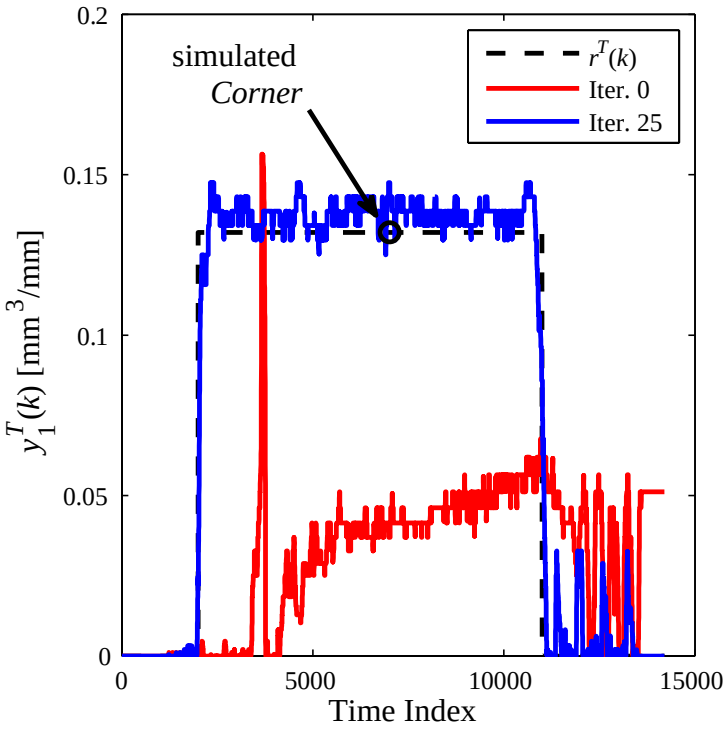


Figure 5.4: Time domain *training set* results at select iterations for *configuration* H_1.

$$\text{RMS}\left(e(k)\right) = \sqrt{\sum_{K} e^2(k)/K}. \tag{5.2}$$

The noticeable difference between *configurations* is revealed in the input signals identified by the ILC algorithm, Fig. 5.7. Material m_1 is more viscous than material m_2, causing H_1 to have a slower system response and consequently $u_1^T(k)$ is identified to have approximately 1.25 times the input magnitude as $u_2^T(k)$ at the input signal spikes corresponding to the *Start* and *Stop* tasks.

5.3 Signal Segmentation

Both $u_1^T(k)$ and $u_2^T(k)$ contain the individual *basis signal* information for *configurations* H_1 and H_2, respectively. In the case of μRD, each *basis signal* is demarcated

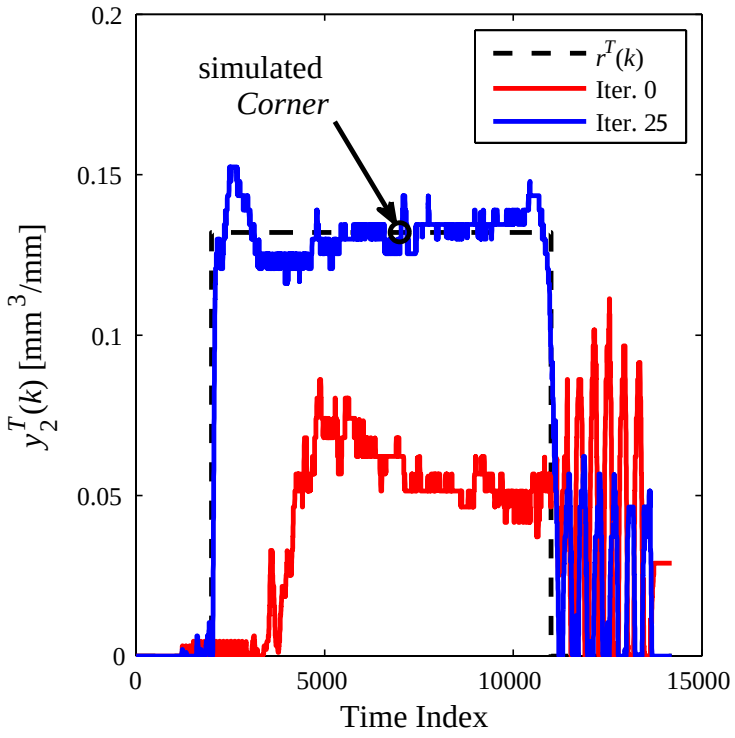


Figure 5.5: Time domain *training set* results at select iterations for *configuration* H_2.

by distinct transitions in signal magnitude. $u_1^T(k)$ and $u_2^T(k)$ are segmented at these demarcation points for storage into the *basis signal library*. Fig. 5.8 shows the *basis signal* domains segmented from $u_1^T(k)$. $u_2^T(k)$ is segmented at the same time indices, not shown. *Start*, *Stop*, and *Corner basis signals* are of finite length because these tasks occur at a single location. The *Steady-State* and *No-Flow basis tasks* are regulation tasks and therefore the *basis signals* are set to a constant value. The *Steady-State basis signal* is set to the mean input magnitude during the steady-state region. The *No-Flow basis signal* is set to zero. Each $u_{n,d}(k)$ along with their lead, $lead_n$, and signal length, K_n, is stored in the *basis signal library* with the data structure given in (3.9).

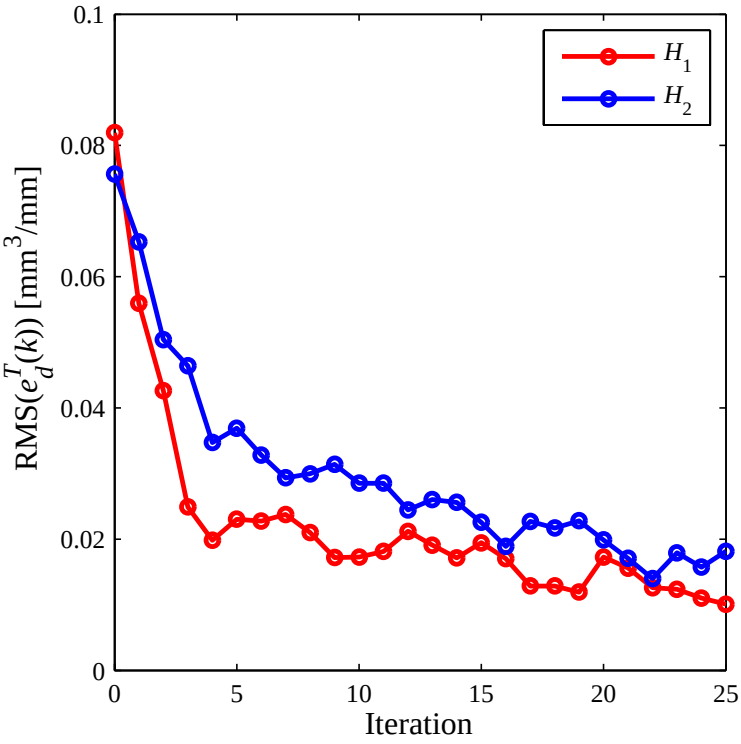


Figure 5.6: Iteration domain *training set* results for *configurations* H_1 and H_2.

5.4 Signal Extraction

Fabrication instructions for the μRD system use a custom command structure that is similar in form to G-Code [47]. Alongside a command interpreter, a *basis signal* extraction algorithm translates the sequence of commands into an appropriate *basis signal* selection. The *basis signal* extraction logic is similar to the general logic structure outlined in Section 3.3.3. The work here enforces a hierarchy with order: {*Stop*, *Start*, *Corner*, *Steady-State*, *No-Flow*}. Fig. 5.9 displays the set of allowable task sequences for a two material μRD system.

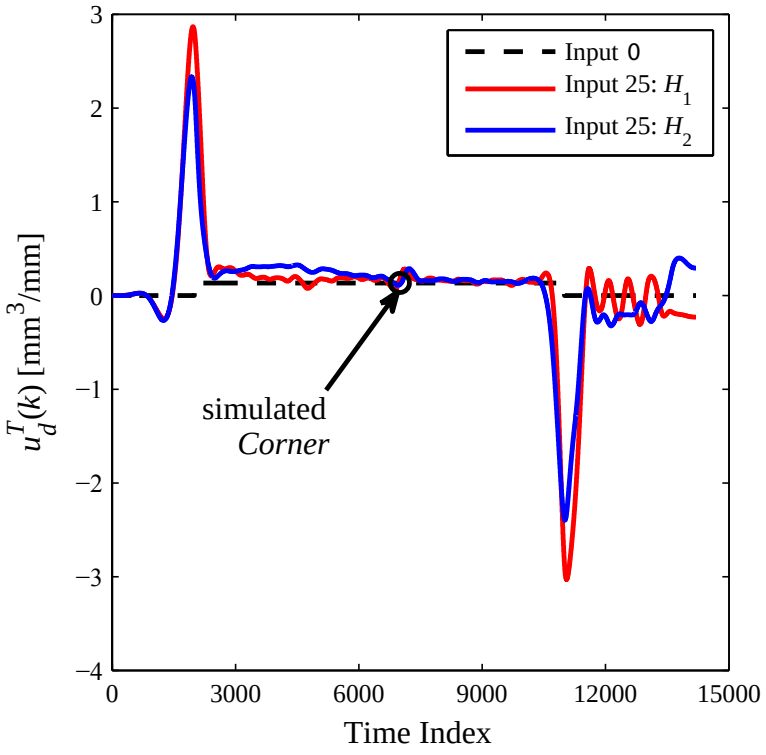


Figure 5.7: Input signal identified for the 25^{th} *training set* iteration for *configurations* H_1 and H_2.

5.5 Operation Set Examples

This section displays three example manufacturing operations using BTILC. All *operation sets* are *set equivalent* with the *training set*. It is important to note that the *basis signals* were not re-identified in-between operations. The only distinction between each manufacturing operation is the machine instructions detailing the architecture of the structure.

5.5.1 Basis Signal Extraction: T + O = T

Here we show a simple example demonstrating the coordinated sequencing of extracted *basis signals*. The machine instructions command the μRD system to manufacture the U-shaped structure diagrammed in Fig. 5.10 with material m_1. The

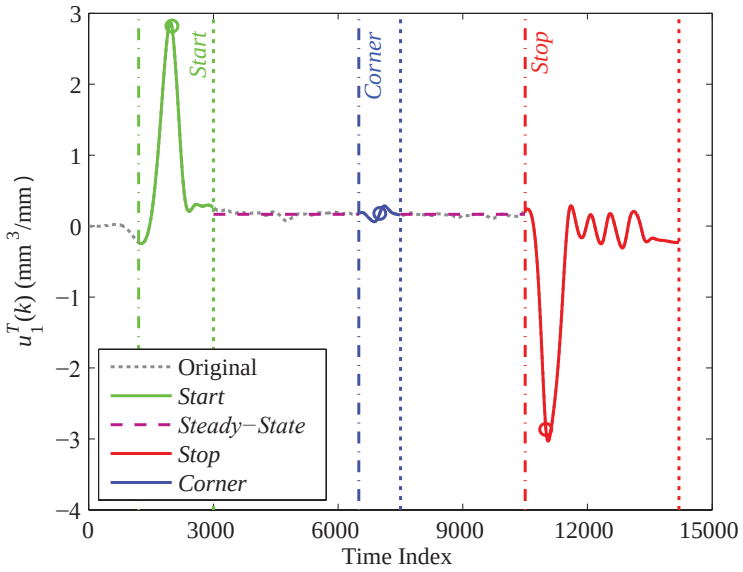


Figure 5.8: Signal segmentation of $u_1^T(k)$ from Fig. 5.7. Vertical lines demark the beginning and end of a *basis signal*. Circles denote the time at which the *basis task* is prompted to occur.

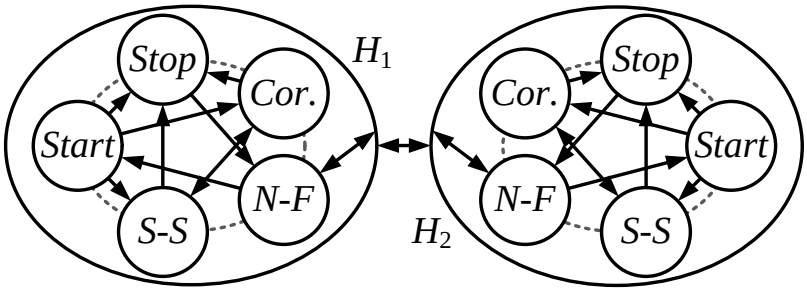


Figure 5.9: Automaton for a two material μRD system represented by *configurations* H_1 and H_2. (S-S = *Steady-State*, Cor. = *Corner*, N-F = *No-Flow*).

appropriately sequenced *basis signals* are shown in Fig. 5.11. The resultant fabricated structure, shown in Fig. 5.10, deviates from desired architecture on average by less than 50 μm and at the maximum less than 250 μm.

5.5.2 Extended Manufacturing Operation: T + O = T

The second example displays the utility of BTILC in lengthy operations. The tic-tac-toe structure diagrammed in Fig. 5.12 requires a build time over 3 *min* in duration,

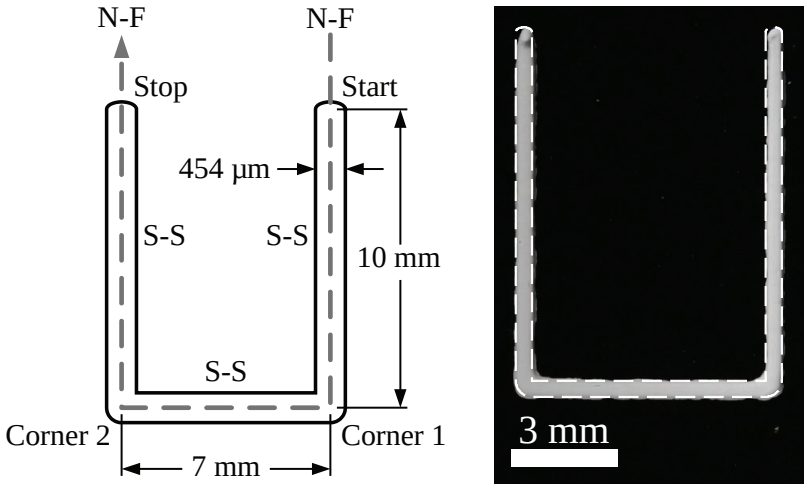


Figure 5.10: Manufacture of a simple U-shaped structure. Structure architecture as well as location of executed *basis tasks* is diagrammed in the left figure. The fabricated part is shown in the right figure. Superimposed on the fabricated part is an outline of the desired part based on proportions between known distances and image pixels. (S-S = *Steady-State*, N-F = *No-Flow*)

which is prohibitively expensive for applications that utilize data-rich sensors like the machine vision system on the μRD. BTILC allows this 3 *min* operation to be accurately controlled using only the information gained in a 14 *sec* identification routine. The manufactured structure is shown in Fig. 5.13. The structural accuracy is improved compared to any published flowrate modulation technique for μRD [64]. There is some excess material at most *Start* task locations from slight material buildup at the nozzle tip after the previous *Stop* task. However, these manufacturing defects could be eliminated with a simple pass through a nozzle cleaning station.

5.5.3 Multiple Configurations: $\mathbf{T + O = T}$

This example displays that the introduced BTILC is adept at controlling systems with multiple *configurations*. The structure is a two-layer 'Block I' lattice containing domains of materials m_1 and m_2, Fig. 5.14. Design specifications such as the line-to-line spacing and material selection were chosen for use in synthetic bone grafts. The medical community demands synthetic bone scaffolds that are strong, mimick-

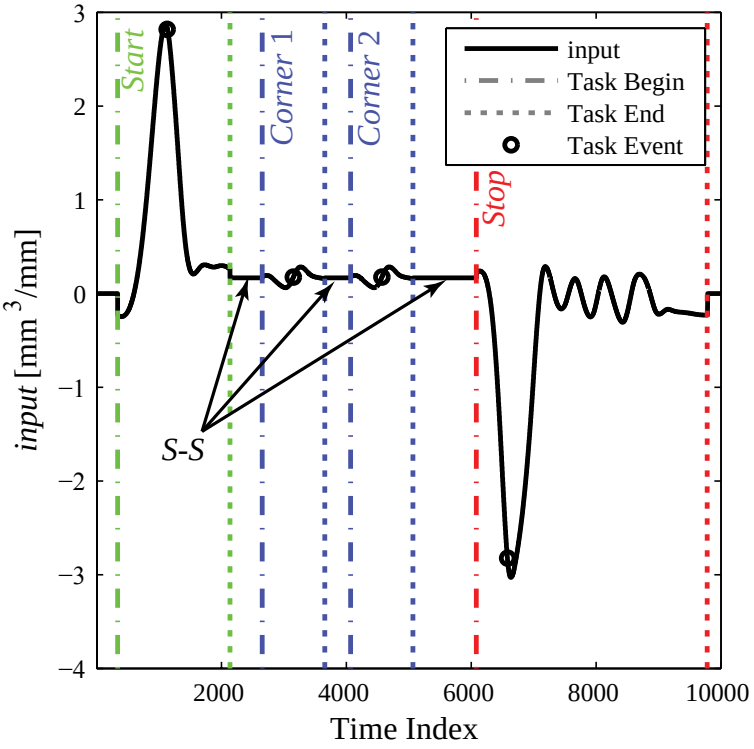


Figure 5.11: Sequence of *basis signals* applied in the manufacture of the structure in Fig. 5.10. Vertical lines demark the beginning and end of each *basis signal*. Circles denote the time at which the *basis task* is prompted to occur. (S-S = *Steady-State*).

ing material properties of natural bone, and highly porous, to allow cell infiltration, vascularization, and bone growth [65]. m_1 has a dense microstructure thereby resulting in a higher material strength. m_2 contains open and interconnected porosity, a microstructure which has been shown to promote bone cell growth as compared to the same material with a dense microstructure [13, 24].

The ability of BTILC to control multiple system *configurations* without reinitiating the ILC algorithm is a unique ability. Material distinctions are not discernable from the optical image in Fig. 5.14, but are clearly shown in the reconstructed μCT image of the structure in Fig. 5.15. Shown is a section through the second layer of the 'Block I' lattice displaying that the dense m_1, lighter lines, attenuates x-ray transmission more than the porous m_2, darker lines. The material distinctions are further

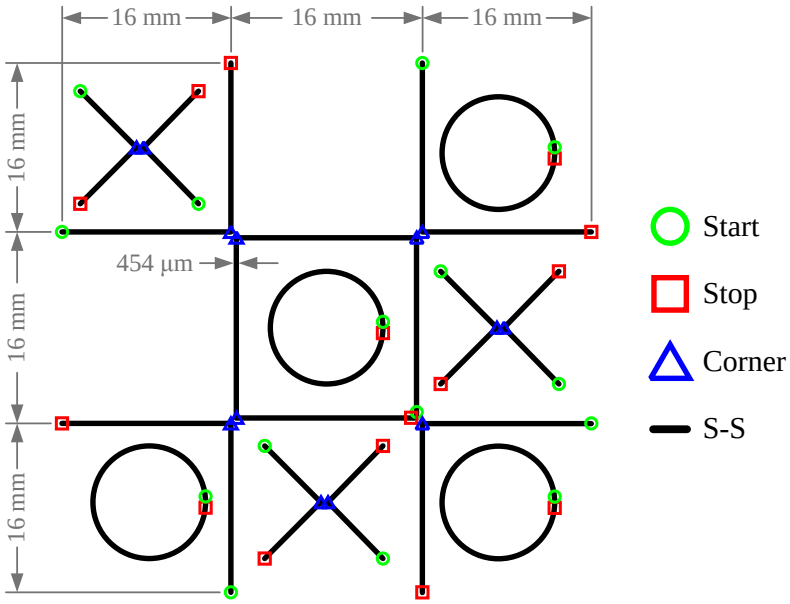


Figure 5.12: Diagram of *basis tasks* required to build the tic-tac-toe structure in Fig. 5.13

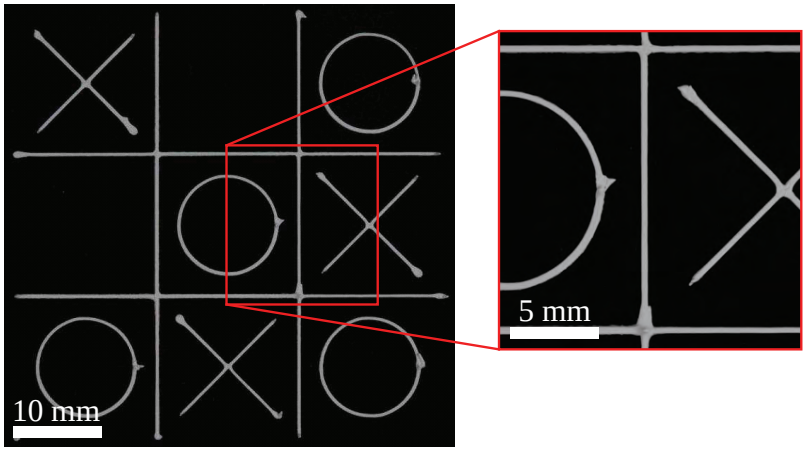


Figure 5.13: Fabricated tic-tac-toe structure.

elucidated in an SEM image of the surface topography of the two materials, Fig. 5.16. The surface of m_1 has small pores from incomplete sintering. Comparatively, m_2 has both more porosity and the interconnection sizes are larger than m_1.

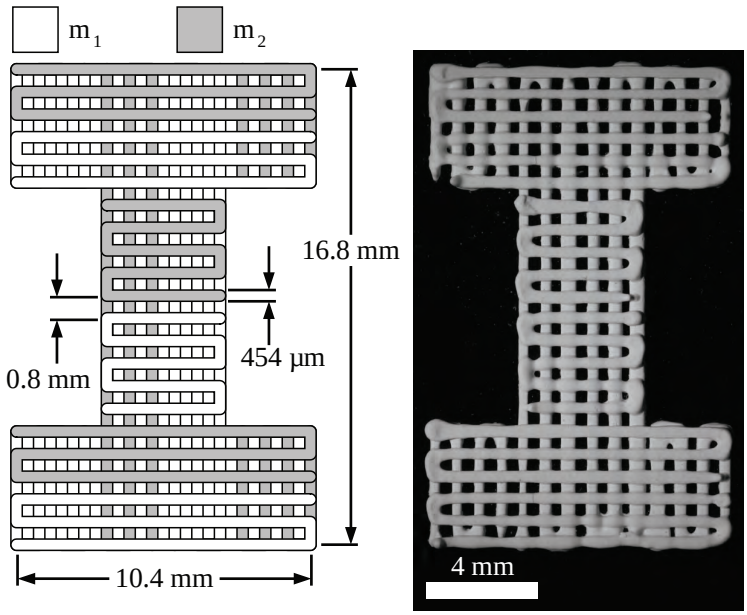


Figure 5.14: Representative image of a fabricated multi-material 'Block I' structure. Materials m_1 and m_2 have different extrusion dynamics. Each *basis signal* is applied based on the desired *basis task* and build material.

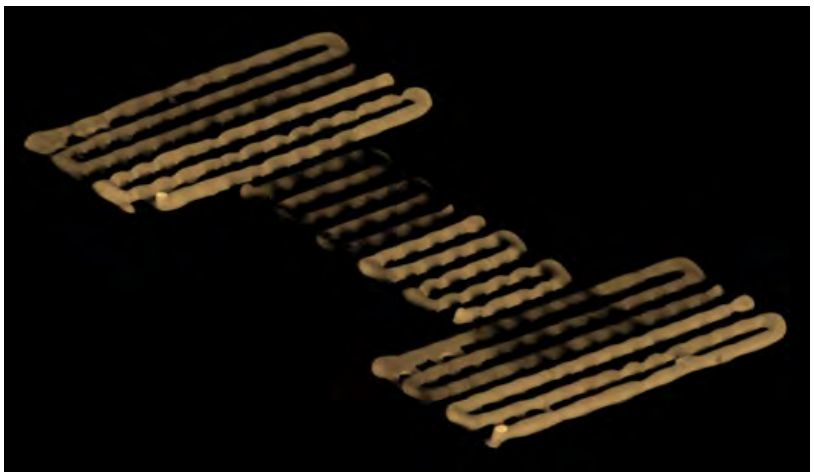


Figure 5.15: Reconstructed μCT image of the second layer of the structure in Fig. 5.14. Light lines denote areas of higher x-ray attenuation from the dense material m_1. Dark lines denote areas of lower x-ray attenuation from the porous material m_2.

5.5.4 Discussion

To the best of the authors' knowledge, the multi-material 'Block I' structure in Fig. 5.14 is the first bioceramic bone scaffold fabricated with multiple domains of different microstructures. In a broader scope, it is the first structure to be fabricated by μRD

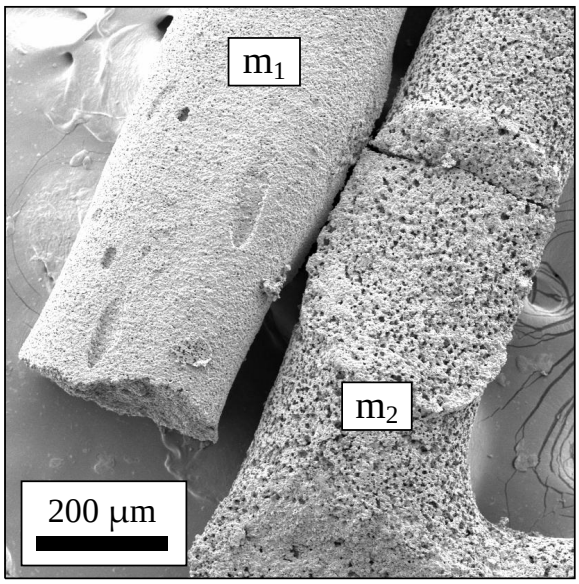


Figure 5.16: SEM image displaying the topography of m_1 and m_2 in a representative scaffold.

without lead-in lines; an extraneous feature that allows flowrate to fully develop. These are two technological achievements that will facilitate advancements on two fronts: 1) on the bone scaffolding front, bone scaffolds with graded microstructures, much like natural bone, are now realizable, 2) on the general μRD front, advanced architectures with internal features, near-net shapes, encapsulated functionalized materials, and direct abutments of different materials are now realizable since the control scheme developed enhances material placement selectivity. The demand for advanced architecture structures by μRD has been well articulated in [64] and [67]. These manufacturing advancements are possible because BTILC is flexible in that all the requisite tasks can be learned in the *training set* and applied with relaxed trajectory constraints in the *operation set*. A final note is that the structure in Fig. 5.14 has an almost 3 minute build time; yet the complete control input was extrapolated from information retrieved in a 14 second training routine. The scalability of the proposed method highlights an important new way to more efficiently use computationally intensive ILC algorithms.

Chapter 6

Advanced Architecture Scaffolds

6.1 Introduction

This chapter applies the BTILC algorithm detailed in Chapters 3 and 5 to the manufacture of advanced architecture structures. We aim to fabricate synthetic bone grafts with designed features on all three design space levels (envelope, macroporosity, and microporosity), Fig. 1.4. Here, we explore fabrication capabilities with a few canonical structures that demonstrate a good foundation for progressing towards anatomically derived bone scaffolds. The chapter is divided into two halves. First, we detail the development of combinatorial test scaffolds that demonstrate the ability to integrate multiple macroporosity and microporosity domains within a single structure to efficiently evaluate a range of scaffold design parameters. The second half demonstrates near-net shape fabrication, detailing structures fabricated with contoured scaffold envelopes, designed internal features, and integrated regions of different microporosities. Motivations for both halves of this chapter is given in the next two subsections, 6.1.1 and 6.1.2. The scaffold manufacturing process, an extension of Chapter 5, is given in Section 6.2. The designs and results for the combinatorial and near-net shape halves then proceed in series, Sections 6.3 and 6.4, respectively. The chapter concludes with the implications of the new fabrication capabilities in Section 6.5.

6.1.1 Combinatorial Test Sample Design Motivation

Combinatorial arrays, such as the microwell plate, have long been a tool for efficient evaluation of cell expression in analytical biological research [68]. Reagents or cell type can easily be selected and isolated in the wells and therefore combinatorial means to fully evaluate a response are a natural choice. synthetic tissue scaffold research has been relegated to test individual, one-to-one comparisons, to evaluate *in vivo* efficacy because there is a lack of manufacturing means to create scaffolds with multiple designed regions. Existing scaffold manufacturing methods are best suited to fabricate uniform structures from a single material [3]. Here, we introduce scaffold designs for efficient evaluation of multiple pore sizes and material compositions within a single scaffold.

The influence of scaffold microporosity on bone formation warrants further investigation. We have designed a test sample to efficiently evaluate HA scaffolds with multiple micro- and macroporous treatments within a single structure, termed the combinatorial test sample. Our design is selected for two reasons. The first is that *in vivo* studies are expensive, labor intensive, and ethically sensitive. We propose to evaluate the span of porosity treatments with as few animals as possible. The second is that the different micro- and macroporous treatments in a combinatorial test sample are colocated within a single defect site. We postulate that the colocation of treatments will reduce the confounding of results due to anatomical location variability and surgical procedural variability.

6.1.2 Near-Net Shape Scaffold Design Motivation

The human skeletal structure has complex features, including contoured features, anisotropies, and hollow marrow cavities. The complex anatomy of the human skeleton presents a considerable challenge to tissue engineers attempting to recreate nat-

ural tissue function with a synthetic structure. In terms of AM processes to build synthetic bone scaffolds, replicating the natural anatomy requires: 1) accurate placement of build materials and 2) the fabrication of unsupported features which must be supported during the layer-by-layer build routine.

Chapter 5 demonstrated the ability to accurately place build materials of varying material properties. The results in this chapter will further demonstrate this ability; however with a considerable advancement in structure complexity. The key contribution of the near-net shape half of this chapter is the ability to fabricate unsupported features. First we must define what constitutes an unsupported feature. For one, a macroporous structure will have unsupported features throughout; each of the macropores is unsupported over the filament span, Fig. 1.5. As long as the span between filament attachment points is less than 2 mm, for the filament diameters relevant here, the filament will maintain its shape and material flow will not be impeded [49]; unsupported macroporous features are easily achievable, represent a considerable advantage over some other AM technologies, and have been demonstrated throughout the literature [3, 49, 51, 69]. For our aims, the unsupported features of interest are those greater than 2 mm in unsupported span; these require modifications to the typical μRD fabrication routine and we will term these large unsupported features.

There are two feasible methods to enable the fabrication of large unsupported features in a structure. One method, commonly utilized in polymer-based AM processes such as FDM, Section 1.1,is to integrate sacrificial material regions into the build routine to support the large unsupported features [15]. Post-process the sacrificial material is removed and the desired structure remains. Commonly the sacrificial material is removed by dissolving in water or by fracture at a weakly adhered delamination interface. The HA build material is water soluble and very weak and susceptible to cracking in its 'green' state before sintering; therefore sacrificial materials that dissolve or delaminate are not a good solution. Another sacrificial material

option is a sacrificial polymer [70]; this option would exploit the low glass transition temperature of polymers, relative to ceramics, to easily remove the polymer without affecting ceramic sintering. Interfacing polymeric and ceramic materials is a considerable challenge. To maintain fabrication accuracy, the fluid viscosities must be carefully matched so that a ceramic material deposited onto a polymeric material does not deform the polymer and vice versa. Additionally, the drying kinetics must be matched so that all materials shrink at the same rate and do not induce structural crack formation during drying. Given the vastly different material characteristics of polymeric and ceramic particles, and therefore different colloidal synthesis methods, matching these viscoelastic and drying kinetic properties is a difficult task.

This chapter presents a second method to build large unsupported features that still maintains the ability of μRD to integrate different materials within a single structure. In this two-step method, we build within a cured polymer mold manufactured by SLA. This method ensures that large unsupported features are supported and that the fundamental build routine specified in Chapters 3 and 5 is unaltered. The molds fabricated by SLA are inexpensive ($\approx$\$5 each) and can be manufactured in just hours; thereby not significantly adding to the process cost or impeding workflow progress. Critically, the method permits many of the conceivable features for a bone scaffold to be fabricated with the main exception being unsupported features with blind features beneath them. The set of feasible features will be discussed in more detail in Section 6.5.

6.2 Methods

6.2.1 Manufacturing Platform

Scaffolds are built using the nozzle-based AM method μRD detailed in Chapter 4. Material extrusion is controlled using the BTILC algorithm detailed in Chapters 3

and 5.

6.2.2 Scaffold Manufacture

Chapter 5 details a manufacturing work flow that assumes no knowledge of the system to be controlled. However, after BTILC has been run once, there is information on hand that can be used to streamline the work flow. Fig. 6.1 demonstrates a streamlined work flow that uses previous ILC information so that an operator does not have to re-initiate the ILC process; that is they can start the ILC process from an iteration number greater than zero. This streamlined work flow is given in explicit details in Appendix Section C.9.

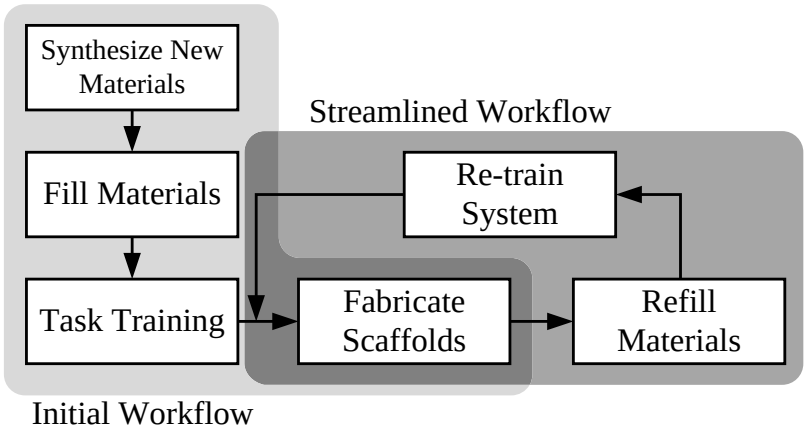


Figure 6.1: Workflow for the manufacture of combinatorial test scaffolds.

Given a newly synthesized set of build materials, build materials are loaded into 5 mL syringes and fitted with an appropriately sized nozzle. *Basis task* information is identified in an initial training routine without previous information stored in the basis signal library. After a sufficient number of training iterations, ≈20, input signals are stored in the *basis signal library*. The manufacturing operation is then run with the μRD system. All scaffold fabrication is performed in an oil bath that is filled from the bottom by a gravity fed system. The oil prevents earlier deposited layers from drying before later deposited layers, thereby preventing stress gradients. The oil

level is regulated to be just below the most recently added layer and capillary forces from the porous structure draw the oil to the surface of the scaffold, immersing the structure.

Scaffolds are deposited until all materials in the syringes are exhausted. Syringes are refilled with material, however slight inconsistencies between syringes and the fit between the plunger and the syringe walls reduce the performance achievable from information in the *basis signal library*. Fortunately, the information stored in the *basis signal library* is still useful. Previous library information is used as the initial information in the training routine and the system can be refined to where performance equivalent to that achieved in the initial workflow is achieved after 5 iterations.

This sequence is shown in Fig. 6.1, where after a relatively labor intensive initial signal identification, the workflow is streamlined and there is little expense to fabricate subsequent scaffolds. Based on user estimates, the streamlined workflow cuts labor time by three fourths and a two material system can re-trained in about an hour. The RMS of the error for a representative initial training and then re-training is shown in Fig. 6.2. The continuity between the initial training and re-training will depend on the material system and the consistency of the training procedure.

6.2.3 Fabricating Scaffolds Within Molds

The designs in Section 6.4 utilize molds that are designed to support large unsupported features. Fig. 6.3 demonstrates the additional steps in the work flow for Design 7. These steps are identical for Designs 5 and 6 as well. The mold designs we propose have small capillaries permeating the molds to permit the oil to reach the scaffold. Molds are affixed to the well of the oil bath using clamps. A registration feature is used to locate the mold coordinates in the μRD reference frame. Designs 5 – 7 are all symmetric about the Z-axis and only need one registration point; non-symmetric

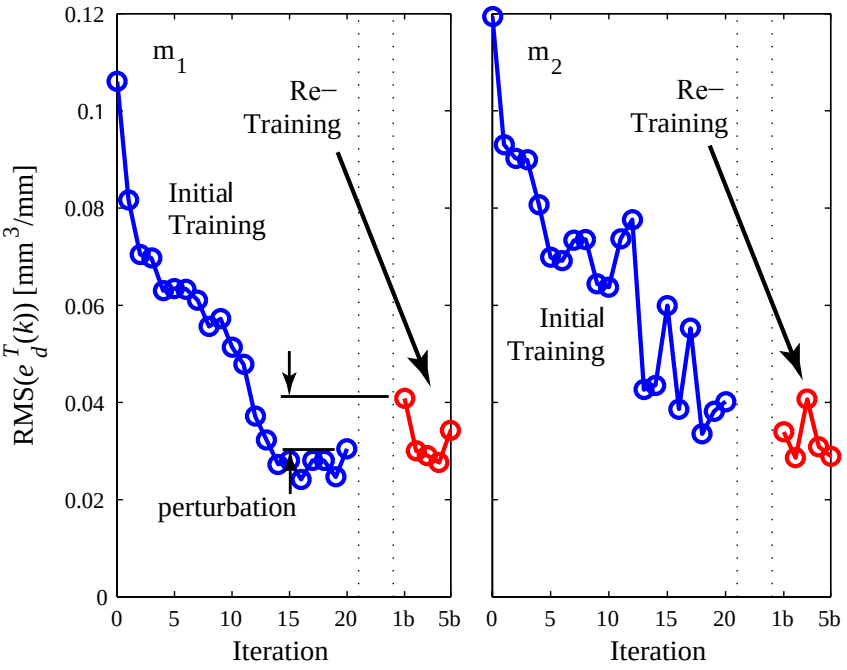


Figure 6.2: RMS of the error signal for the initial training and re-training for materials m_1 and m_2. The performance level of after 20 iterations of the initial training can typically be achieved after 5 iterations of re-training using the information already in the *basis signal library*.

designs will need two-registration points and an additional coordinate transformation step to correct for a mold not perfectly in-line with the μRD axes. Once the proper registration is calibrated, the mold is filled using the typical work flow shown in Fig. 6.1. After fabrication, scaffolds are dried in air for 24 hours; convex features will pull away from the mold as the scaffold dries and concave features will tighten on the mold. The issue of constrained drying of concave features is discussion in Section 6.5. Scaffolds are removed from the mold and sintered using the normal protocol, Section 4.3.2.

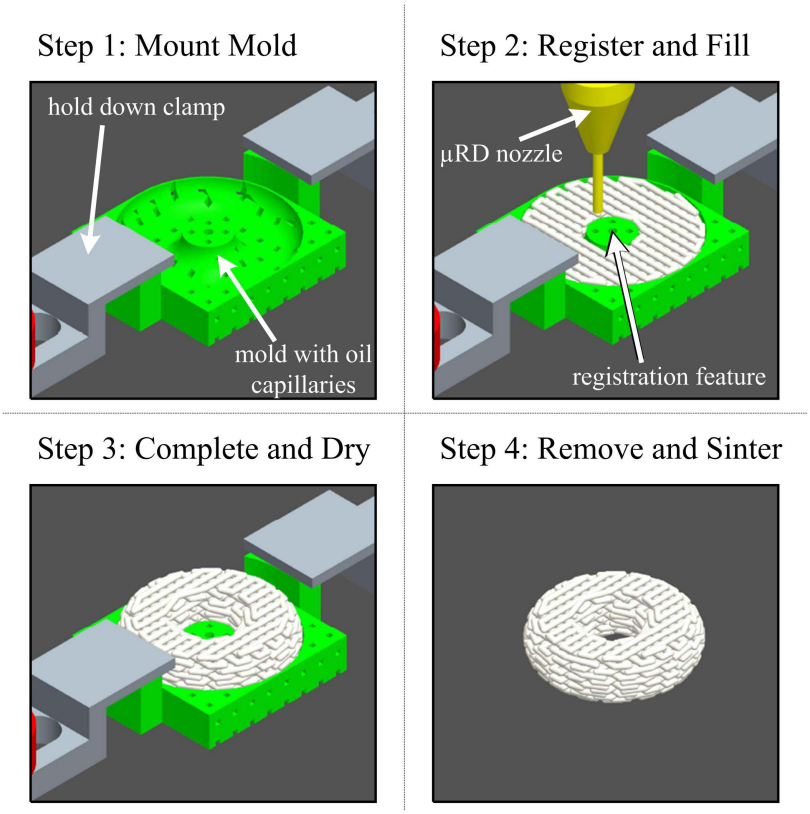


Figure 6.3: Demonstration of Design 7 being fabricated within a mold.

6.3 Combinatorial Test Samples

6.3.1 Scaffold Designs

The combinatorial test sample is a tool for biologist and biomechanics researchers to evaluate the *in vivo* response to an array of mechanical designs. Our purpose here is to demonstrate that the developed tool is capable of spanning a range of porosities of interest. The range can be visualized as a three dimensional space, Fig. 6.4, in which we demonstrate that a range of different macroporous and microporous designs are reachable. Macropore interconnection designs are given by an uppercase 'M' and microporous designs are given by a lowercase 'm'. Region communication denotes whether there is a direct path for fluid transport between designed scaffold regions. Here, we select four specific designs that sufficiently demonstrate fabrication

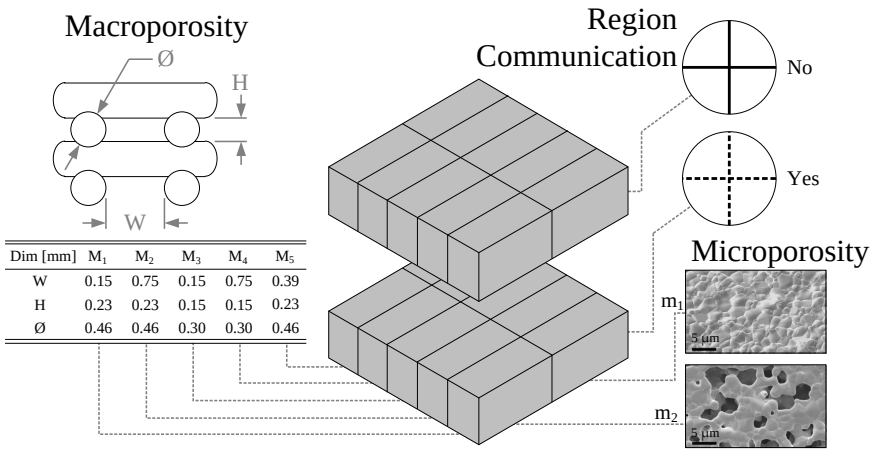
feasibility over the range in Fig. 6.4 and we provide their pore dimensions. Fig. 6.5a provides the different pore designs in each region and illustrates region communication for each design. Designs 1 – 3 assess bone ingrowth quality in four geometrically identical quadrants; we postulate that each of these quadrants will be exposed to identical *in vivo* conditions and any difference in bone ingrowth quality can be attributed to quadrant architecture. Design 4 is a concentric design to assess bone ingrowth profiles as a function of material microporosity. Bone grows into a scaffold implanted into a defect site from the periphery [9] and we aim to investigate methods to expediently grow bone in internal regions.

More specific architectural properties are given in Fig. 6.5b. Region communication is clarified in Fig. 6.5b, where a wall of build material prevents fluid flow in non-communicating scaffolds and where there are open channels across regions in communicating scaffolds.

Major scaffold dimensions are designed such that they can be easily tested *in vivo* in the general animal model; however the target clinical applications are oral and maxilliofacial surgery and therefore scaffolds are designed to be implanted in the ramus of the mandible of pigs. Scaffolds are cylindrical for insertion into drilled cylindrical defects and the height is approximately the thickness of a bicortical defect in the ramus of an adolescent pig or adult minipig mandible (7-8 mm), Fig. 6.5b. The following subsections provide rationale for the designs selected.

Macroporosity

The range of macroporosities selected for the four designs span a subspace of the range commonly cited as necessary for bone ingrowth. The range is broad (150 - 750 μm), however the literature does not state a definitive optimal interconnection size, in terms of bone ingrowth rates and steady-state bone volume, across this range when a single material composition is used [24]. Whether or not there is an optimal macroporosity



Dim [mm]	M_1	M_2	M_3	M_4	M_5
W	0.15	0.75	0.15	0.75	0.39
H	0.23	0.23	0.15	0.15	0.23
Ø	0.46	0.46	0.30	0.30	0.46

Figure 6.4: Representation of the span of design options demonstrated. The reachable set of macroporosities and microporosities are a continuous distribution whereas region communication is binary (represented by connected and disconnected sets, respectively).

when the material composition has a high level of microporosity is an open question. Within this macroporous range, Design 4 evaluates the macroporous design used in the *in vivo* studies detailed in [8,9,13] and will provide a direct comparison to previous work. We did not demonstrate the maximum stated pore interconnection size (1 mm) because scaffold strength is compromised when void sizes approach this limit. Any macropore interconnection size within the 150 - 750 μm range is readily achievable by properly positioning the manufacturing toolbit in space and using the developed BTILC algorithm, Chapters 3 and 5. The macropore space is also defined by the filament diameter, Ø, chosen for the structure. Here two filament diameters, 300 and 460 μm, are chosen to demonstrate flexibility in design options. Filament diameter is an important determinant of scaffold stiffness.

Microporosity

Two microporosities, 0 $vol\%$ and 50 $vol\%$ nominally (m_1 and m_2, respectively), were chosen to match previously tested microporosities in *in vivo* studies in [8, 9, 13]. Any

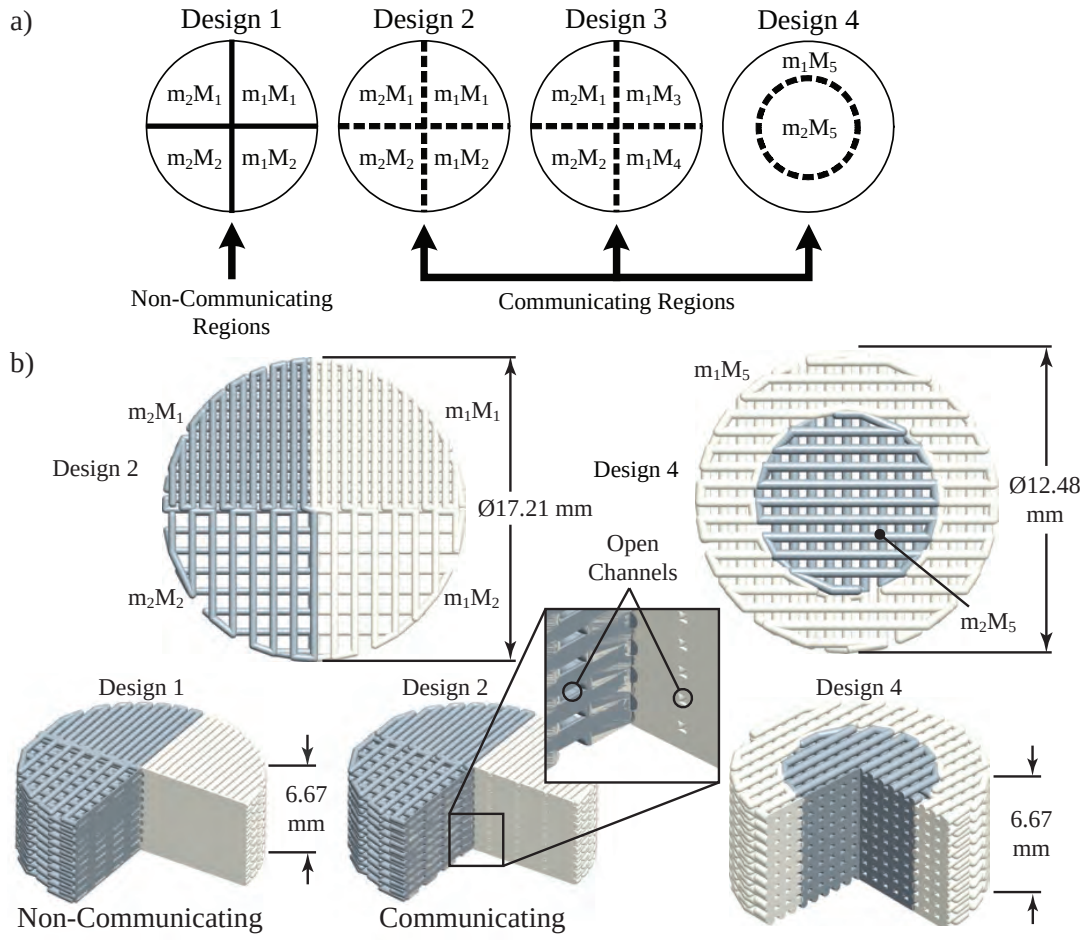


Figure 6.5: Combinatorial test scaffold designs. a) Diagram of the designs fabricated in this study. Specifications are given for each region and interface. b) Top views display specific features in Designs 2 and 4. Cross-section views display the region interfaces for Design 1 (non-communicating), Design 2 (communicating), and Design 4 (communicating). The region interface is completely walled off in Design 1, whereas there exist open channels for fluid transport across the region interfaces in Designs 2 and 4.

material composition within the 0 - 50% range is readily achievable by including an appropriate amount of a pore forming agent during materials synthesis [58]. Looking forward to more advanced scaffold designs, tailoring the microporosity enables the local anatomy to be matched with higher fidelity and it will be of interest to have scaffolds containing many discretized composition regions.

Region Communication

Two boundary conditions at the region interfaces are tested. One in which the regions are completely isolated by a solid wall of build material and the other in which there are open channels for fluid transport between all the regions. Equally feasible are designs in which some regions communicate and some do not. The ability to regulate communication between regions provides an experimental control in which we may be able to assess whether nearby dissimilar mechanical environments affect bone ingrowth characteristics.

6.3.2 Evaluation

μCT is used to measure critical architectural dimensions of combinatorial test scaffold. Dimensions W, H, and Ø, from Fig. 6.4 are measured for each region in Designs 1 – 3. Three two-dimensional slices through the scaffold for each region are analyzed in both the X-Z and Y-Z plane. At least 25 measurements in each dimensions are recorded per slice. Fig. 6.6 shows the measurements performed on a representative two-dimensional slice taken at the middle of region m_1M_2 in Design 2. Additionally, the interconnection sizes of each channel at the region interfaces are measured. Design 4 was fabricated for a study that is independent from Designs 1 – 3 and macrostructural architecture has not be investigated quantitatively and will be assessed qualitatively from optical data.

6.3.3 Results

Fabricated and sintered scaffolds with the region designs given in Fig. 6.5 are shown in Fig. 6.7. Higher magnification images of the interface between regions demonstrate the differences in the macroporosity, region communication, and filament size. Qualitatively, the structures are uniform in each region, with repeatable filament di-

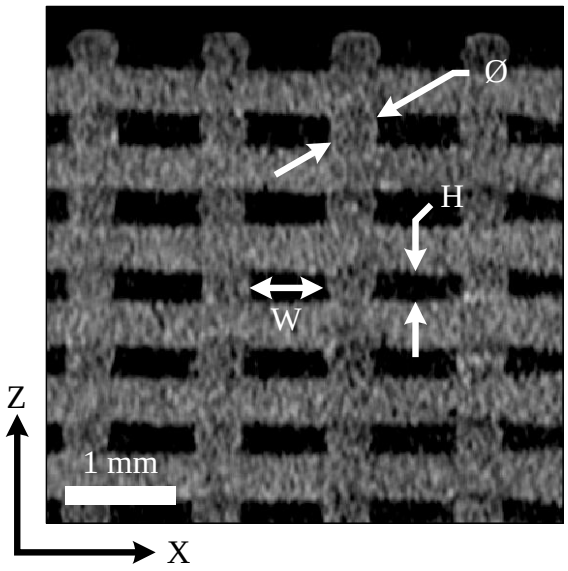


Figure 6.6: Representative μCT image displaying the macropore measurements taken with each design. This particular image is taken from Design 2, region m_1M_2. Axes orientation is shown on the figure, where the Y axis is orthogonal to the X-Z plane.

ameters and spacings between filaments, and the different materials interface with intimate contact. Additionally, there is no evidence from the optical, μCT, and SEM imaging modalities of sintering-induced cracking at the interface between the different material compositions.

μCT images of each design show the internal macrostructure at the region interfaces, Fig. 6.7 Column 3. The direct pathways for fluid transport are clearly seen in Designs 2 and 3, whereas channels are absent in Design 1. Minimum channel interconnection sizes are given for each region interface in Designs 1 – 3, Tb. 6.1. Logically the minimal interface interconnections are at the m_1M_3/m_2M_1 interface in Design 3; this interface is between two tightly spaced macroporosities (W = 150 μm) where there is a filament diameter mismatch. Macrostructural data from the uniform regions in each region is provided in Tb. 6.2. There is a sharp distinction between the different macroporosity designs, Fig. 6.8. The measured data corresponds well with the nominal designed interconnection size in the W dimension, however there is less correspondence in the H dimension.

Table 6.1: Minimum Interconnection Size at Region Interfaces. Mean ± Standard Deviation [μm]

Interface	Design 1	Design 2	Design 3
m_1M_1/m_1M_2	none	330 ± 60	-
m_1M_1/m_2M_1	none	270 ± 40	-
m_1M_2/m_2M_2	none	260 ± 50	-
m_2M_1/m_2M_2	none	290 ± 50	310 ± 60
m_1M_3/m_1M_4	-	-	180 ± 30
m_1M_3/m_2M_1	-	-	120 ± 50
m_1M_4/m_2M_2	-	-	280 ± 60

Table 6.2: Interconnection Measurements for Macroporosities M_1, M_2, M_3, and M_4. Mean ± Standard Deviation [μm]

Dim.	M_1	M_2	M_3	M_4	Nom.
W	160 ± 30	730 ± 30	140 ± 20	700 ± 30	$\{150, 750\}$
H	280 ± 30	250 ± 50	170 ± 40	130 ± 30	$\{230, 150\}$
Ø	400 ± 30	410 ± 20	280 ± 20	310 ± 20	$\{460, 300\}$

The two microstructures are shown in the SEM image in Fig. 6.9. This image, taken at the geometric center of the top layer in a representative scaffold shows the distinct microstructural differences between the two material compositions. Whereas composition m_1 is almost fully dense, save for nanometer scale incomplete sintering porosity, m_2 has a considerable volume of interconnected porosity. From previous work, the actual pore volume percent is <2% in m_1 [31] and 46% in m_2 [58].

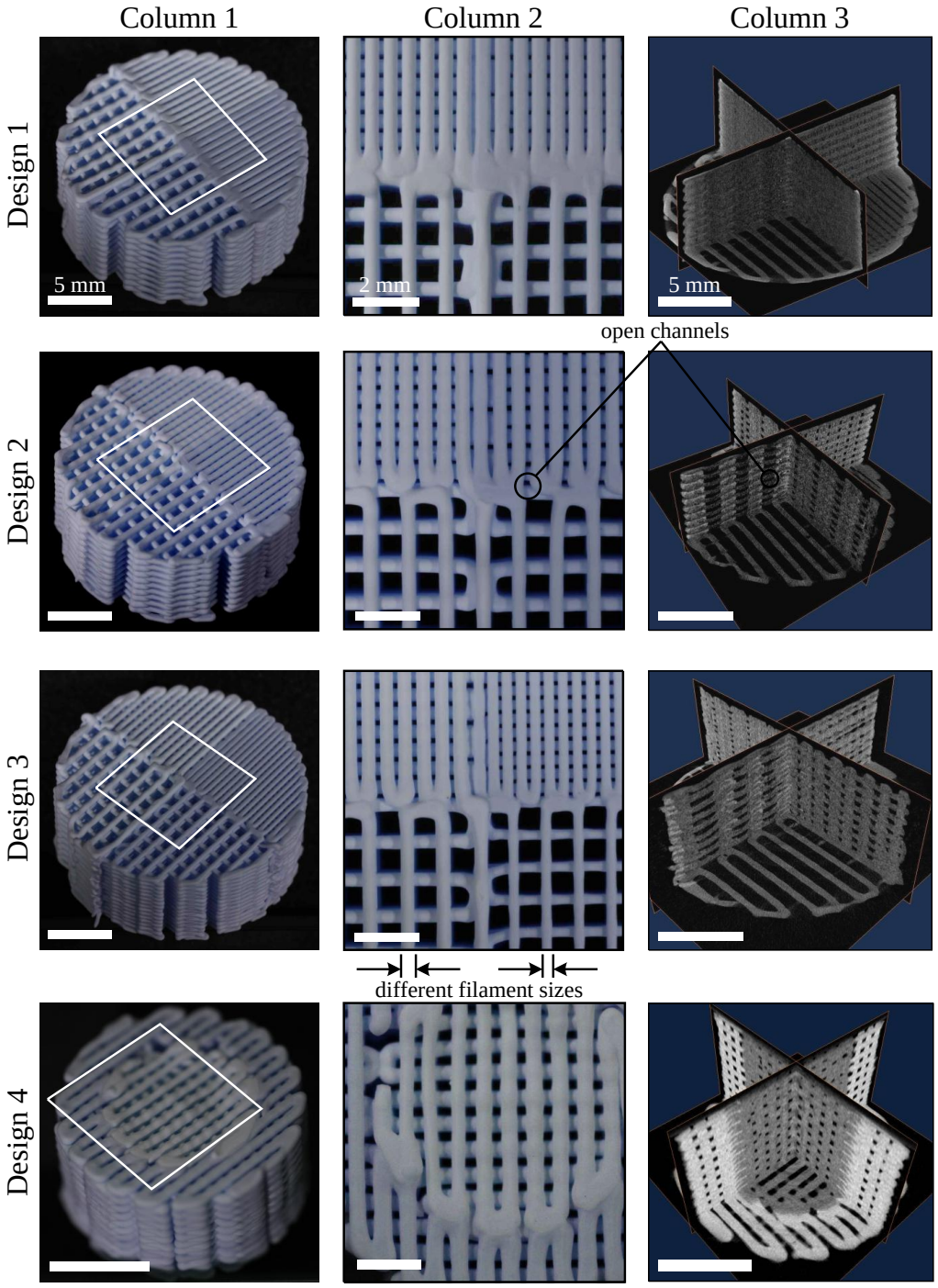


Figure 6.7: Manufacturing results for Designs 1 – 4. Columns 1 and 2 are optical images of the designs. Column 3 displays orthogonal slices through the set of μCT data showing the internal macropore design and region communication. Design 1 has a completely walled off region interface preventing fluid transport. Designs 2 and 3 have open channels between regions. Design 3 incorporates multiple filament sizes within a single scaffold. Design 4 is a concentric design.

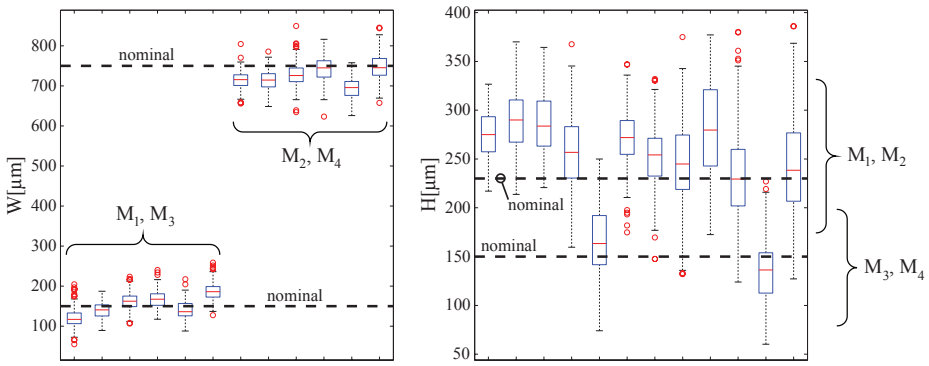


Figure 6.8: Boxplot representation of the set of measured interconnection sizes for each region of Designs 1 – 3. Box horizontal lines represent the 25%, 50%, and 75% quartiles of data. Circles represent outliers. The nominal designed interconnection sizes are given for comparison. Specific labels for each region are omitted for clarity. Data displays a distinct difference between the macroporosity design levels.

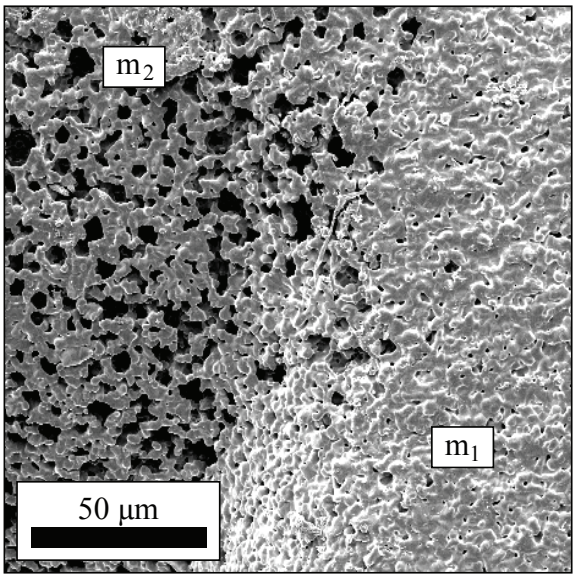


Figure 6.9: SEM data displaying the microstructural differences at the interface between distinct material compositions.

6.4 Near-Net Shape

6.4.1 Scaffold Designs

Near-net shape scaffold designs, Designs 5 – 7, are displayed in Fig. 6.10. Designs 5 – 7 all have macroporosity M_5, as this is the common macroporosity used the *in vivo* studies in [8, 9, 13] and demonstrates the potential to extend monolithic scaffold studies to more advanced architecture studies. Designs 5 and 6 have a spherical envelope but have different internal structures. Design 5 integrates two different build materials; a shell of material m_1 completely encases a core of material m_2. We chose this design to demonstrate the feasibility of scaffolds with internal interfaces between dissimilar materials, possibly to replicate the interface between cortical and trabecular bone. Design 6 has a hollow ellipsoidal cavity, design to demonstrate that features similar to a marrow cavity are feasible. Both of the shells for Designs 5 – 6 have an alternating pattern of odd layers with circular patterns and even layers with a radial pattern to demonstrate an alternate build pattern from the grid type pattern utilized in [8, 9, 13] and Designs 1 – 4. In fact, Design 6 integrates a circular/radial region with a grid region, demonstrating the ability to facilely switch between build patterns and possibly could be used to align filaments with an anticipated principle stress. Design 7 is a torus design with downward facing concave feature open to the X-Y plane; this design demonstrates that small concavities of this type are feasible, even though the mold will constrict the scaffold as it shrinks while drying. Design 7 has the grid type build pattern used in Designs 1 – 4. Designs 5 – 7 are evaluated qualitatively using optical imaging and μCT.

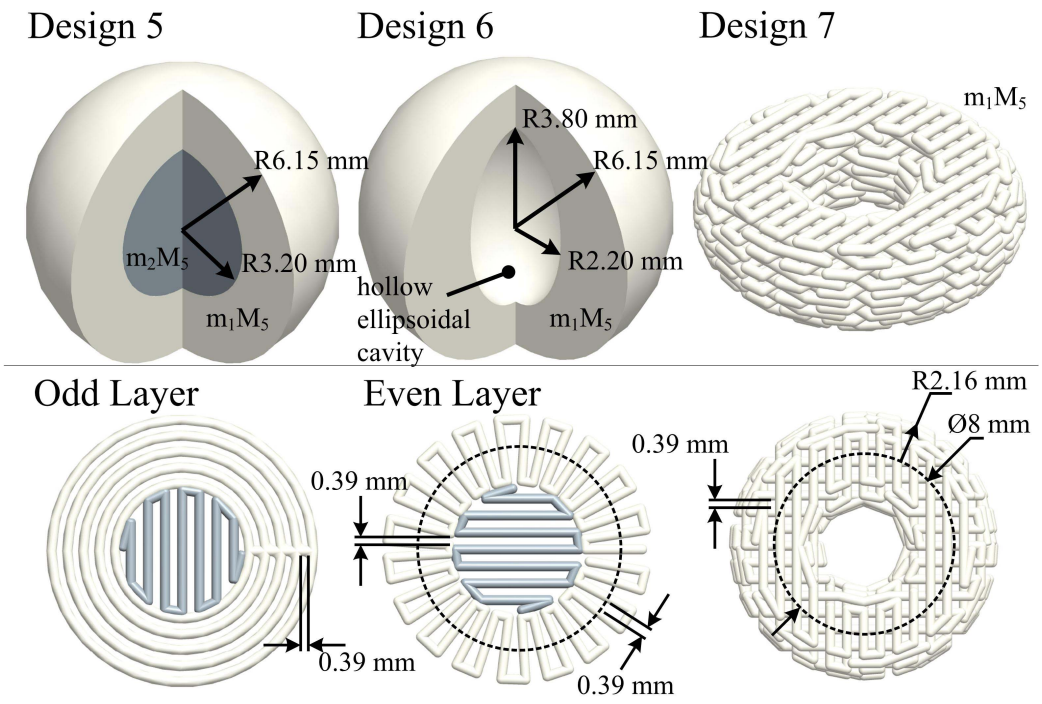


Figure 6.10: Designs 5 – 7. Designs 5 and 6 have a spherical envelope where Design 5 integrates two different microporosities and Design 6 has a hollow ellipsoidal cavity. The shells of Designs 5 and 6 use an alternating macropore pattern of circular and radial patterns for each layer; this circular/radial pattern can easily be interfaced with a grid pattern. Design 7 is a torus with a grid macropore pattern.

6.4.2 Results

Fabrication results are displayed in Fig. 6.11. Similar to Section 6.3, there is no evidence of sintering or shrinking induced cracking, either from the interface between two different materials, Design 5, or from constrained shrinking around the mold supporting the concavity in Design 7. On the surface, Designs 5 and 6 look identical, Fig. 6.11, however the difference in design is clear from the reconstructed μCT data in Fig. 6.11 Column 3. The ellipsoidal cavity in Design 6 maintains its shape despite being unsupported. In Design 6, the ellipsoidal cavity has a steep slope throughout most of the layer-by-layer build routine and therefore is self-supporting. At the top of the ellipsoid, the slope rapidly becomes more shallow, however at this location the distance to be spanned by a filament is short and an ellipsoidal cavity

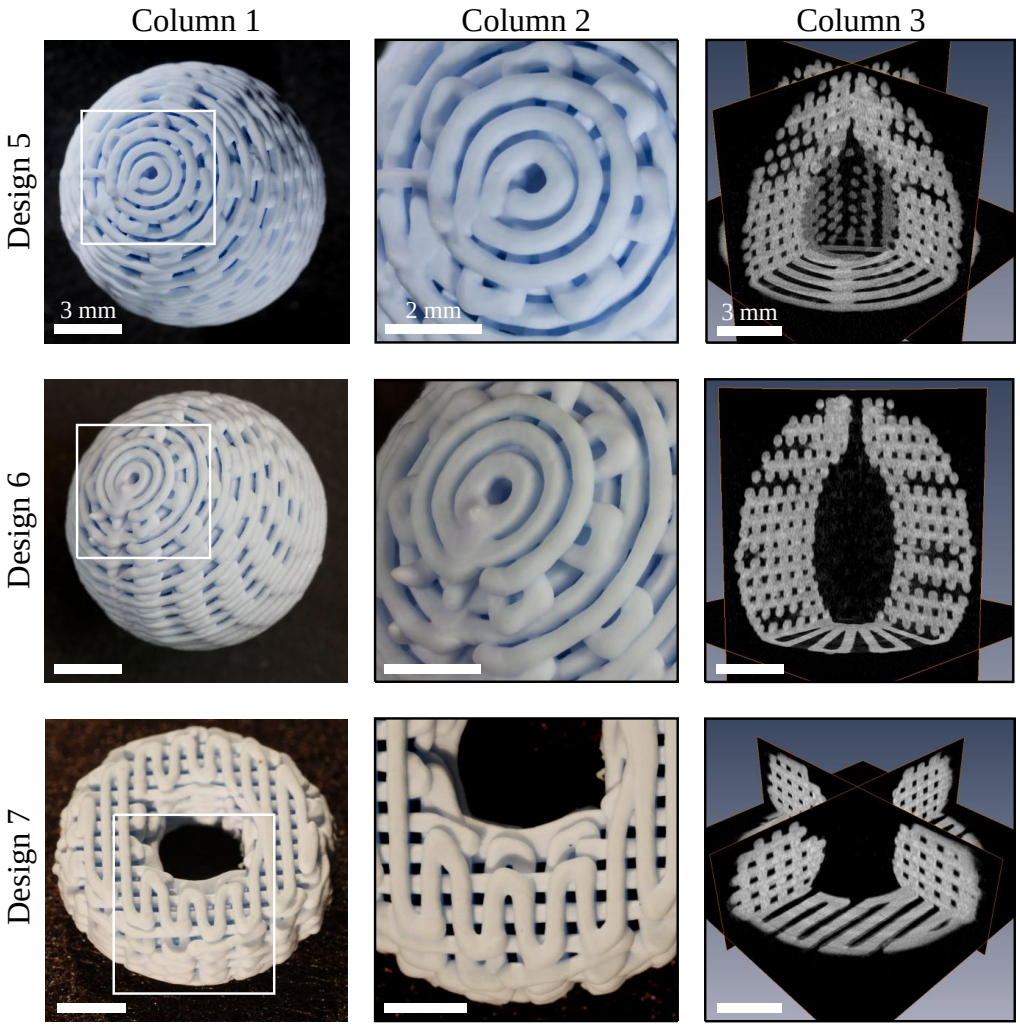


Figure 6.11: Manufacturing results for Designs 5 – 7. Columns 1 and 2 are optical images of the designs. Column 3 displays orthogonal slices through the set of μCT data. On the surface, Designs 5 and 6 look identical, however the difference in design is clear from the reconstructed μCT data in Column 3. Design 5 integrates two materials with different microporosities; differences in the microporosity are identifiable by light and dark regions, denoting materials m_1 and m_2, respectively. The ellipsoidal cavity in Design 6 maintains its shape despite being unsupported. Design 7 demonstrates surface and internal features for a torus where the convex and concave contours of the scaffold envelope are supported by a mold.

can be completed with little deviation from the desired design. Logically, not all hollow cavities will be self-supporting, but some cavity designs will be, demonstrating promise for recreating anatomically derived marrow cavities.

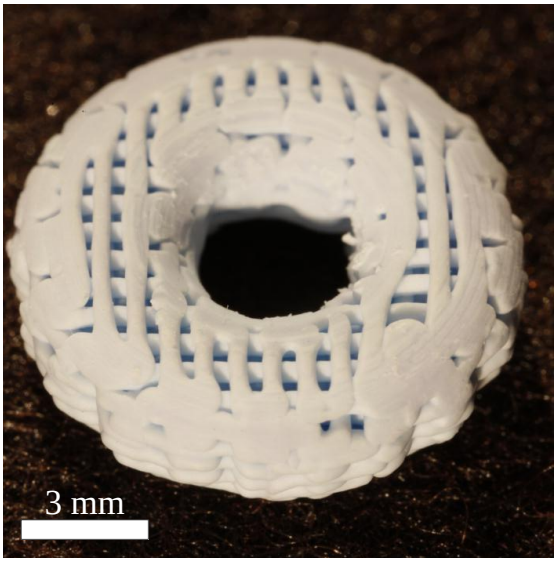


Figure 6.12: Surface of Design 7 that directly abuts the mold to support large unsupported features. The mold surface does modify the surface characteristics and constrict macropore interconnection size at the scaffold periphery.

The surface of the mold does affect the scaffold surface characteristics, Fig. 6.12. Regions abutting the mold are noticeably flattened with more constricted macropore openings to the periphery of the scaffold, as compared to Fig. 6.11. By a qualitative assessment, the macroporosity open to the periphery is considerably larger than the stated 100 μm interconnection size necessary for bone cell and vasculature infiltration [24] and the slightly constricted pores are not expected to diminish *in vivo* efficacy.

6.5 Discussion

The ability to build near-net shape scaffold envelopes with precisely located regions of different compositions enables unique capabilities on four fronts. The following subsections will discuss new capabilities and the implications.

6.5.1 Efficient In Vivo Evaluation

The ability to integrate multiple materials with different micro- and macrostructures within a single scaffold permits efficient evaluation of porosity designs *in vivo*. Structures such as the combinatorial test sample may represent a new paradigm in *in vivo* testing. It is advantageous to reduce the number of animals used in a study and have the designs colocated to yield more accurate comparisons. With proper manufacturing and evaluation tools, it is unnecessary to test each design with its own individualized test.

Although we focused on HA scaffolds here, this work could be identically extended to test multiple base materials as well. For instance, a β-TCP region directly apposed to a HA region with an identical macrostructure could be used to test the relative *in vivo* response (in terms of degradation rates, growth distribution, and growth volume) of each. The number of compositions in a single scaffold is limited by the number of extrusion systems in the array and the limit at which a designed local environment volume is too small to yield statistically significant data. The μRD system used here accommodates four compositions; adding more requires multiplexing a known toolset.

The advanced manufacturing capabilities demonstrated here are predicated on the ability to uniquely define design regions. The macrostructure data displays that μRD utilizing the BTILC control method creates the desired interconnection sizes. In the W dimension, the middle 50% of the interconnections, box region of the boxplots in Fig. 6.8, are fabricated within 75 μm of the nominal design point in all quadrants. In the H dimension, the middle quartiles of data are within 100 μm of the design. Given the well known wide range of permissable interconnection sizes we believe that this level of manufacturing accuracy is sufficient for direct comparisons between macroporosities.

6.5.2 Evaluation of In Vivo Response to Heterogenous Interfaces

Human tissue is both heterogenous and anisotropic. An ultimate goal for many tissue engineering researchers is to best reproduce this complex anatomical environment by artificial means. To reproduce this environment, say an attempt to replicate the transition between cortical bone to trabecular bone, we must first test how the body responds to these artificial representations with *in vivo* studies. An interesting question is how does the body react to sharp gradients in mechanical stiffness when the gradient is not near a native cortical / trabecular bone interface. Despite impressive advancements in scaffold manufacturing technologies, artificial scaffolds will always be an approximation of the natural environment. Current research approximates the natural environment with discretized regions of anatomically derived features [4, 71]. What is needed are tools such as the combinatorial test sample to systematically analyze the *in vivo* response to material property gradients.

This work displays that the interface conditions can be tailored. The interface can be completely blocked to preclude fluid and cell transport or be sufficiently open. The minimal interconnection size for all quadrant interfaces, Tb. 6.1, is larger than the stated minimum allowable interconnection, 100 μm [24], and larger than the diameter of blood vessels [72], ensuring that local fluid transport across the interface will not be impeded.

6.5.3 Directed Bone Growth

Integration of functional materials has long been a part of manufacturing processes in the electronics industry. For instance, the integration of electrode and dielectric piezoactive materials for piezoelectric actuators. Given the ability to integrate

multiple materials in bone scaffolds, analogous functionalities can be included to optimize bone scaffold designs. One functionalization could be the directed placement of high-microporosity, bone growth inducing regions of material with integrated, low-microporosity load-bearing struts. Structures such as this could be optimized to bear physiological loads while facilitating paths for bone to grow towards central regions of the scaffold. We believe functionalities such as this would aid in the initial healing response of implanted scaffolds, quickly providing scaffold strength and resiliency to post-surgery injury.

Future work will investigate the ability of microporous materials to drive new bone growth into inner scaffold regions with designs similar to Design 4. We will be able to assess whether inner regions that typically do not grow bone at early time points will have hastened development of natural bone by appropriate material selection. A near-net shape version of Design 4 is demonstrated with Design 5.

6.5.4 Contoured Scaffold Envelopes

Our method of building scaffolds within a support mold to enable the fabrication of large unsupported features expands the set of conceivable scaffold features. Currently, our investigation of the feasible set of scaffold features is not exhaustive. We leave a complete exploration for future studies. Fig. 6.13 demonstrates some of the feasible features, some that need further exploration, and infeasible features. Equally feasible are some combinations of feasible features, such as a contoured structure that is concave and open to the X-Z or Y-Z plane and convex and open to the X-Y plane.

The feasibility of the set of downward facing concave features is currently undetermined, Fig. 6.13. Some of these concave features are self-supporting and do not need a mold, as evidenced by the ellipsoidal cavity in Design 6. Concave features with shallow slopes may easily release from a mold because the normal forces generated by

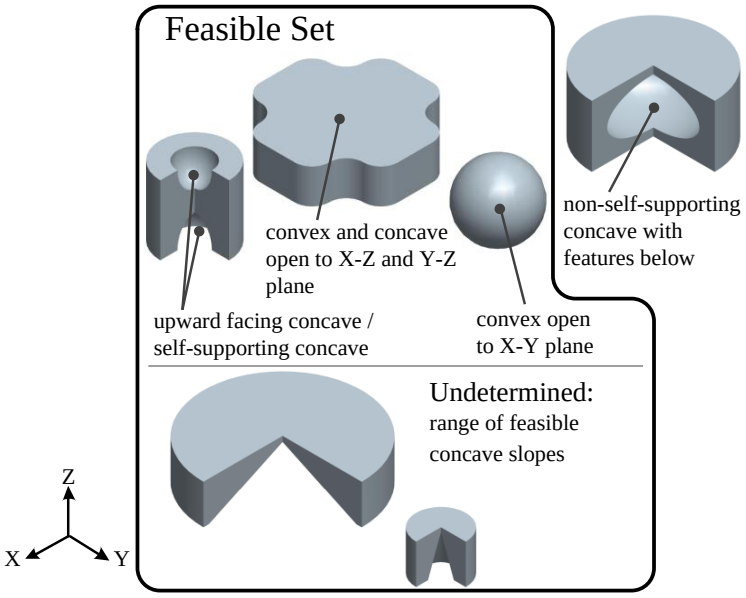


Figure 6.13: Feasible and infeasible design features using the methods given here. Some features require further investigation and are left for future work. Axes provide the axes orientation for the directions referenced in the figure.

constrained shrinking around a shallow mold will be low. Future work will investigate which concave slopes are self-supporting and which are self-releasing without cracking the material in its 'green' state. Currently, what has been shown is that some of the basic features that are expected to be used in advanced architecture scaffolds can be fabricated by the methods given here. Even designs that are incapable of being fabricated by traditional manufacturing methods, such as designs with hollow features, can be fabricated.

Chapter 7

Bumpless Transfer

7.1 Introduction

Iterative Learning Control (ILC) is an effective methodology for precision control for systems that track repeated trajectories [10]. Input signals generated by ILC are applied in open-loop or as a supplement to feedback control, typically achieving performances beyond what is capable by feedback alone. The ILC algorithm relies on trajectory repetition to learn an approximate inverse signal of the dynamics, disturbances, and unmodeled dynamics [73]. Therein lies a primary limitation of ILC. If the trajectory changes, the ILC algorithm must be reinitiated.

The targeted applications for this chapter are manufacturing systems or material interrogation systems that require both high precision and process flexibility. Example applications include μRD, electrohydrodynamic jet printing [74], and atomic force microscopy [75]. These systems are versatile in that they require no tooled setups or dies, having little impedance from concept to product so that designs can be flexibly interchanged. However, the reliance of ILC on process repetition is not well aligned with these applications, inhibiting efficient implementation. The objective of this research is to explore adaptations of ILC that are as flexible as these applications.

Previous attempts at adding trajectory flexibility have de-emphasized the time specificity of ILC, instead focusing on learning a set of dynamics, and applying learned signals to a similar set of dynamics [11, 12, 43, 46, 76]. These methods and more are

given in Chapter 2. The BTILC method detailed in Chapters 3 and 5 presents a very different adaptation of ILC from those listed above, maintaining the time specificity of ILC. BTILC focuses on learning a set of tasks, termed *basis tasks*, through a training routine, and investigating effective methods to apply this *basis task* information to construct complex trajectories. BTILC orchestrates a tradeoff between trajectory flexibility and some performance degradation. For the μRD application, the loss in performance is not pronounced because the material extrusion dynamics are slow and *set equivalence* is easily satisfied because the space of *basis tasks* is small. For the target flexible manufacturing systems introduced in Chapter 1, the positioning systems are characterized by fast dynamics and are therefore more susceptible to errors as a result of large state deviations at the transitions, δx_n from the discussion in Section 3.4.2. There are an infinite number of motion primitives that could be selected as *basis tasks*, making *set equivalence* difficult to satisfy. Ideally, we wish to improve trajectory tracking over feedback control alone with BTILC, without sacrificing the innate flexibility of these systems.

This work builds on the BTILC framework. Importantly, it investigates a relaxed trajectory constraint where the *basis tasks* sequences in the *training* and *operation sets* do not satisfy *set equivalence*, Definition 8 from Chapter 3. This constraint relaxation allows for a larger set of operation trajectories to be accomplished with the same training information, although tracking performance is lost. A novel bumpless transfer scheme for open-loop signals is presented that regains some lost performance.

The chapter builds on the BTILC definitions, implementation, and performance information is given in Chapter 3. Section 7.2 gives a brief performance example using the positioning system system detailed in Chapter 4. Section 7.3 introduces the bumpless transfer scheme. The experimental setup and results are given in Sections 7.4 and 7.5, respectively. Throughout the results, a three-way comparison will be made between the typical implementation of ILC, BTILC, and BTILC with bumpless

transfer. Section 7.6 provides a brief discussion of the important results.

7.2 Positioning Performance

To remind the reader, the superscript T denotes the *training set* and O denotes the *operation set*. This chapter frequently compares BTILC and bumpless transfer with BTILC to the conventional ILC algorithm; conventional ILC will be denoted by the superscript $L(j)$ where j is the iteration index.

Performance degradation at the transition indices is discussed in greater detail in Chapter 3. The salient point from this discussion is that performance degradation is minimized when there is *set equivalence*, as given in *Definition 8*. However, this condition may be restrictive in some cases precisely because it requires N^2 task transitions in the *training set* to identify a transition between every *basis task* in the *operation space*. For the *operation space* given in this chapter, that would require 49^2 unique identifications. To weaken this condition, we can neglect set equivalence and only impose a trajectory smoothness condition:

$$\begin{aligned} r_i^O(K_i - 1) &= r_{i+1}^O(0) \\ (q^{-1} - 1) r_i^O(K_i - 1) &= (1 - q) r_{i+1}^O(0) \end{aligned} \tag{7.1}$$

where $i = \{0, 1, \ldots\}$ are consecutive *basis tasks* in the *operation set*, $\Delta_i + K_i = \Delta_{i+1}$, and q is the forward shift operator, $qx(k) = x(k+1)$. That is, the positions at either side of the *basis task* transition must be coincident and the velocities must be the same. Consider the positioning system in Fig. 4.1 tracking an arbitrary trajectory that satisfies the weakened condition (7.1). The lack of *set equivalence* leads to performance degradation, Fig. 7.1, where the performance degrades immediately

after the transition indices, approaching feedback control performance, $e_d^{L(0)}(k)$. The input signal that is generated by BTILC is not smooth in this case. This is because the identified *basis signals* were identified for different task initial conditions, because there was not *set equivalence*. This large bump in input signal magnitude reduces performance near transition time indices.

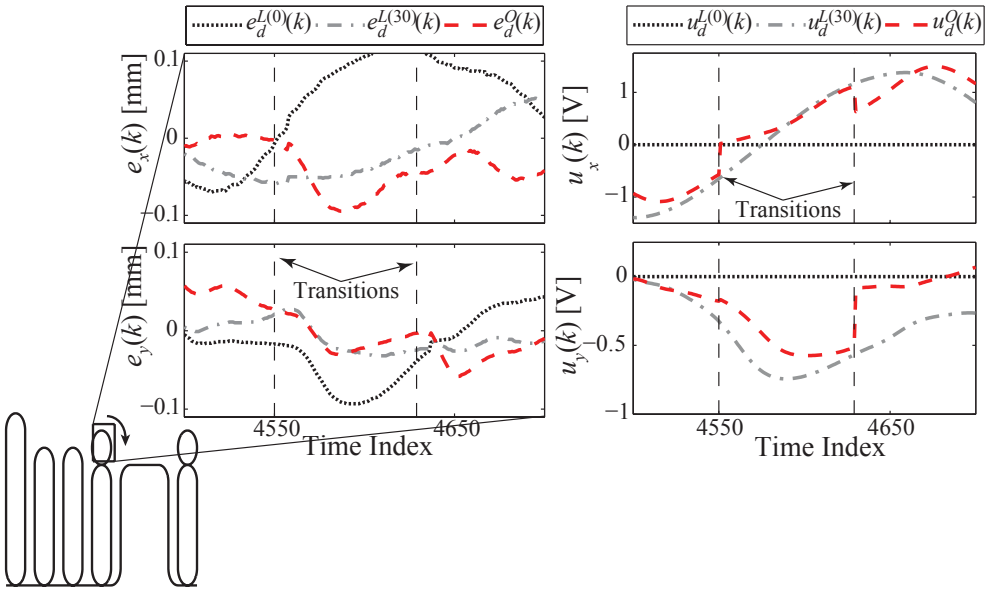


Figure 7.1: Axes error signals and corresponding open-loop input signals during a task transition without set equivalence. At the basis signal, $u_d^O(k)$, transition points there is a sharp change in input signal magnitude yielding a transient degradation in operation set performance, $e_d^O(k)$, approaching the performance of feedback control, $e_d^{L(0)}(k)$.

7.3 Bumpless Transfer

ILC produces a supplemental signal that is the inverse of the system dynamics, repeated disturbances, and uncertainties [73]. Smoothness constraints such as (7.1) do not account for higher order derivatives in the reference signal. Given that the ILC input signal achieves an inverse of the dynamics and other factors and that *basis task* sequence affects transition states, x_n^O, there will be *basis signals*, $u_n^O(k)$, that are not

appropriate for the instantaneous state conditions at the transitions. Consequently, performance will degrade. The question is how do you compensate for state differences between the *training* and *operation sets*, δx_n, when,in general, you have no knowledge of the states? Instead of translating a state difference to a control action, perhaps performance can be improved by smoothing the BTILC transitions in Fig. 7.1.

Here, we borrow the concept of bumpless transfer from the feedback control community [77]. Systems with a switched control scheme will have transient performance degradation after a controller switch because the two controllers have different objectives and therefore different input magnitudes at the transition time. The idea in bumpless transfer is that transition performance can be improved by forcing the latent control signal to emulate the active signal at the instant of transition. After a transition, control action converges to that of the new controller at a rate given by the bumpless transfer weighting filters.

We have modified the bumpless transfer algorithm for use on open-loop signals, Fig. 7.2. Prior to a *basis task* transition, each *basis signal* in the *operation set* is applied to a cascade of filters where the latent signal, $u_l(k)$, tracks the filtered input signal of the i^{th} *basis signal*. At the transition, switches s_2 and s_3 flip and the $i+1^{th}$, bumpless filtered, *basis signal* is applied to the plant. The tunable parameter is the digital filter $b(q)/a(q)$ where the polynomials $a(q)$ and $b(q)$ are designed to reject a constant $u_{i+1}(0)$ and pass the signal $u(k)$ before the transition index Δ_{i+1}.

$u_l(k) =$

$$\begin{cases} \overbrace{\frac{a(q)}{a(q)+b(q)}}^{\text{reject}} u_{i+1}(0) + \overbrace{\frac{b(q)}{a(q)+b(q)}}^{\text{pass}} u(k) & \text{for } k < \Delta_{i+1} \\ \frac{a(q)+b(q)}{a(q)+b(q)} u_\Delta(k) & \text{for } k \geq \Delta_{i+1} \end{cases} \tag{7.2}$$

where

$$u_{\Delta}(k) = \{u(\Delta_{i+1} - \delta), u(\Delta_{i+1} - \delta + 1), \\ \ldots, u(\Delta_{i+1} - 1), u_{i+1}(0), \ldots\} \tag{7.3}$$

and δ is the order of the polynomial $a(q) + b(q)$.

Stability is guaranteed when the roots of polynomial $a(q) + b(q)$ are contained within the unit disk; $\max_q\{|q| < 1 : a(q) + b(q) = 0\}$. Critically, this filter does not modify $u_{i+1}(k)$ at time indices away from Δ_{i+1}. That is

$$u(k) \to u_{i+1}(k) \text{ as } k \to (\Delta_{i+1} + K_{i+1})\,. \tag{7.4}$$

Fig. 7.3 demonstrates the bumpless filter performance with arbitrary signals from the experimental section, Section 7.5. Without bumpless transfer, the signal sent to the physical plant would be the trace that starts from $u^O(k)$ and abruptly transitions to $u_{i+1}(k)$ at the time index $k \approx 150$. Notice that the bumpless transfer filter smoothes the transition between *basis signals* but does not attenuate the active signal as time proceeds. The all-pass nature of this designed bumpless transfer scheme is important because each *basis signal* contains content that improves positioning performance. Excessive filtering, such as using a low-pass filter to attenuate transients at the transitions, would remove this important signal content.

7.4 Experimental Setup

BTILC with and without bumpless transfer is tested experimentally on a serial positioning system. The positioning system is described in detail in Chapter 4, providing frequency responses of the models for the open-loop plant, Fig. 4.7, and complemen-

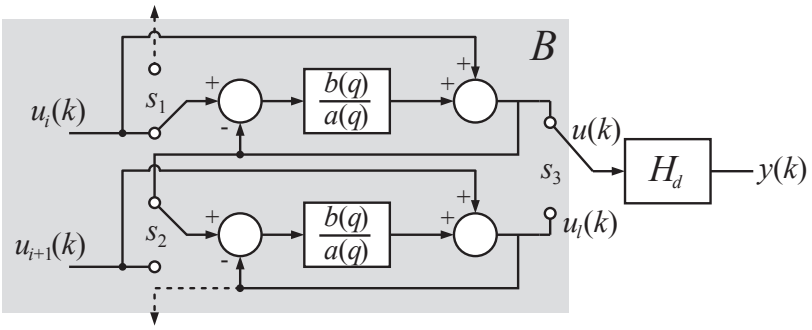


Figure 7.2: Section of a block diagram of the bumpless transfer filter $B\left(u^O(k)\right)$ in a cascade of subfilters.

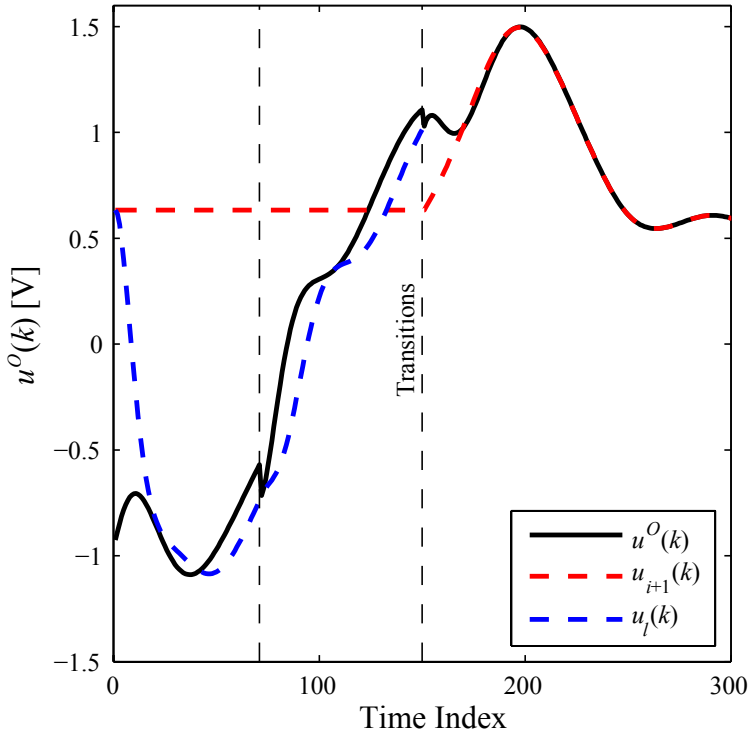


Figure 7.3: Demonstration of the bumpless transfer filter in Fig. 7.2. The filter first has a zero initial state. The latent signal, $u_l(k)$, then tracks the current input signal, $u^O(k)$, with time. At the moment of transition, the filter switches which signal it is tracking and quickly tracks the new active signal, $u_{i+1}(k)$. As time progresses, the filter tracks the new active signal with no signal attenuation.

tary sensitivity function, Fig. 4.8. More important to ILC and BTILC application is the frequency response for the transfer function from the ILC input to the output, $\frac{y(z)}{u^L(z)} = \frac{H_d(z)}{1+k_d(z)H_d(z)}$, displayed in Fig. 7.4.

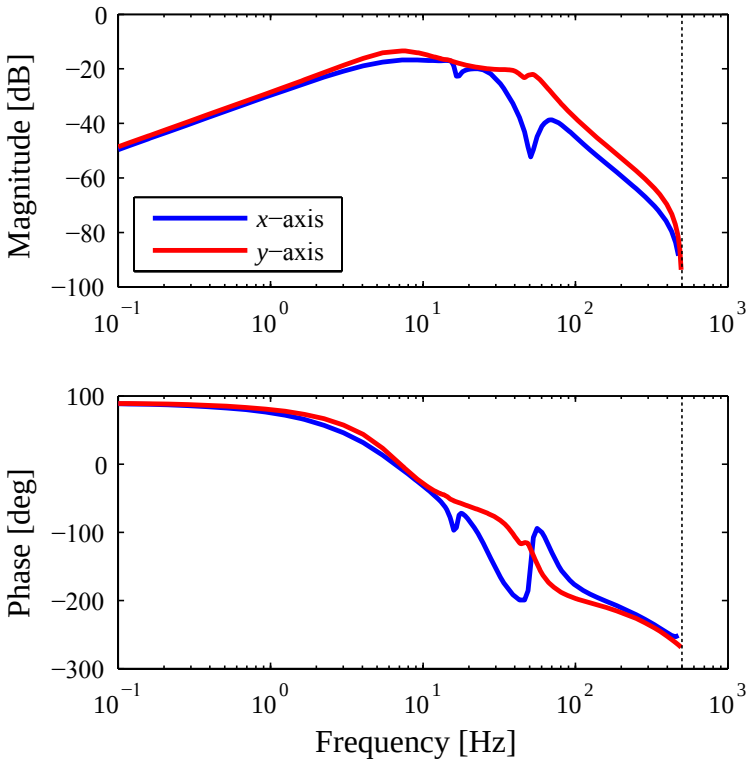


Figure 7.4: Frequency responses of the transfer function $\frac{y(z)}{u^L(z)} = \frac{H_d(z)}{1+k_d(z)H_d(z)}$ for the x and y-axes.

7.4.1 Training Set

The *training set* is designed to include linear motion primitives and circular motion (clockwise (CW) and counterclockwise (CCW)) primitives. Fig. 7.5 shows the designed training routine. The *training set* trajectory consists of six circuits around a perimeter in the same plane, $r_z(k) = 0$, with each circuit designed to identify different task information. Descriptively, there are six *basis tasks* types in the *operation space*:

$$\begin{aligned} R^O = &\{linear, linear\ start, linear\ stop, \\ &CCW, CW, dwell\} \end{aligned} \tag{7.5}$$

where each type has classifiers, giving 49 unique *basis tasks*. Classifiers are given in

Table 7.1. The adjacency matrix for the *training set* is given in (7.6).

$$\mathbf{T} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_{8\times 8} & \mathbf{0} & \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{I}_{4\times 4}\mathbf{I}_{4\times 4} \end{bmatrix} & \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{I}^{\perp}_{4\times 4}\mathbf{I}^{\perp}_{4\times 4} \end{bmatrix} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{8\times 1} \\ \mathbf{I}_{8\times 8} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \begin{bmatrix} \mathbf{0} & \mathbf{I}^{\perp}_{4\times 4} \\ \mathbf{0} & \mathbf{I}^{\perp}_{4\times 4} \end{bmatrix} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \begin{bmatrix} \mathbf{0} & \mathbf{I}_{4\times 4} \\ \mathbf{0} & \mathbf{I}_{4\times 4} \end{bmatrix} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{1\times 8} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{7.6}$$

where $\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{\perp} = \begin{bmatrix} b & a \\ d & c \end{bmatrix}$.

Table 7.1: Basis Tasks Classifiers in Training Set

Circuit	Type	Vel $[mm/s]$	Rad $[mm]$	Dir/Quad
1	Linear	10	0	+x,+y,-x,-y
2	Linear	20	0	+x,+y,-x,-y
3	CCW	20	1	3,4,1,2
4	CCW	20	4	3,4,1,2
5	CW	20	1	3,4,1,2
6	CW	20	4	3,4,1,2

ILC is applied to the positioning system given in Chapter 4 with the reference trajectory in Fig. 7.5 and with a standard update law, (2.2). $Q(q)$ is chosen to be a Gaussian filter and $L(q)$ a standard Proportional-Derivative (P-D) type learning law, however, other filter choices are feasible. Filter details are provided in (7.7), (7.8), and in Tb. 7.2. ILC is run for 30 iterations. The identified *basis signals* are considered to be the best input to achieve their respective *basis tasks*. Each *basis signal* is stored

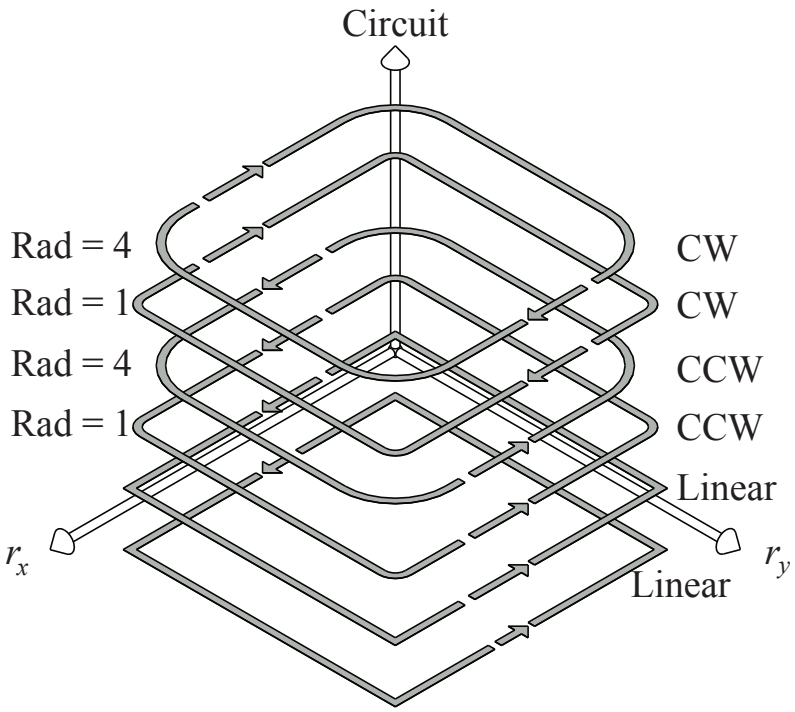


Figure 7.5: *Training set*. All six circuits are in the same plane with each circuit designed to identify different *basis task* information. *Training set* contains the *basis task* types {*linear*, *linear start*, *linear stop*, *CCW*, *CW*, and *dwell*}.

in the *basis signal library* for application in the *operation set*, as specified in Chapter 3.

$$L_d\left(e_d(k)\right) = k_{P,d}e_d(k) + k_{D,d}\left(e_d(k) - e_d(k-1)\right) \tag{7.7}$$

$$Q_d\left(x(k), k^*\right) = \frac{1}{\sum_{i=0}^{L} e^{-\frac{(t_s(k^*-i))^2}{2\sigma^2}}} \sum_{i=0}^{L} x(k^*-i)e^{-\frac{(t_s(k^*-i))^2}{2\sigma^2}} \tag{7.8}$$

Table 7.2: P-D Type ILC Filter Gains

d	k_P	k_D	σ
x-axis	1.2	25	1.02×10^{-2}
y-axis	1	10	7.36×10^{-3}

7.4.2 Operation Set

Four different *operation set* trajectories are tested. The two in the top row of Fig. 7.6, have an *operation set* that is equivalent to the *training set*: $\mathbf{T} + \mathbf{O} = \mathbf{T}$. The two in the bottom row of Fig. 7.6, violate *set equivalence*, satisfying only the weakened constraint in (7.1). Namely, these trajectories contain direct $CCW \leftrightarrow CW$, $CCW \leftrightarrow CCW$, and $CW \leftrightarrow CW$ *basis task* transitions. Therefore, $\mathbf{O}(25:32, 25:32) \neq \mathbf{0}_{16\times16}$ and $\mathbf{T} + \mathbf{O} \neq \mathbf{T}$. Three of the trajectories $\{(1,1),(1,2),(2,1)\}$ are motivated by the raster patterns used in micro-scale fabrication processes [74] or material interrogation processes [75]. $(2,2)$ is arbitrarily chosen to explore the flexibility limits of BTILC. A three-way comparison is made for each *operation set* trajectory: (a) trajectory tracking for the conventional ILC algorithm run to 30 iterations, $y^{L(30)}(k)$, (b) BTILC, $y^{O}(k)$, (c) and BTILC with bumpless transfer, $B\left(y^{O}(k)\right)$. Note, that in some instances a comparison is made to just feedback, $e^{L(0)}(k)$, to assess performance gains over standard control.

The bumpless transfer filter is chosen heuristically

$$\frac{b(z)}{a(z)} = \frac{0.0117z + 0.0195}{z^2 - 1.819z + 0.8187} \tag{7.9}$$

to reject the influence of the $i + 1^{th}$ *basis signal* and pass the signal from the i^{th} *basis signal* before a task transition, see (7.2). The frequency response of each of the transfer functions given in (7.2) for the design in (7.9) is shown in Fig. 7.7. The pole-zero map for each transfer function shows that the digital filter is stable and minimum phase, Fig. 7.8.

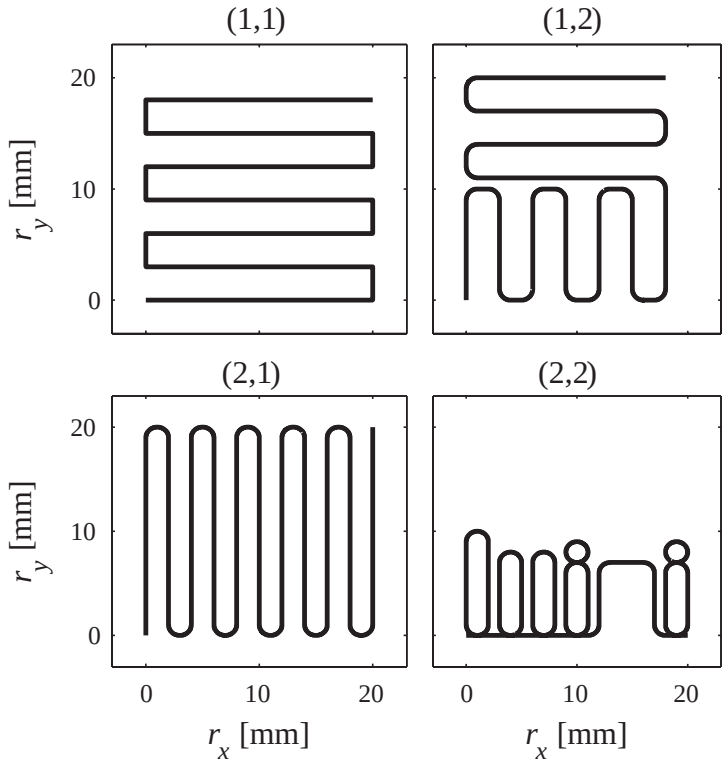


Figure 7.6: *Operation set* trajectories to test in simulation. *Operation sets* in the top row satisfy *set equivalence*, $\mathbf{T} + \mathbf{O} = \mathbf{T}$. *Operation sets* in the bottom row do not satisfy *set equivalence*, $\mathbf{T} + \mathbf{O} \neq \mathbf{T}$.

7.5 Experimental Results

The y-axis is capable of more accurately tracking reference trajectories than the x-axis because the x-axis has relatively more inertia. Consequently, error signals in the y-axis are smaller in magnitude and have higher signal-to-noise ratios. ILC is most effective in decreasing error in the x-axis and therefore this section will focus on x-axis results. y-axis results follow the same trends, however to a lesser degree. All BTILC data is an average of five trials with the same input, $u_d^O(k)$ or $B\left(u_d^O(k)\right)$.

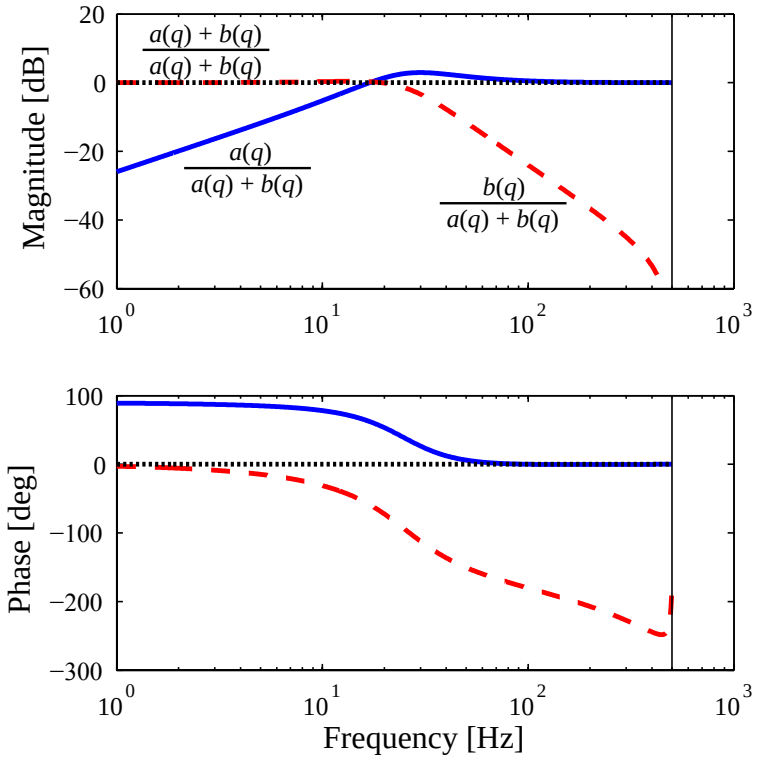


Figure 7.7: Frequency response of chosen design for the bumpless transfer filter. Annotated transfer functions correspond to (7.2).

7.5.1 Training Set

ILC applied to the *training set* trajectory given in Fig. 7.5 reduces the tracking error by almost 75% in the x-axis and over 40% in the y-axis, in terms of error RMS, (5.2), Fig. 7.9. Improved performance is also demonstrated in the spatial domain. Fig. 7.10 displays a contour plot. Whereas the system using just feedback, $y^{L(0)}(k)$, does not accurately track the high-frequency corner locations, applying ILC significantly improves tracking, $y^{L(30)}(k)$. Results in Fig. 7.9 and 7.10 are typical of ILC applied to positioning systems.

This work aims to expand on typical applications of ILC. From this *training set* data, input signal information can be extracted and applied to a more general set of trajectories. Fig. 7.11 demonstrates the segmentation of the input signal applied

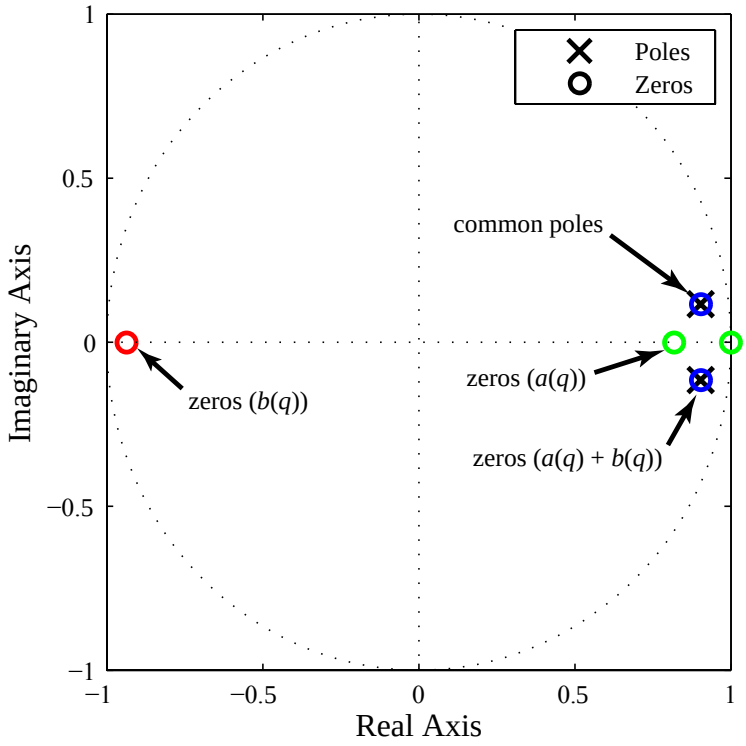


Figure 7.8: Pole-zero map of the chosen bumpless filter design. Note that after a task transition, the filter transfer function switches to one in which the zeros of the system perfectly cancel the poles.

to the x-axis at Iteration 30, $u_x^{L(30)}(k)$. Signal segmentation is a engineering design problem. It is important to capture enough signal content such that the tracking improvements from ILC are fully realized. However, the longer the chosen *basis signal* segments, the less flexible BTILC becomes. Given the non-causal nature of ILC it is important to chose a demarcation point prior to the actual task event of interest.

7.5.2 Operation Set: $\mathbf{T} + \mathbf{O} = \mathbf{T}$

For *operations sets* chosen such that $\mathbf{T} + \mathbf{O} = \mathbf{T}$, the BTILC input signal, $u^O(k)$, is almost identical to the input signal from a typical implementation of ILC, $u^{L(30)}(k)$, Fig. 7.12. This is because the transition sequencing has been maintained and there-

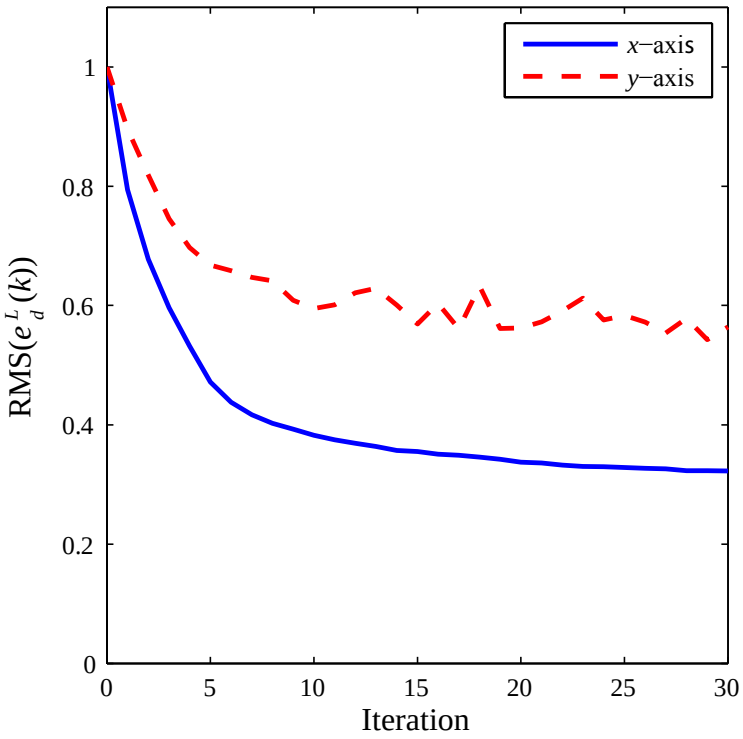


Figure 7.9: Iteration domain results for the training set.

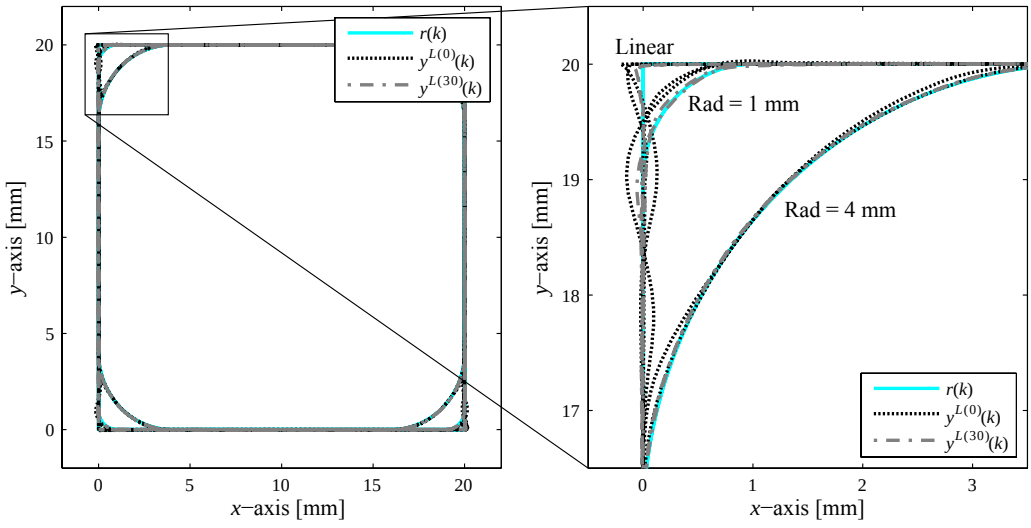


Figure 7.10: Contour plot of tracking performance for select iterations of the training set. Without ILC, $y^{L(0)}(k)$, the system poorly tracks the reference trajectory at the high-frequency locations. After 30 iterations, the tracking performance is much improved, $y^{L(30)}(k)$.

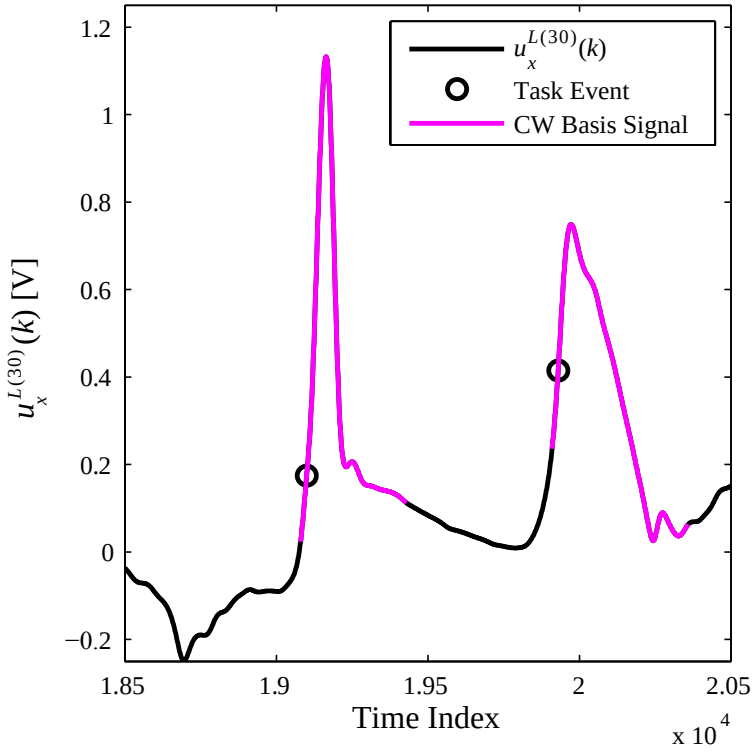


Figure 7.11: Segmentation of two CW basis signals from the ILC input signal generated from Iteration 30 in the *Training Set*.

fore the state dissimilarities at the transitions, δx_n in (3.12), are minimal. Fig. 7.15 displays that the RMS of the entire error signal degrades less than 21% in the x-axis by using BTILC for control. Details are found in Table 7.3. In fact, BTILC has a RMS error tracking performance better than RMS $\left(e^{L(10)}(k)\right)$ on average for the x-axis.

Since the state dissimilarities are small, (3.12), the transitions between *basis signals* are smooth and therefore bumpless transfer modifies the input signal minimally. Fig. 7.12 demonstrates that bumpless transfer input signal, $B\left(u^O(k)\right)$, is almost identical to the *operation set* input signal, $u^O(k)$. Bumpless transfer reduces RMS of the x-axis error by 4%, Table 7.4.

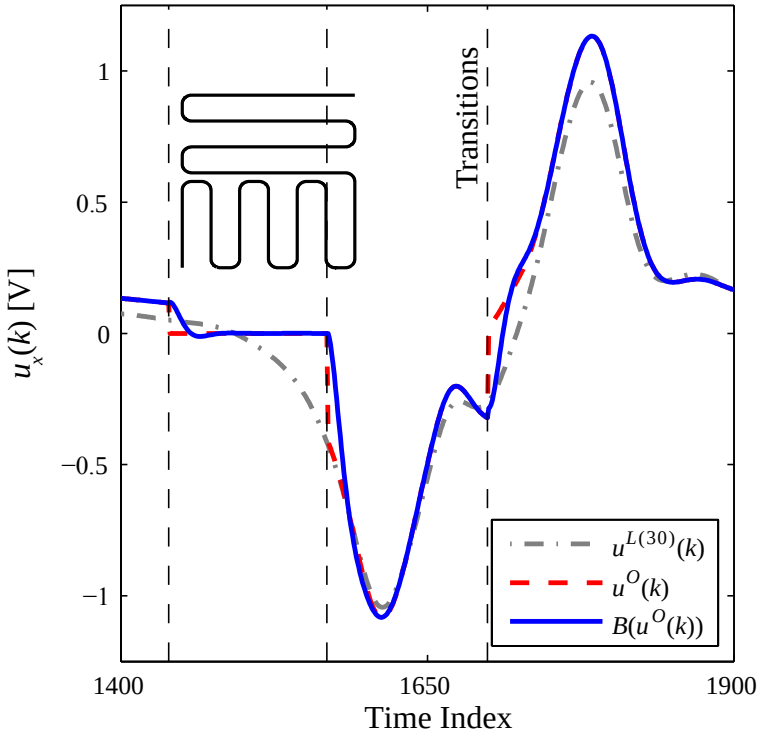


Figure 7.12: *Operation set basis task* sequences that have *set equivalence* have smooth transitions. Bumpless transfer filtering has only a small affect at transitions.

7.5.3 Operation Set: $\mathbf{T} + \mathbf{O} \neq \mathbf{T}$

When there is not set equivalence, $\mathbf{T} + \mathbf{O} \neq \mathbf{T}$, the state dissimilarity, (3.12), is large. Therefore, the identified *basis signals* are not properly designed for the given *basis task* states, x_n^O. This result is demonstrated in Fig. 7.13 in which the ideal ILC signal, $u^{L(30)}(k)$, has a considerably different signal shape than the concatenated *basis signals*, $u^O(k)$. Bumpless transfer helps bridge the gap between *basis signals*, better approximating $u^{L(30)}(k)$ at the transition points. The influence of bumpless transfer can be seen in Fig. 7.14 where the tracking error for $B\left(e^O(k)\right)$ is comparatively less than $e^O(k)$; whereas BTILC approaches feedback performance following transition indices, BTILC with bumpless transfer performs significantly better than feedback. Tracking performance does degrade, as given by Table 7.3 and Fig. 7.15, however

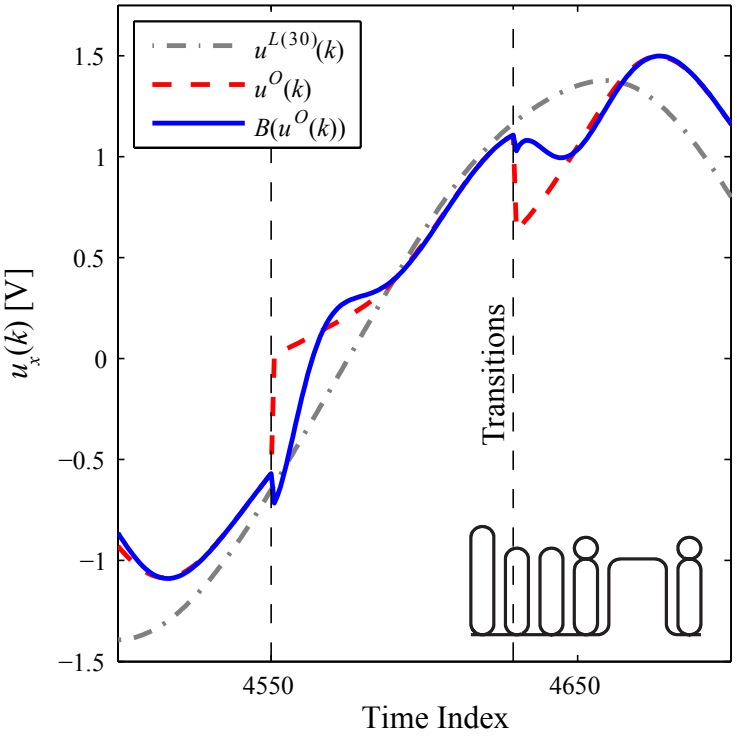


Figure 7.13: *Operation set basis task* sequences without *set equivalence*, $\mathbf{T} + \mathbf{O} \neq \mathbf{T}$, have discontinuous transitions. Bumpless transfer filtering modifies transitions, smoothing out input signal and better approximating the ILC input signal, $u^{L(30)}(k)$.

bumpless transfer does regain some performance losses. Applying bumpless transfer to BTILC gains a performance improvement of over 12% on average in the x-axis, Table 7.4, in terms of the RMS error. This comparison is shown in a different form in Fig. 7.16. Bumpless transfer improves BTILC, especially in the x-axis and when there is not *set equivalence*. Also, note that the control is remarkably consistent; the standard deviation, denoted by error bars, is very small in comparison to the total RMS Ratio. Even with $\mathbf{T} + \mathbf{O} \neq \mathbf{T}$, Fig. 7.15 shows that bumpless transfer provides tracking performance that is better than RMS $\left(e^{L(14)}(k)\right)$ on average, in the x-axis.

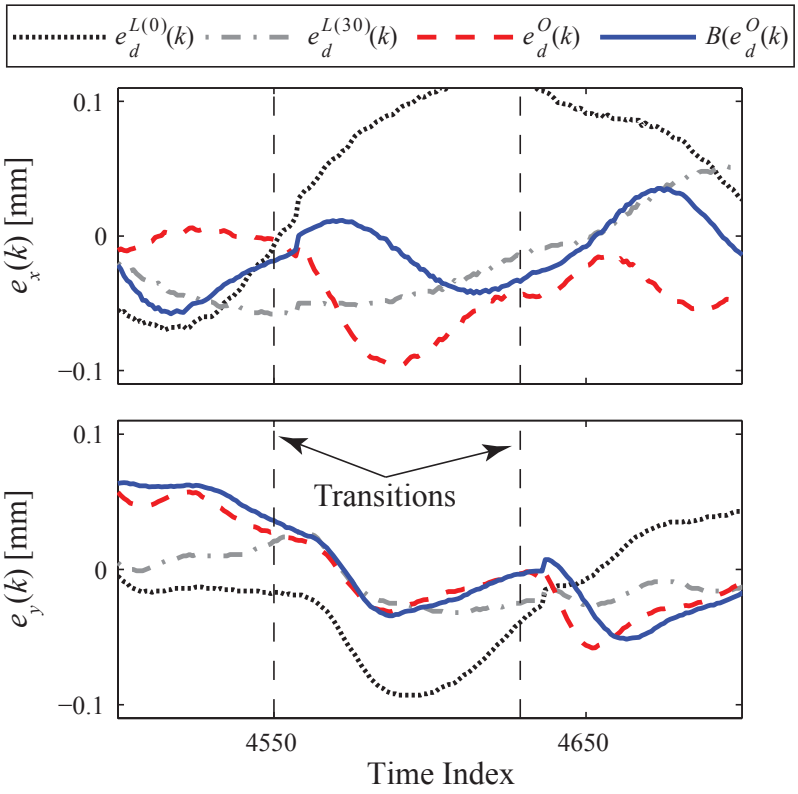


Figure 7.14: Error signal comparison. Compared to BTILC, $e_d^O(k)$, applying bumpless transfer for BTILC reduces error transients at the task transitions.

Table 7.3: Comparison: $\frac{\text{RMS}\left(B\left(e_d^O(k)\right)\right)-\text{RMS}\left(e_d^{L(30)}(k)\right)}{\text{RMS}\left(e_d^{L(30)}(k)\right)} \times 100\%$

	$\mathbf{T+O=T}$		$\mathbf{T+O\neq T}$	
Axis	(1,1)	(1,2)	(2,1)	(2,2)
x	19.34	22.15	57.83	30.58
y	79.55	41.61	-9.52	48.46

Table 7.4: Comparison: $\frac{\text{RMS}\left(e_d^O(k)\right)-\text{RMS}\left(B\left(e_d^O(k)\right)\right)}{\text{RMS}\left(e_d^O(k)\right)} \times 100\%$

	$\mathbf{T+O=T}$		$\mathbf{T+O\neq T}$	
Axis	(1,1)	(1,2)	(2,1)	(2,2)
x	0.37	7.64	5.03	19.59
y	2.82	1.24	2.69	-3.84

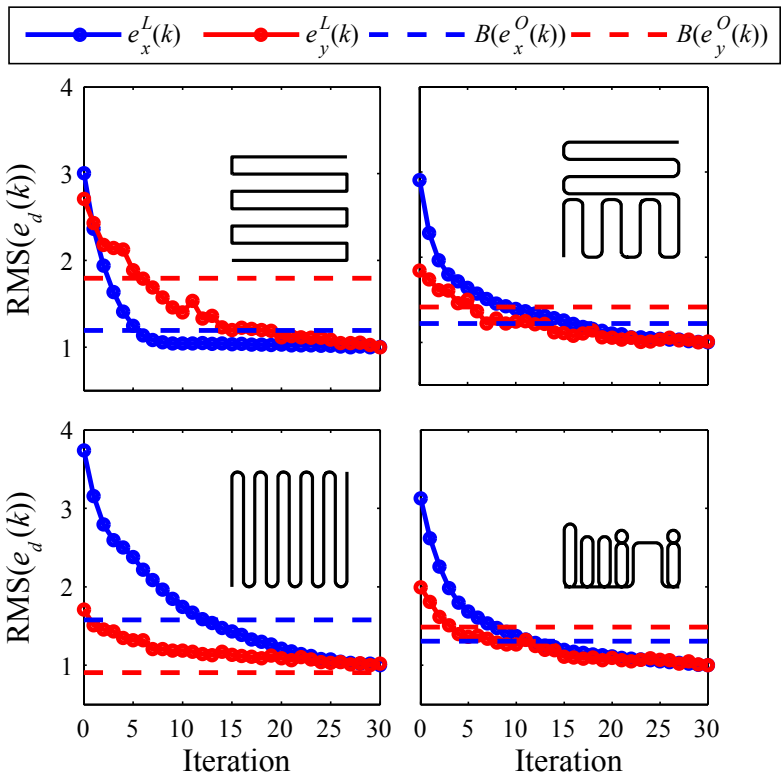


Figure 7.15: RMS error for each *operation set*. All data is normalized such that $\min_j \left(\text{RMS}\left(e_d^{L(j)}(k)\right)\right) = 1$. $B\left(e_d^O(k)\right)$ data represents mean performance for 5 trials. This data spans the Iteration axis for the sake of comparison, although it represents the performance achievable without having to iteratively apply and update an input signal.

7.6 Discussion

The results display a marked improvement over feedback control. Depending on the application performance requirements, BTILC presents a viable option for control improvement for *operation sets* that have weakened trajectory constraints. Furthermore, performance can be improved at instances where there is a *basis task* transition that is not equivalent to the *training set* by applying bumpless transfer. The results display how BTILC is capable of significant tracking performance improvements over feedback with a wide variety of *operation set* reference signals without retraining the system for a new trajectory. In both axes, $\text{RMS}\left(B\left(e_d^O(k)\right)\right) < \text{RMS}\left(e_d^{L(11)}(k)\right)$ on

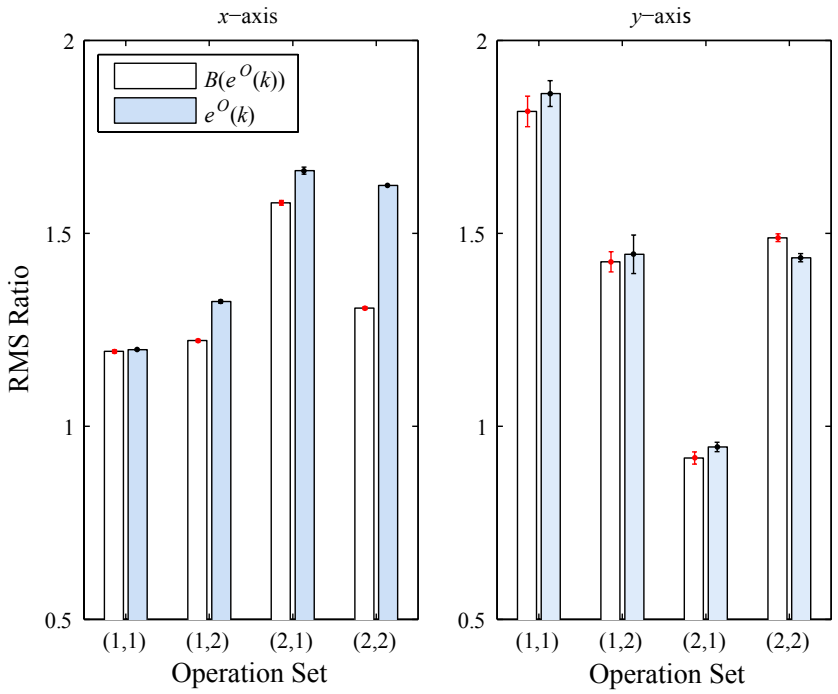


Figure 7.16: RMS Ratio for $B\left(e_d^O(k)\right)$ and $e_d^O(k)$. RMS ratio is defined as RMS $\left(B\left(e_d^O(k)\right)\right)/\min(L)$ and RMS $\left(e_d^O(k)\right)/\min(L)$, respectively, where $\min(L) = \min_j \text{RMS}\left(e_d^{L(j)}(k)\right)$. Bars represent the mean performance for 5 trials and error bars represent the standard deviation.

average; therefore 11 iterations of tracking performance is achievable just by intelligently using information already on-hand. In terms of the flexible manufacturing systems given in Section 7.1, significant improvements in toolbit positioning can be realized while maintaining the inherent flexibility of their workflows.

The design of the bumpless transfer filter $b(z)/a(z)$ presents a tradeoff between a fast response with minimal signal attenuation and a slow response so that large signal discontinuities can be bridged. A key example of this tradeoff can be seen in Figs. 7.12 and 7.13. The fast filter response ensures a smooth transition with quick convergence to the subsequent *basis signal* when $\mathbf{T} + \mathbf{O} = \mathbf{T}$. However, the system reacts too quickly when $\mathbf{T} + \mathbf{O} \neq \mathbf{T}$ and does not fully attenuate the transient. Future work will investigate this tradeoff and filters designed to account for transition characteristics.

Chapter 8

Conclusion

Additive Manufacturing (AM) processes are inherently flexible fabrication tools and are therefore adept at manufacturing patient specific synthetic tissue grafts. This work investigated a specific AM process, micro-Robotic Deposition (μRD). μRD had been typically relegated to the fabrication of *monolithic* structures. We have developed a new manufacturing control methodology that elevates μRD beyond *monolithic* capabilities and towards the fabrication of advanced architecture calcium phosphate scaffolds for skeletal defect repair.

These new manufacturing capabilities were enabled by modifying an established manufacturing control method, Iterative Learning Control (ILC), to be more amenable to AM processes. We term this flexible adaptation of ILC, the Basis Task Approach to ILC (BTILC). BTILC reorients the traditional ILC algorithm as a task specific input signal identification method. By decomposing trajectories into basis tasks, two constraints that rigidly limit ILC flexibility are relaxed: trajectory and dynamic invariance. The BTILC algorithm was evaluated throughout the thesis with the manufacture of synthetic bone scaffolds. In addition to the μRD process, BTILC was further evaluated with an alternate system, investigating performance with the general positioning problem. In particular, the positioning study investigated the sequencing of individual basis tasks. A novel bumpless transfer filter for open loop signals was developed that modifies the supplementary input signal at basis task transitions so that it better approximates signals identified by standard ILC algorithms.

Since the BTILC algorithm is an original approach to ILC, there remain some open

questions that we will investigate in the future. One such study will further investigate the equivalence between the training and operation sets. This includes developing methods to estimate the states at the task transitions time indices and then using knowledge of the system inverse dynamics to compensate for a state dissimilarity between the training and operation sets.

Importantly, the manufacturing control capabilities displayed here can be applied to the fabrication of complex scaffolds. Scaffolds can be fabricated within molds that support large unsupported features to enable near-net shape fabrication. Seven unique scaffolds are fabricated from a nearly identical manufacturing routine, displaying that the designed manufacturing work flow is efficient and flexible. The fabricated scaffolds display sufficient tolerances on architectural features, distinct divisions between porosity treatment regimes, contoured shapes, and large internal cavities.

We believe that flexible manufacturing systems such as the one developed here are well suited to replicate natural bone function with high-fidelity. With the ability to strictly control the placement of dissimilar materials comes the ability to functionalize bone scaffolds. Domains of different macroporosities, material compositions, and material chemistries (e.g. hydroxyapatite and β-tricalcium phosphate) are all envisioned as methods in which bone scaffolds with tailored funcationality can be designed and manufactured. Additionally, we will further investigate the use of molds in the manufacturing work flow that enable near-net shape fabrication. Of critical importance is a full evaluation of the types of near-net shape features that are feasible with our manufacturing process.

Appendix A

Build Material Fabrication Protocol

Hydroxyapatite Ink Fabrication Protocol

Last updated: 11–12–2007 by David Hoelzle

Written by Sheeny Lan and David Hoelzle

A.1 General Notes

- Ink performance, and viscosity, is most sensitive to the solids loading. Make sure to scrape all solids from weighboats as best possible to minimize sources of error. Also, take extra care when weighing the wet and dry samples after centrifuging.
- Do not fill centrifuge tubes all the way up. The higher level causes a high level of solid separation, causing agglomerates to form. Limit centrifuge fill to 2/3rds full.
- Send your completed spreadsheets back to Dave Hoelzle so he can keep them on record.

A.2 Procedure

Day 1

1. Calcine powders for 10 hours at 1100C using the large furnace in 111 MEB. Contact Dave Hoelzle concerning use of kiln if it is your first time.

Day 2

1. AM. Ball mill for 14 hours in ethanol

   (a) Add 150 g HA and 300 mL of ethanol with all the grinding media to large ball milling Nalgene jar. Ball milling Nalgene jar has been turned black from multiple ball milling operations, only use this jar.

   (b) Place jar into can and tape opening so that jar cannot fall out of can

2. Place a sieve over a large dish - pour HA suspension and media out of bottle onto sieve and rinse out bottle with ethanol

3. Rinse sieve and media well with ethanol such that most of the HA is flushed into the dish

4. Put dish in small blue oven (MEB 111) to allow powder to dry (≈1 day at higher temperature settings)

Day 3

1. A good reference paper is Michna, S., Wu, W., Lewis, J.A., "Concentrated hydroxyapatite inks for direct-write assembly of 3-D periodic scaffolds," *Biomaterials,* 26 (2005) pp. 5632-5639. Procedure follows paper except for the addition of PMMA, filtering step, and pH target value.

2. Based on the mass of HA powder needed, use spreadsheet to determine volume of water and Darvan 821A to add. Contact Dave Hoelzle for spreadsheet. Generic spreadsheet has been added to ABBLab gmail account.

3. Grab a clean beaker and put in water and appropriate amount of Darvan 821A with magnetic mixing bar

4. Adjust pH to 10 using 5M NH_4OH. Increments depend on size of batch. Point of reference: 100g batch requires ≈900 mL of base to reach pH 10.

5. Add 1/3 of HA powder

6. Put parafilm over the beaker opening and sonicate for 3 min.

7. Add next 1/3 of HA powder and sonicate

8. Add last 1/3 of HA powder and sonicate

9. Slowly pour HA mixture from beaker into a Nalgene bottle (can add DI water to get remaining HA out of beaker)

10. Put bottle on paint shaker for 50 minutes

11. Sonicate for 4 min

12. Transfer slurry into centrifuge tubes such that all tubes are filled with the same volume (can use DI water to rinse remaining HA out of bottle). Do not fill centrifuge tubes more than 2/3 full. Overly filled centrifuge tubes cause too much separation and particle consolidation occurs prematurely.

13. Centrifuge at 2000 rpm for 60 min

14. Rinse out Nalgene bottle, put tape on bottle and label it, add media – measure mass of bottle (with lid on) – record

15. Pour excess water from centrifuge tubes and then scoop HA out into bottle
16. Put bottle on paint shaker for 60 minutes
17. Mass a small weighboat
18. Take ≈2 gram sample and put into weighboat (record relevant masses on spreadsheet)
19. Store bottle of HA in fridge
20. Place weightboat sample into furnace at 35C for at least 12 hours

Day 4

1. Measure mass of weighboat + dry sample and calculate HA solids loading/volume percent
2. Using spreadsheet determine mass of PMMA, volume of additional water needed, Methocel, and 1-Octanol
3. Add in PMMA (mass out in weighboat and then transfer)
4. Add water (with pipette)
5. Add methocel (with bottle on balance)
6. Add 1-octanol
7. Place on paint shaker for 30 minutes
8. Add HNO_3 to decrease pH and increase viscosity (viscosity is what is most important here) – add HNO_3 10-20 mL at a time – measure pH and shake for 10 minutes between additions. Addition increments based on ink volume.

9. When viscosity seems about right, add 50 – 100 mL of PEI and then shake for 10 minutes

Appendix B

Basic Deposition Protocol

Last updated: 9-14-11 by David Hoelzle

Written by Amanda Hilldore and David Hoelzle

B.1 Warnings

1. Familiarize yourself with the Emergency Stops (E-Stop) (Fig. B.1). These are the red buttons located on each side of the robot. When the E-Stop is hit, all power is cut off to the micro-Robotic Deposition (μRD) machine. After an E-Stop has been hit, these buttons have to be twisted to be deactivated and then the machine has to be reset at the Power On location.

2. The linear motors which drive the stages on the μRD machine are very powerful and can move the stages at speeds greater than 1 m/s. This is more than enough power to kill you. **Never stick your head in the μRD with the amplifiers activated.** The amplifiers are ***active*** when you click the ***Start*** button in the WinCon Server.

3. If you have a problem that is not addressed in this protocol, you think you've broke something, or you are not sure what to do, do not hesitate to find or call Dave Hoelzle. No matter what time of the day. It is much better to solve your problem correctly then to put the μRD out of commission for a few weeks. Dave's cell phone number is 614-256-7388.

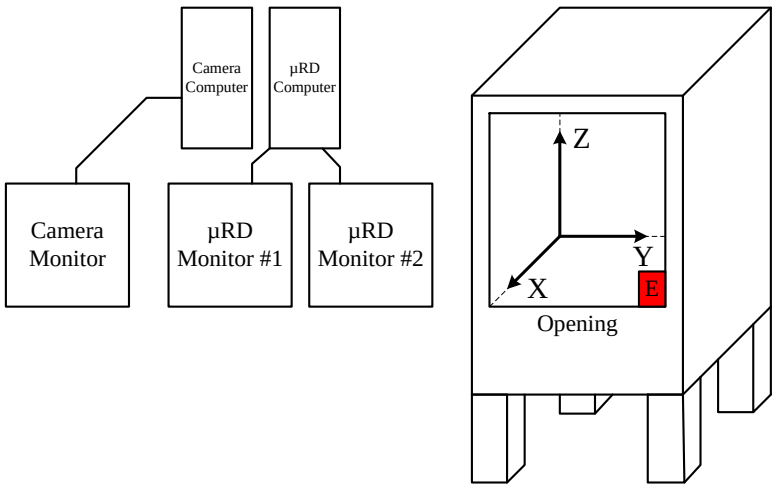


Figure B.1: Axes and computer layout. Axis directions are also displayed on the GUI. Notice the E-Stop in the bottom RH corner of the μRD opening.

4. Please do not modify the μRD or computer interface without first contacting Dave Hoelzle.

B.2 General Information

Robot and Workspace Layout:

The μRD and interface was designed to be a customizable controls test bed and is therefore not very user friendly. To help the user quickly understand some of the caveats of the μRD operation, this section provides a general overview. Detailed step-by-step instructions will be given in the procedure section. Below are layouts for robot space (Fig. B.1), robot image (Fig. B.2) WinCon Interface (Fig. B.3), Graphical User Interface (GUI) (Fig. B.4), Matlab Simulink diagram (Fig. B.5), and scaffold manufacture (Fig. B.6).

Note:

The robot is setup for functions more complicated than making simple lattices. Therefore there are many buttons you will not use. Avoid clicking these buttons so you do

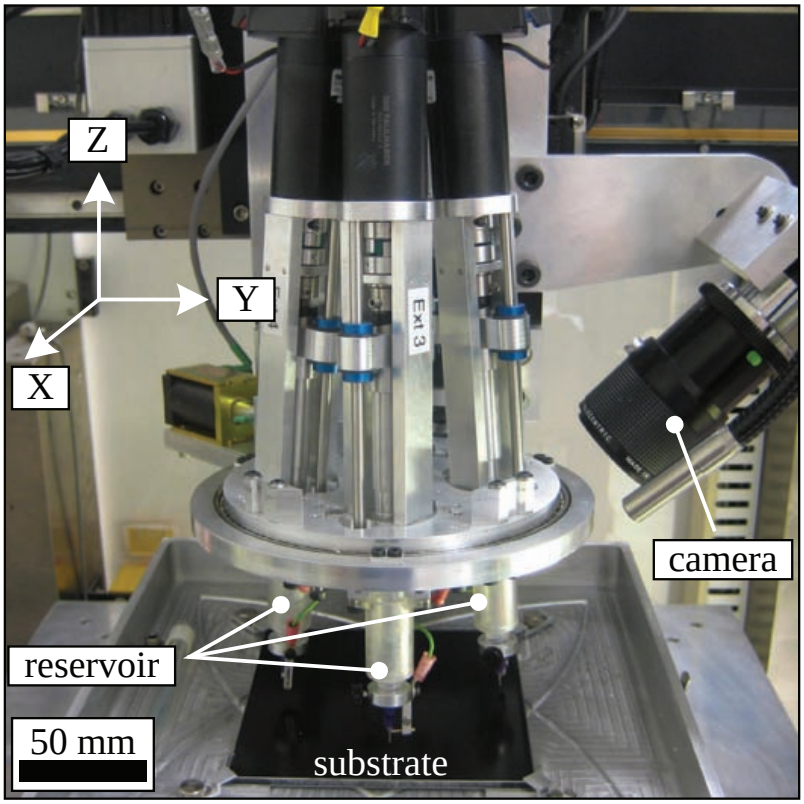


Figure B.2: Image of μRD robot with multi-nozzle deposition head attached.

not have unexpected results. Also, you will only need to use extrusion system 1.

B.3 Software Overview

The μRD is controlled using a graphical program called Simulink, which is embedded in the program Matlab. Another program, Wincon, is used to interact between Simulink and the physical robot.

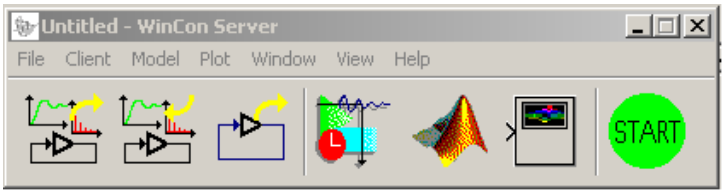


Figure B.3: WinCon Server. The only essential buttons on this window are the Start button and the Plots (just left of Start) button. The Start button turns on the amplifiers so make sure all body parts are clear of the robot before pressing Start. The Plots button opens up signals scopes for systems diagnostics and data recording.

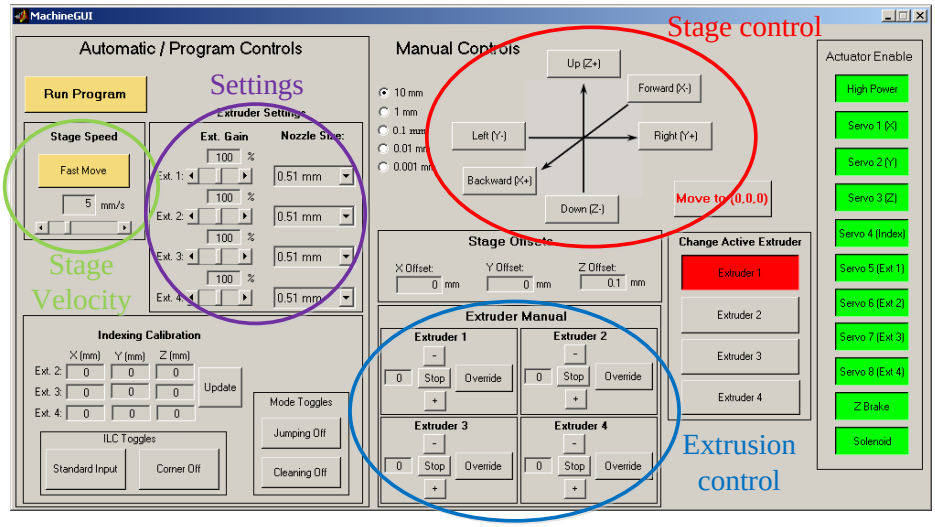


Figure B.4: GUI. Stage Control; Used for general positioning. Stage Velocity; Sets stage velocity. Extrusion Control; Controls extruder displacement and speed.

B.3.1 GUI Operation

Manual Stage Control

This is the most used portion of the GUI. The μRD axes are moved by clicking on the directional arrows. The increment in which the stage moves with each click is determined by the radio buttons in the upper LH corner of the stage control area.

PLEASE READ THIS NEXT PARAGRAPH!!

The *Move To (0, 0, 0)* button can be tricky and if not used properly may damage

scaffolds or the robot. When the *Move To (0, 0, 0)* button is clicked, the robot is directed to move to position (0,0,0), which is the location the robot was at when the Start button was initially pressed. The robot will take the shortest path to (0,0,0) which may intersect with your deposited scaffold or solid object. Until you are familiar with how the robot behaves, do not press this button while the amplifiers are on. Additionally, always make sure the to reset the (0,0,0) point by clicking *Move To (0, 0, 0)* when the amplifiers are off. For instance, if you press the Start button when the display reads (20, 30, -30), the robot will move to location (20, 30, -30) relative to its position when Start button was clicked. **As a rule of thumb, NEVER press *Move To (0, 0, 0)* when the amplifiers are running, and ALWAYS press *Move To (0, 0, 0)* just before turning on the amplifiers.**

Stage Velocity

This slider bar sets the velocity at which the stage travels at. Typical deposition speeds are 5 and 10 mm/s. Clicking the Fast Move button quickly increases the velocity to 30 mm/s.

Extrusion Control

During typical operation, the program being run dictates the speed that the extruders move. When running a program, all extruder override buttons should be clicked off. In between programs, you may desire to raise or lower the extruders to load or unload ink, or test the ink flow. To move the extruder, the Override button must be active. The up and down arrows then control the speed at which the extruder moves. The speeds are arbitrary, but a speed of 1 corresponds to the proper extrusion velocity required to deposit at a tip speed of 5 mm/s. Just a note, to make the programming easier, the axis is reversed from the Z stage axis. Positive velocities mean that the

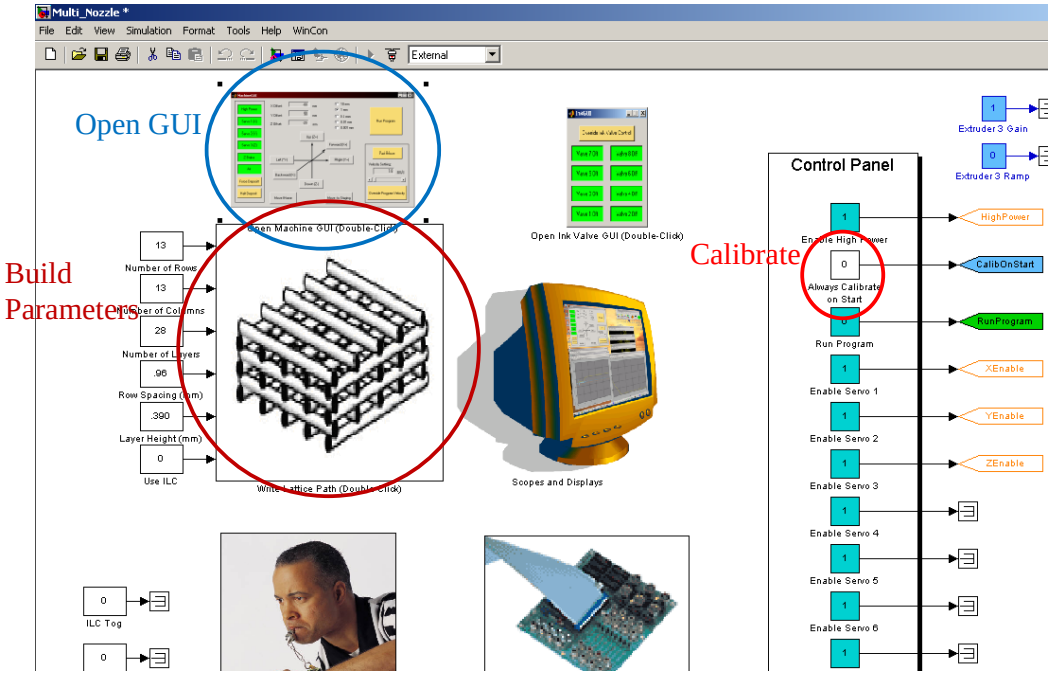


Figure B.5: Simulink diagram. Most parts of this diagram should not be altered. Important parts are circled.

extruder is driving downwards, expelling positive amounts of volumes of ink.

Settings

This section modifies how each extrusion system moves. Select the nozzle size you are using from the pull-down menu. The Ext. Gain slider is rarely used so leave it at 100%; it just modifies the extruder speed during a program.

Other Buttons

Buttons not mentioned here are extras used for other purposes and should not be clicked.

B.3.2 Simulink Diagram Operation

Open GUI

Double clicking this box opens the GUI. If the GUI is already open and you try to open a second GUI the program will give you an error. This is mentioned again in the Troubleshooting Section.

Build Parameters

These boxes set the dimensions of the lattice you want to build in mm units. After the appropriate units are entered, the lattice picture must be double clicked to build the vector V. V is an ordered list of instructions for the robot. After double clicking the lattice icon, the vector V will then be displayed in the Matlab Workspace. If after depositing a lattice of one dimension and you wish to change dimensions, you must double click the lattice image to rebuild the vector V. Also, you will have to rebuild using WinCon; described in the procedure section.

Calibrate

The operation of this section is described in the Trouble Shooting section. In general this box should always be set to 0.

B.4 Deposition Procedure

1. Loading ink into a syringe

   (a) Thread syringe cap onto the bottom of the syringe.

   (b) Put lamp oil in the syringe and coat the walls. Discard extra.

   (c) Add ink. The amount depends on the lattice being made.

(d) Put the snap cap on.

(e) Centrifuge at 3000 rpm for 3 min.

i. Make sure to counter-balance with water if necessary.

ii. Screw the syringe into the cap of the centrifuge tube.

2. Prep μRD.

(a) Clean substrate with ethanol.

(b) Spray substrate with an even coat of hairspray.

(c) Let substrate dry.

(d) Place substrate on square risers in oil bath.

(e) Place substrate clamps on substrate corners and tighten with thumb screws.

(f) Fill bath with oil, deep enough to fully immerse lattice.

3. Turn on machine.

(a) Push the power on button.

(b) Wait a few seconds.

(c) Push the power on button again. The green light on the top should be lit.

4. Prep computer.

(a) Open Matlab 6.1.

i. Make 'D://hoelze2/DaveDeposition w/o ILC' the current directory.

ii. Double click 'Multi_Nozzle.mdl'.

(b) Define lattice architecture

i. Enter lattice build parameters.

ii. Double click lattice icon.

iii. Make sure 'V' is in the Workspace.

(c) Wincon (at the top menu) Clean.

(d) Wincon Build.

(e) Double click GUI icon.

i. Make sure X, Y, and Z offsets = [0, 0, 0.1]!!

ii. If not, click '*Move To (0, 0, 0)*'.

iii. Change nozzle size to the correct size.

(f) Start the camera (Left computer).

i. Start Programs ATI Multimedia TV.

ii. Click the setup button that looks like a checkmark.

A. Stills gallery Browse My Documents\Lattices\Folder labeled as the current date.

5. Get ink ready.

(a) Add syringe to machine.

i. Take snap cap off.

ii. Put a red plunger in reservoir.

iii. Push down with allen wrench until air is gone but ink isn't coming out.

iv. Take off syringe tip cover.

v. Screw on a tip with the right diameter (0.51 = purple).

vi. Insert syringe into syringe holder in machine.

vii. Use syringe clamps to affix syringe. Do not fasten to tight.

viii. Manually move robot head to about the correct XY location.

(b) Recheck offsets (X,Y,Z) Offsets = (0,0,0.1).

(c) Click 'Start'.

(d) Put tip in oil relatively quickly so ink does not dry out (not too far down in the oil bath though).

(e) Shine light at tip.

(f) Move camera to tip and focus.

i. Camera does not react well to high intensity light. If screen flickers blue, adjust the light or camera aperture to be dimmer.

(g) Zero the tip.

i. Move camera to tip.

ii. A light above the closest E-Stop indicates when the tip is contacting the substrate. Using the system is optional, with practice the tip can be accurately zeroed by sight. If you use the light system, make sure the nozzle tip is clean because ink does not conduct electricity well.

iii. Zeroing the tip is an iterative process. First start by moving the tip close to the substrate in 1mm increments. When the tip is close switch to 0.1mm increments and move until you contact the plate. Once the tip is within 0.1mm increments, switch to 0.01mm increments until tip is once again just touching the substrate. Being within 0.01mm is adequate for lattice deposition.

iv. Once zeroed, you must move the tip up to the proper fly height. Move the Z axis up 0.77 times the nozzle diameter in mm for the 1^{st} layer.

(h) Get air out and check ink.

i. Click 'override' in Manual Extruder section.

ii. Move extruder down slowly until it contacts the red piston.

iii. Click 'Stop'.

iv. Run at 1 to see how ink behaves.

A. Let it run for a while.

B. Ideally see individual rows that form a cylinder.

v. Then move the tip in the XY plane and see.

A. how much the ink stretches (more = good).

B. if the cylinder tips over (good).

(i) Move the tip to the plate, but not the starting position.

(j) Fast lines (Optional, used for testing the ink).

i. Unclick 'Fast Move' (Velocity should be at 5 mm/s).

ii. Change to 1mm increments.

iii. Make sure override plunger is on.

iv. Change the plunger speed to 1.

v. Move nozzle to make sure you have a good line.

6. Starting lattice.

(a) Move tip to where you want the upper left corner of the scaffold to be (Fig. B.6).

i. Do not forget about the lead in lines – don't put the tip too close to the edge of the plate.

(b) Unclick 'Override' plunger speed.

(c) 'Run Program'.

(d) Move camera after lead-in lines and take pictures.

7. Ending lattice.

   (a) When the lattice if finished, move the tip in the +z and –x directions. This is because when Run Program is deactivated, the tip will return to the position at which Run Program was activated at, and will drag through your part.

   (b) Click 'Run Program'.

   (c) Click 'Override'.

   (d) Change the plunger speed to –(75-100).

   (e) Click 'Stop' when the plunger reaches the top starting position.

      i. WARNING: there is no safety stop currently installed, so make sure to not go too far.

   (f) Click 'Stop' in WinCon window.

   (g) Manually move the head close to you (and not over the oil bath).

   (h) Remove and discard syringe.

B.5 Troubleshooting

This is an incomplete list of all the computer errors that you may encounter. If you come across any errors not listed here, email Dave Hoelzle and also record the error so it can be added to this section.

Vector V not built

Matlab needs the vector V to be in the Command Window to describe all the points in space which the robot must travel. Without it the trajectory cannot be calculated. To fix this problem, simply input the proper lattice parameters and double click on the lattice picture in the Simulink diagram.

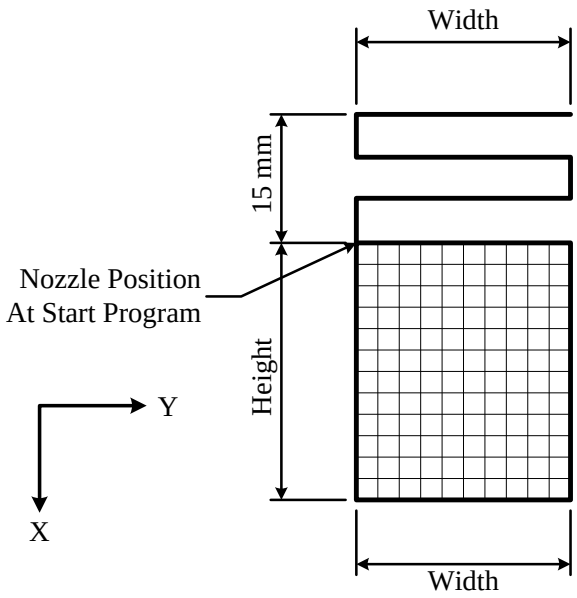
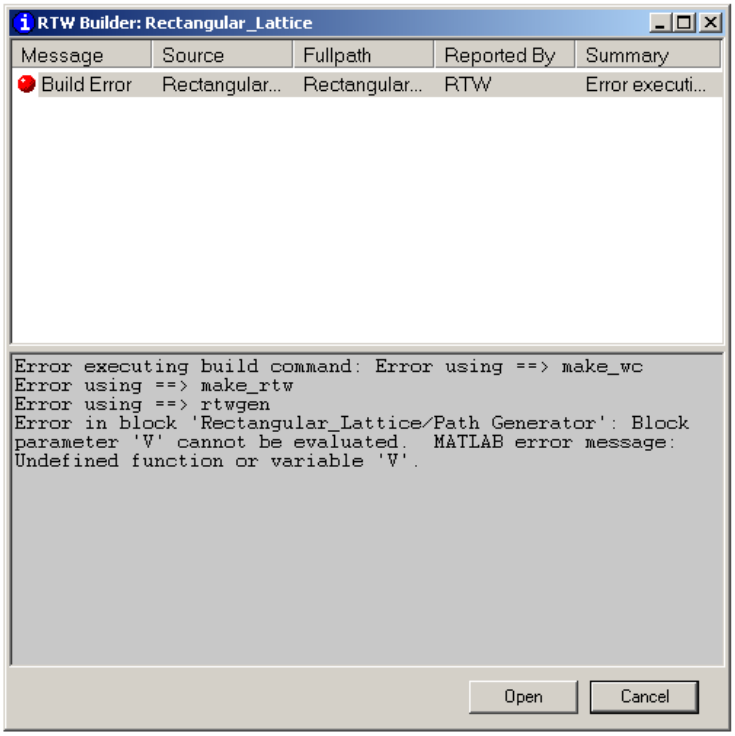
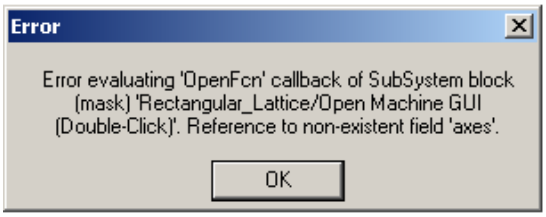


Figure B.6: Diagram of scaffold layout and build directions



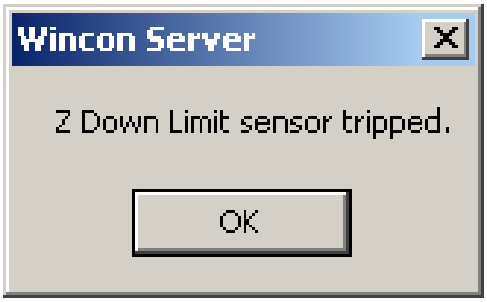
Too many GUI's open

An error is produced when the GUI symbol is double clicked when there is already



another GUI open. Close the error box and open the GUI that is already open.

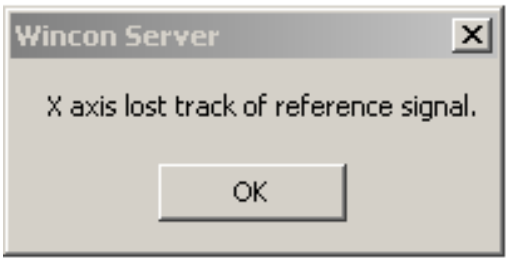
Limit sensor tripped



When one of the stages is moved to the extents of its motion a limit switch which protects the robot is tripped and cuts off power to the motors. When a limit switch is tripped the robot has to be recalibrated. In the Simulink diagram, change the 0 in the Always Calibrate On Start box to a 1. Check to make sure the Offsets have been reset to (0,0,0.1) by clicking 'Move To (0,0,0)'. Next click the 'Start' button on the WinCon Server. The robot will move to the back lefthand corner of the system, calibrate, then move to the middle of the system. Be sure to change the Always Calibrate on Start box back to a 0 so the system is not recalibrated every time it is turned on.

Lost axis tracking

The error between the reference position and the actual position has become greater



than 1 mm. This usually indicates that a robot stage has run into a solid object. Make sure that there is nothing in the way of the robot. Next hit the 'Start' button to active the amplifiers. Move the robot away from any blockages.

Extrusion System x lost tracking

Extrusion system x (where x is the extrusion system number) has reached it's limits. Either restart the system and move the extruder away from it's limit, or reach into the extrusion system and manually twist the lead screw away from the blockage.

Unexpected part built

Either the vector V has not been recalculated or the trajectory has not been rebuilt. Make sure that all the part dimensions are correct and are in the correct units. Double click the scaffold icon to recalculate vector V. Under the Wincon menu click Clean then click Build.

Bubbles are hydrolyzing out the ink

The switch for the zeroing light is still on, passing electricity through the ink, causing the ink to hydrolyze. Turn the switch off.

Appendix C

BTILC Protocol

Original Author: Samantha Polak; July 20, 2011

Last Edited: David Hoelzle; August 17, 2011

C.1 General Notes

- This document provides a complete, step-by-step procedure to apply the Basis Task Approach to Iterative Learning Control (BTILC) to the micro-Robotic Deposition (μRD) system. A conceptual description of BTILC is provided in: Hoelzle, et al., *TCST*, 2011, which are essentially Chapters 3 and 5 in the thesis.

- Application to advanced architecture structures is given in Hoelzle, et al., *ASME J Biomech Eng.*, 2011, which is essentially Chapter 6 in the thesis.

- A Basic Protocol is given in Appendix B. Please consult this document for basic μRD function.

- It is recommended that the user reads all three of these documents to gain deeper understanding of the process.

- Always perform all BTILC operations at the velocity you intend to build the structure. Operating at another speed will result in improperly identified basis signals and poor results.

C.2 Software Needed and Computer Setup

- *Software Needed:* Matlab, Cites VPN tool (if using a laptop), ImageJ (available for free online), an appropriate CODEC for DIVX format video (available for free online, consult Matlab help files), and the set of Matlab script files (script files available by request (hoelzle2@gmail.com); they are also provided in Appendix F).

- *Computer Setup:* This procedure requires you to frequently transfer image, video, and matrix format data. This is most easily done through the network. Network addresses: robot computer = \\mechse-alley-05.ad.uiuc.edu; camera computer = \\mechse-alley-35.ad.uiuc.edu

  - You must be logged in via a Cites VPN before trying to 'Run – computer address'

  - After 'Run' and entering the computer address you may be prompted to enter your password. Be sure the 'user name' is uiuc\'your_uin' before entering your password. e.g. uiuc\joecool2

C.3 Summary

This protocol is given in 4 tasks; please see the annotated version of Fig. 6.1 from Chapter 6 in Fig. C.1. Task 1–3 are sequential task in which each task is dependent on information from the previous task. Task 4 provides a shortcut method given that Tasks 1–3 have been previously applied to the given material system. The four tasks are:

1. Identification of system dynamics

2. Identification of basis task input signals (basis signals)

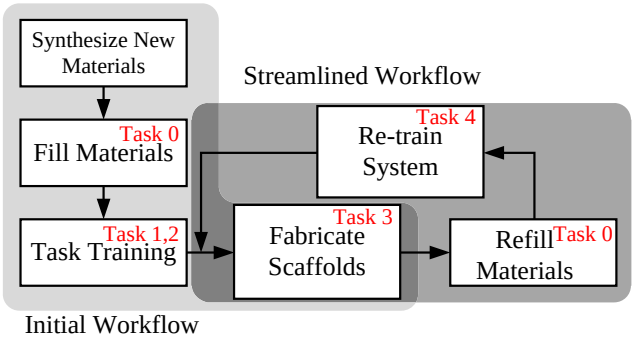


Figure C.1: Protocol workflow. This figure is a modification of Fig. 6.1 from Chapter 6

3. Application of basis signals to scaffold manufacture

4. Streamlined application of BTILC

C.4 General Start-Up Information

All BTILC work uses the Matlab model file MultiNozzleItLearn.mdl. This model file is in the directory: D:hoelzle2\Daves Deposition wo ILC 08-16-05

C.4.1 Start-Up

As stated in the Basic Protocol (Appendix B), when Start is clicked, the axes will move to the position given in the offsets. Please read the Basic Protocol (Appendix B) so you completely understand what will happen. Analogously, the rotational system will automatically index to the Active Extrusion system. The μRD does not know which extrusion system is which, so prior to clicking 'Start' you should always manually move Extrusion System 1 to the position closest to the back wall of the deposition system mount so it correctly identifies Extrusion System 1. Always ensure that Extruder 1 is the active system at startup.

There are two basic blocks you will need to familiarize yourself with:

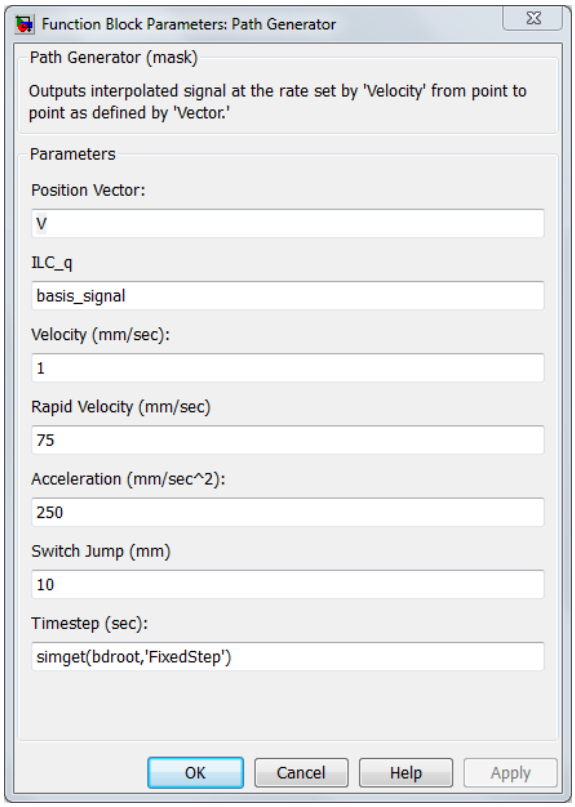


Figure C.2: Path Generator block

C.4.2 Path Generator Block

Training the system to control the fluid flow rate (Training Set in Hoelzle, *TCST*, 2011) requires regularly updates of databases that will be used for path and plunger profile generation. In the current setup, there is not an automatic method to update this database. You will have to manually update the databases. Fig. C.2 provides the Path Generator block. Within the Path Generator block there are two important databases:

1. Position Vector = machine language that defines the movement of 3 axes and coordinates the switching between different materials

2. ILC_q = basis signal library (signal information for the control of the plunger)

You are able to update these databases regularly, provided that the databases are of the same size. For example if you replace a 3002 x 16 size database with a 3002 x 16 database, the database will update successfully. If the size changes, the system will provide an error. Also, the name of the database must change for the computer to recognize the change. It is simple to change the name of database. For example, if you want to change a position vector `V`, however the new database has the same name as the old database, type `Vnew = V`; in the command window. Then type `Vnew` in the Position Vector line.

C.4.3 Graphical User Interface (GUI).

A basic GUI is described in detail in Appendix B. For manufacture using BTILC, additional features (Fig. C.3) are utilized:

1. Standard Input / ILC Input: This button toggles between using a plunger input signal that is a direct proportion of the axes speed (normal deposition mode) and a plunger input signal that is specified by a database generated by ILC or BTILC.

2. Corner Off / Corner On: This button toggles between recognizing a corner task in a scaffold build and not recognizing. When off, Corners are replaced with a simple Steady-State basis signal.

3. Jumping Off / Jumping On: This button toggles between simply moving to points in a structure and jumping (moving away from the structure in the positive Z direction, moving, then moving down) between points where there are breaks in the material flow.

4. Cleaning Off / Cleaning On: When off, nothing happens. When on, in between material extrusion portions, the nozzle will jump from the structure being fab-

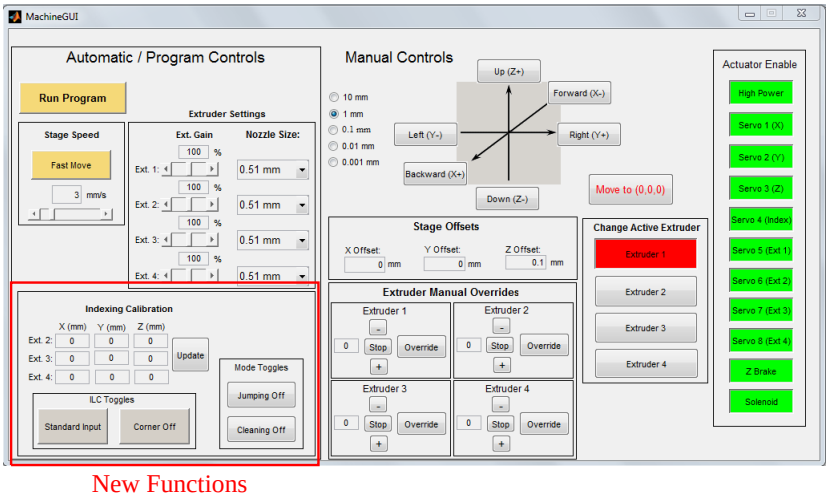


Figure C.3: Graphical User Interface (GUI)

ricated and a cleaning station to clean the nozzle tip. Each time you enable this function you have to specify where the cleaning station is. With 'Extruder 1' active, move the nozzle tip to the cleaning station and toggle on 'Cleaning On'. The computer will store this location. Each time a clean operation is run, the axes will jog through the cleaning station to wipe the tip. With each subsequent clean operation, the cleaning location shift 0.25 mm in the positive X direction to prevent a buildup of excess material in one location in the cleaning station.

C.4.4 Test Routines

There are two basic test routines you will use.

1. **Task 1.** For the identification of system dynamics: `LineTestVStop` or `LineTestVStopMatB`. Each will produce a position vector in the workspace, `V2a` or `V2b`, respectively. A is for Extrusion System 1 and B is for Extrusion System 2. This test immediately starts flowrate, moves the axes in the positive Y direction to build up steady-state flow, then performs a two-point turn and heads in the

negative Y direction. Just after the turn the flowrate is commanded to turn off to assess dynamics of ink shutoff.

2. **Task 2.** For basis signal identification: `LineTestV` or `LineTestVMatB`. Each will produce a position vector in the workspace, `V1a` or `V1b`, respectively. A is for Extrusion System 1 and B is for Extrusion System 2. This test move 1 mm in the positive Z direction, then proceeds in the negative Y direction. During move, the flowrate is pulsed, attempting to fabricate a thin cylinder. At the end of the pulse, the nozzle moves up in the positive Z direction and then back to the starting position.

C.4.5 ** The Tape Trick **

The BTILC Protocol is dependent on the correct registration of positions. Occasionally you will have to 'Stop' the μRD to upload new data. When you Stop, the Z axis falls from gravity momentarily before the Z-brake catches it, making your registration uncertain. There is a simple fix for this:

1. Switch the camera feed to robot
2. Blow up the camera computer screen to be full screen
3. Find a unique feature on the screen (not the nozzle, the camera moves with it) and place a piece of tape at the feature to mark its location in the camera's frame of reference
4. 'Stop' the μRD
5. Update data and rebuild
6. Click Move to (0,0,0). 'Start' the robot

7. move the Z axis until your feature and the tape align

8. this is your new registration point, reference from this point accordingly

C.5 Task 0: Loading Ink into μRD system using BTILC

The red pistons need to be physically connected to the μRD plunger to accurately control flowrate with BTILC. There are specially designed bottom attachments for the plungers to positively lock with the red pistons.

1. Thread 1 red piston onto the threaded screw at the bottom of the plunger

2. Load syringe (tip cap on, no sleeve, no screws)

3. Remove air from syringe

   (a) Move piston at speed 50 downward

   (b) When plunger get close to ink - slow to 15

   (c) When oil begins to leave past piston - slow to 1

   (d) Gently pinch syringe body to let the oil and air slide past piston. There should be little to no air in between the ink and the piston. Slight leakage of ink past piston during pinching is acceptable.

4. Put clear plastic syringe sleeve on (and screws) and extrude ink to the level of the collar/sleeve

5. Put appropriately sized nozzle on

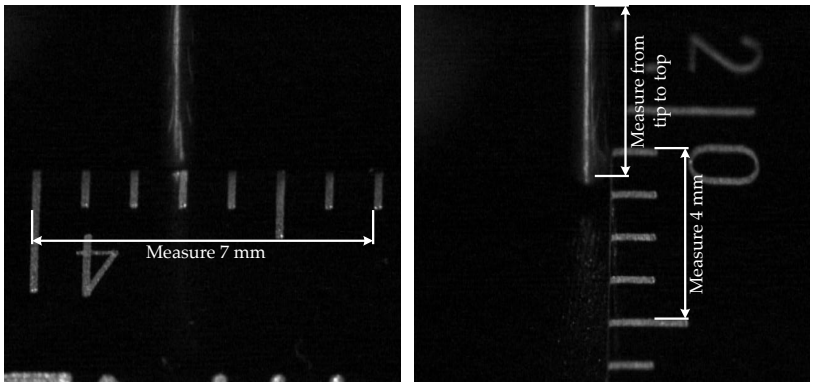


Figure C.4: Machine vision system Calibration

6. Zero tip in the center of the plate, move 0.420 mm in the positive Z direction. This is your Z offset position.

7. Record Z offset in your lab notebook

8. Take pictures (on camera computer; robot video feed) with the ruler (Fig. C.4) – one horizontal and one vertical

9. Move tip off the plate (over the factory rounded edges), move tip down 1mm

10. Fill tray with oil using gravity fed system until oil just touches the tip

11. Run at extrusion system at speed 1 to get the bubbles out and ink flowing well

C.6 Task 1: Identification of System Dynamics

The chosen Iterative Learning Control (ILC) algorithm uses information about the system dynamics to update the input signal from iteration-to-iteration. We must first identify a mathematical model, (C.1), that relates the plunger velocity (system input) to the material extrusion rate (system output). In the frequency-domain, this

model will be a first-order system.

$$\frac{Q_{out}(s)}{Q_{in}(s)} = \frac{K}{\tau s + 1}. \tag{C.1}$$

For the specific test in Task 1, (C.1) can be written in the time-domain as:

$$Q_{out}(t) = KQ_{in}^{*}e^{-\tau t} \tag{C.2}$$

where Q_{in}^{*} is the nominal operating flowrate.

Procedure:

1. Open the correct Position Vector .m file and run the script.
2. Load databases
   (a) **For Extrusion System 1**
      i. Open `LineTestVStopA.m` → Run
      ii. Enter in Path Generator block:
         A. Position Vector = `V2a`
         B. ILC_q = `zeros(8003, 16)`
   (b) **For Extrusion System 2**
      i. Open `LineTestVStopB`. → Run
      ii. Enter in Path Generator Block:
         A. Position Vector = `V2b`
         B. ILC_q = `zeros(8003, 16)`
3. Wincon Clean
4. Wincon Build

5. Open Plots at the left of the Start button
    (a) Ext Velocity (adjust the Y-axis to [-20 20])
    (b) Axes Compile (auto-scale Y-axis)
6. Make sure Indexing Calibration values (lower left corner of control GUI) are all set to 0 and click Calibrate
7. Click 'Start': This will turn on the machine, ensure that you are away from all moving parts.
8. **For Extrusion System 2**
    (a) *The μRD references every position from Extrusion System 1. Compensate for this by:*
    (b) Move Extruder 1 to the active position (nearest the wall/head)
    (c) Make sure 'Extruder 1' is clicked as the active one in the GUI
    (d) Click 'Start'
    (e) Then click on 'Extruder 2' in the active extruder part of the GUI; Extruder 2 will automatically move to the active position.
    (f) You are now set to continue the rest of the program
9. Move tip to Task 1 start location (Fig. C.5) and the recorded Z offset height
    (a) Raise tip up 1mm (you lowered it so that it would be in oil – remember)
    (b) Move to Task 1 start location (Fig. C.5)
10. **Run 10 trials to measure the average response**
    (a) *First Time*
        i. Do not record first trial:

ii. 'Run Program' and observe trial

iii. Move 2mm in positive X direction

iv. Unclick 'Run Program'

(b) *Run 10 trials*; each time:

i. 'Record' on camera computer

ii. 'Run Program'

iii. Stop 'record' on camera computer

iv. Move 2mm in positive X direction

v. Unclick 'Run Program'

11. Change video names to 'Trial **x**' (x=1:10)

12. Measure the pixel/mm value for the calibration pictures (Fig. C.4) using the line tool in ImageJ (Crtl+m gives you the measurements):

(a) Horizontal picture – measure 7mm

(b) Vertical picture

i. measure 4mm

ii. measure the nozzle length

13. In `Video_Processing_CF.m` enter the distances you measured:

(a) `HorCalibrate = measured horizontal distance/7; %Horpixel/mm`

(b) `VertCalibrate = measured vertical distance/4; %Vertpixel/mm`

(c) `NozSpot = measured tip to top; % distance in vertical pixels to nozzle center, changes at any movement of camera`

14. Transfer Videos to 'System ID' folder on laptop

15. Run `Compile.m`: this will run `Video_Processing_CF.m` 10 times. Fig. 4.4 gives the basic algorithm. Two important actions are required by the user.

    (a) The first trial analyzed will require you to select a Region of Interest (ROI). Fig. C.6 gives a good region of interest selection.

    (b) A motion detection algorithm will automatically recognize the video frame when the robot commences movement. This works correctly approximately 95% of the time, but sometimes fails. If it fails, you can override the selection by hitting 0 when it asks if you are happy. Follow instructions to manually select the first moving frame.

16. Open and run DataCompile.m

17. Adjust `K` and `tau` (Line 18 and 19) to get the best model; it is most important to match the response immediately after the step down. Far from the step down the dynamics are very nonlinear and a linear model such as Eq. (C.1) is not appropriate.

18. Record K and τ; eg:

    (a) `K = 1.04;`

    (b) `tau = 2.7;`

Task 1 is complete and you have a model of system dynamics

C.7 Task 2: Identification of basis task input signals

1. Put in a new substrate in the robot – be sure not to get any oil on it

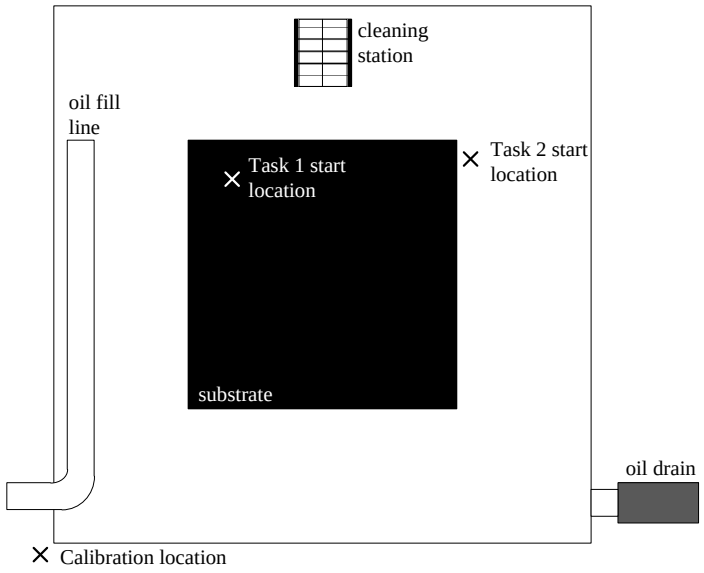


Figure C.5: Oil bath layout

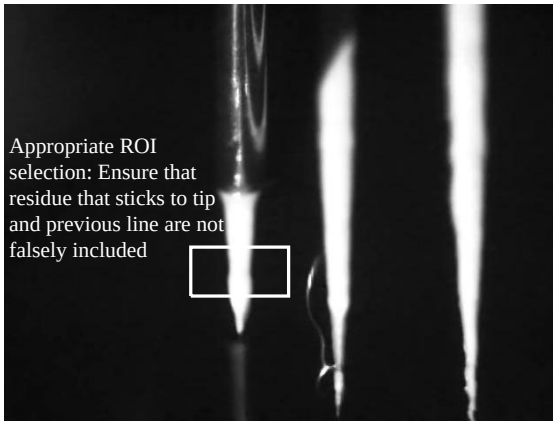


Figure C.6: Region of Interest (ROI) seletion.

2. Load databases

   (a) **Extrusion System 1**

      i. Run '`LineTestV.m`'

      ii. Switch the Position Vector to '`Vla`'

      iii. Ensure that the ILC_q vector is `zeros(8003, 16)`

(a) **Extrusion System 2**

i. Run 'LineTestVb.m'

ii. Switch the Position Vector to '`Vlb`'

iii. Ensure that the ILC_q vector is `zeros(8003, 16)`

3. Update `Video_Processing_CF.m` with the distances you measured from Task 1:

   (a) `HorCalibrate = measured horizontal distance/7; %Horpixel/mm`

   (b) `VertCalibrate = measured vertical distance/4; %Vertpixel/mm`

   (c) `NozSpot = measured tip to top; % distance in vertical pixels to nozzle center, changes at any movement of camera`

4. Update `ILC_Implement_MI_TV.m` with the K and τ values you recorded from Task 1. e.g.

   (a) `K = 1.04;`

   (b) `tau = 2.7;`

5. Move tip to the Task 2 start location (Fig. C.5), keep tip 1mm below recorded tip offset value (program moves it up 1)

6. *Initial Attempt*

   (a) 'Run Program'

   (b) Move 1mm in the positive X direction

   (c) Unclick 'Run Program'

   (d) Observe to ensure appropriate behavior

7. *Routine for 21 Iterations:* Each iteration you will run one "normalizing" trial to reset the system because it will have set dormant for a few minutes. Then

you will run the actual trial where you will record the performance. In-between trials you will processing trial performance with an ILC algorithm to compute and updated input signal.

(a) 'Record'

(b) 'Run Program'

(c) Stop 'Record'

(d) Move 1.5mm in the positive X direction

(e) Unclick 'Run Program'

(f) The first few iterations you will have to manually clean the nozzle tip because ink will not stop flowing accurately. To do so:

   i. Move 20mm in the positive Z direction to clean it off while returning to the start

   ii. Move 20mm in the negative Z direction to reset to the correct operating height

(g) Label video 'It**x**' where **x** = 1:21

(h) Transfer video to laptop 'ILC' folder

(i) In Command Window: `[time,u,error,Q,RMS,max_error] = ILC_Implement_MI_TV(0.3, 5, 1, 1, 21, x);`

   i. Extrusion System 1: x=1

   ii. Extrusion System 2: x=2

(j) Syntax for input = (P, Qband, order, StartIt, EndIt, Material)

(k) Follow Matlab prompts

(l) If there is a spike in the diameter before ink flows, the start frame was chosen improperly. → Type '0' for not happy and follow directions

(m) When script is done, a file will appear in 'ILC' as 'UFileY' (Y=x+1)

(n) Transfer the file to the robot computer

(o) On robot computer in command window type '`load UFileY`', again Y is the iteration

(p) **When you run the test associated with '`load UFile2`' make sure to toggle on the 'ILC Input' in the bottom left corner of the machine GUI, leave this toggled on for all iterations greater than 1**

(q) Change the ILC_q in the Machine GUI to the appropriate '`Uodd`' or '`Ueven`'

(r) *Repeat Iteration Routine until all 21 iterations have been run*

If you make a syntax error or you want to redo an iteration:

- Enter the `ILC_Implement_MI_TV` command with the 'StartIt' changed to where you want to start at. eg: UFile11; It11; StartIt = 11

You have completed Task 2

C.8 Task 3: Application of basis signals to scaffold manufacture

You have identified an input signal with ILC that best controls the manufacture of a thin cylinder of material. ***For multi-material structures, you will have to perform Tasks 1 and 2 for both materials.*** Now we must process this information so it can be used for more advanced architecture structures. Task 3 is given for a single material structure; if you are fabricating a multi-material structure you must insert the proper highlighted steps.

C.8.1 Single Material Structures

1. Type in the following statement in the command window: `[ustart,usteady,uend,udecel,UComp]=usplice_complete(u(21,:), 1.2,3,10.5,14.2,6.5,7.5,x);`

   (a) **Extrusion System 1**: `x`=1

   (b) **Extrusion System 2**: `x`=2

2. Review Figure 300; the breakup of different basis task should correlate well with changes in magnitude of the input signal. You can adjust the bounds of the basis tasks by changing the numbers in the argument line; consult the syntax information contained in the file. Also, if you would like to use an input signal besides the one from iteration 21, change `u(21,:)` to `u(j,:)` where `j` is the desired iteration.

3. `usteady` should be near the nominal value

   (a) Nominal value = $1.1 \times \frac{\pi}{4} D^2$

   (b) D = Tip diameter = 0.51

   (c) 1.1 = correction factor to make manufacture more robust

4. `UComp` is the important section of data; UComp will have the structure:

$$\text{UComp}_{m\times 16} = \left[\begin{array}{cccc|c|c|c} Start_1 & S-S_1 & Stop_1 & Cor._1 & \text{Ext. 2} & \text{Ext. 3} & \text{Ext. 4} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{array}\right] \tag{C.3}$$

   where Ext. 2-4 will have the same structure as Ext. 1, m is the row size of the matrix, which will be the length of the longest task plus 2, and the `UComp` will always be 16 columns wide.

5. In the Matlab command window type:

   (a) `Save('Name','UComp','-v4')`

      i. '`Name`' = whatever you want

      ii. '`-v4`' = the version of Matlab it needs to be saved as; the robot computer runs an old version of Matlab

6. ** For multi-material structure, insert Multi-Material Structures Part A here**

7. Move to the robot computer

8. On robot computer:

   (a) load '`Name`'

   (b) run the program that builds a Position Vector V for your advanced architecture part. *Note: Designing this program is not a trivial task. Consult Sam (spolak2@illinois.edu) or Dave (hoelzle2@gmail.com) for instructions.*

   (c) Update Position Vector and ILC_q in the Path Generator to `V` and `UComp`, respectively. ** You will probably have to Stop the robot to update these databases, use the tape trick**

   (d) ** For multi-material structure, insert Multi-Material Structures Part B here**

   (e) Toggle 'Corner On' and 'Jumping On'

   (f) Move to the cleaning station (Fig. C.5) and toggle 'Cleaning On'

   (g) Move to desired starting location

   (h) Click 'Run Program'

C.8.2 Multi-Material Structures

Multi-Material Structures Part A

`usplice_complete.m` only catalogs a basis signal library from one basis signal at a time. To compile a multi-material basis signal library, you have to manually manipulate the data. Type in the following lines for a 2 material system:

1. `[ustart1,usteady1,uend1,udecel1,UComp1]=usplice_complete(u1(21,:), 1.2,3,10.5,14.2,6.5,7.5,1);`
2. `[ustart2,usteady2,uend2,udecel2,UComp2]=usplice_complete(u2(21,:), 1.2,3,10.5,14.2,6.5,7.5,2);` where `u1` and `u2` are names of your input signals for Extrusion Systems 1 and 2, respectively.
3. `UComp = [UComp1(:,1:4), UComp2(:,5:16)];`

*** Note: usplice_complete.m can easily be modified to perform this manual manipulation. Consult Dave (hoelzle2@gmail.com) if you want to attempt to automate this step.*

Multi-Material Structures Part B

You have to calibrate the relative locations of the nozzles. Extrusion System 1 is always the reference nozzle; note that all Z height and cleaning station locations should be referenced from Extrusion System 1. Below is the calibration routine (expand for systems with more than 2 materials):

1. Ensure that all Indexing Calibration values are 0 and click 'Update'
2. Move to the Calibration Location (Fig. C.5)

3. You will be filling out the following table; record in lab notebook:

$$\begin{bmatrix} \text{Level Set} & \text{Ext. 1} & \text{Ext. 2} & \text{Diff.} & \text{Ext. 2, check} \\ (y_{pos}, z_{pos}) & x_1 & x_2 & x_2 - x_1 & x_{2,\text{check}} \\ (x_{pos}, z_{pos}) & y_1 & y_2 & y_2 - y_1 & y_{2,\text{check}} \\ (x_{pos}, y_{pos}) & z_1 & z_2 & z_2 - z_1 & z_{2,\text{check}} \end{bmatrix} \tag{C.4}$$

   (a) Move the nozzle of Extrusion System 1 in the negative X direction until it just touches the side of the oil bath (should be at 0.01 mm precision)

   (b) Record this position into x_1 and also record the Y and Z position into y_{pos} and z_{pos}, respectively.

   (c) Do the same for the Y and Z axes; record into y_1 and z_1

   (d) Click 'Extruder 2' to activate Extrusion System 2

   (e) Run the same procedure with extrusion system 2 using the same level set values as previous, record x_2, y_2, and z_2.

   (f) Calculate the Diff. column

4. Input the Diff. column into the Indexing Calibration portion of the GUI, Ext. 2 X, Y, and Z.

5. Click 'Update'

6. Click 'Extruder 1', allow the switch, then click 'Extruder 2'; the calibration values are now reflected in the new extrusion system position

7. Check that the calibration is accurate. Run the above routine with extrusion system 2. New values should be less than or equal to 0.02 mm from x_1, y_1, and z_1, respectively.

C.9 Task 4: Streamlined application of BTILC

Tasks 1-2 follow a linear progression of collecting data that is used for BTILC. Once these tasks have been completed for a given build material you do not need to restart with task 1. Instead, you can skip Task 1 and perform an abbreviated Task 2. Fig. C.1 demonstrates the streamlined workflow; this figure is a modification of Fig. 6.1 from Chapter 6.

1. Perform Task 0

2. Copy the `DataSave.mat` file to the directory you will be working from

3. Upload the `UFilex` file you want to start from into the robot computer, where `x` is the iteration number. A good choice is 17. You will also want to delete all `UFilex` files from the robot computer so you do not make a mistake when accessing data.

4. Enter the correct Position Vector (V1a or V1b) and the correct input signal (Uodd or Ueven) into the Path Generator.

5. Update `Video_Processing_CF.m and ILC_Implement_MI_TV.m` with the correct calibration values and system dynamics. See Tasks 1 and 2.

6. Also, change `Video_Processing_CF.m`: on line 95 change `if (Iteration == 1)` to `if (Iteration == x)` where x is the selected iteration number.

7. Run ILC: `[time,u,error,Q,RMS,max_error] = ILC_Implement_MI_TV(0.3, 5, 1, x, 21, y);` where `x` is the starting iteration and y is the extrusion system number.

8. Run this abbreviated version of Task 2 for each material required then proceed to Task 3.

Appendix D

Cross Coupled Iterative Learning Control of Systems with Dissimilar Dynamics: Design and Implementation

Appendix D is an exact recreation of reference [78]: K.L. Barton, D.J. Hoelzle, A.G. Alleyne, and A.J. Wagoner Johnson, "Cross Coupled Iterative Learning Control of Systems with Dissimilar Dynamics: Design and Implementation," *International Journal of Control*, vol. 84, no. 7, pp. 1223 - 1233, 2011.

Please note that this paper uses the conventional ILC notation, e.g. $e_j(k)$, as opposed to the modified notation used throughout this dissertation.

D.1 Introduction

Cross Coupled Control (CCC) has been applied to multi-axis systems in which there is a primary objective that defines manufacturing process performance. Individual axis performance is deemphasized in favor of a coupled axis, appropriately defined to measure the primary performance objective [79, 80]. The classic example of the CCC approach is a computer numerically controlled (CNC) robot where the primary objective is the dimensional accuracy of a manufactured part, not individual axis objectives. Performance is defined by a coupled axis, termed contour error, which is the normal distance from the prescribed trajectory and is a metric of the primary objec-

tive, i.e. dimensional accuracy. The redefinition of performance objectives developed in CCC has been integrated into the framework of Iterative Learning Control (ILC) by [38] to form Cross Coupled Iterative Learning Control (CCILC). ILC is a control algorithm that can be applied to systems that track a repeated trajectory [10]. The algorithm exploits trajectory repetition to improve reference tracking based off input and output information learned in previous iterations. By directly considering the primary objective and exploiting trajectory repetition, CCILC has been shown to achieve superior performance in comparison to CCC and individual axis ILC alone in contoured trajectory tracking problems [38, 81].

CCC and CCILC have been traditionally applied to planar manufacturing robots in which the X and Y axes have similar yet individual dynamics and are actuated and sensed by identical hardware. CCILC is a special form of a Multi-Input Multi-Output (MIMO) approach in which two Single-Input Single-Output (SISO) systems are coupled together through the output. This paper considers CCILC applied to a general set of systems, where the individual dynamics, as well as the actuation and sensing hardware, need not be common among the different systems. Previous CCC publications have alluded to potential problems when dissimilar systems are coupled [79, 80, 82]. The work in this paper builds off of the previous work of the authors [83], exploiting performance disparities between two subsystems that have a coupled objective in manufacturing robots. Given a system containing a fast subsystem and a slow subsystem, we apply a weighting filter that penalizes fast subsystem performance in the frequency ranges that are un-trackable by the slow subsystem. This filter shows up in the derivation of the contour error. The proposed controller framework enforces dynamics in the fast subsystem that compensate for inadequacies in the slow subsystem.

The main motivation for this work is manufacturing systems. Besides the example shown in this paper, performance limitations due to dynamical dissimilarities between

axes arises in other manufacturing systems where the individual control objectives can be easily handled in one subsystem, while the other subsystem is only capable of achieving poor performance results. One example would be robotic manipulators tooling parts on a conveyor line. Here, the agile robotic manipulator can easily compensate for the positioning of the low-bandwidth conveyor system; if the robotic manipulator knows the conveyor positioning error [84]. Outside of manufacturing, some other examples include chemical mixing [85], hybrid system applications [86], and multi-phase system applications such as heating and air conditioning systems [87].

The CCILC method presented here is applied to a micro-Robotic Deposition (μRD) manufacturing system, an Additive Manufacturing process in which a colloidal ink is extruded through a micro-sized nozzle while being positioned in space to fabricate three-dimensional structures [35]. The extrusion and positioning systems are drastically different, with extrusion system performance measured in volume and positioning system performance measured in distance. Additionally, the positioning system has a bandwidth that is over 100 times faster than the extrusion system.

The following sections establish the control problem and outline the solution and μRD implementation. The class of systems valid for this modification of CCILC is defined in Section D.2. Coupling of multiple systems is defined in general and for dissimilar systems in Section D.3. Section D.4 presents CCILC in the Norm Optimal framework. The μRD systems, particularly the two dissimilar axes of interest, and learning controller design are described in Section D.5. Experimental results are presented and discussed in Section D.6. Section D.7 summarizes the paper and provides concluding statements.

D.2 Class of Systems

In this paper we consider stable, linear time-invariant (LTI), causal, discrete-time MIMO systems, P, which perform the same task repetitively. P is given as

$$P \triangleq \begin{cases} x_j(k+1) = Ax_j(k) + Bu_j(k) \\ \quad \delta y_j(k) = Cx_j(k) + Du_j(k), \end{cases} \tag{D.1}$$

$$y_j(k) = \delta y_j(k) + y_o(k) + d_j(k) \tag{D.2}$$

where $k = 0, 1, \ldots, N-1$ is the discrete time index, $j = 0, 1, \ldots$ is the iteration index, $u_j(k) \in \mathbb{R}^{q_i}$ is the control, $y_j(k) \in \mathbb{R}^{q_o}$ is the output, $y_o(k) \in \mathbb{R}^{q_o}$ is iteration-invariant, $d_j(k) \in \mathbb{R}^{q_o}$ corresponds to stochastic (iteration-varying) external disturbances, $x_j(k) \in \mathbb{R}^n$ are system states, and (A, B, C, D) are appropriately sized real-valued matrices. It is assumed that $x_j(0) = x_o$ for all j, and note that $y_o(k)$ can be used to capture iteration-invariant initial conditions, feedback control, and external disturbances. In the lifted-domain [88, 89], the discrete-time behavior of the system is represented by its convolution matrix $\mathbf{P}$ using impulse response data $H_{m,n}(k)$, (D.3), where $\{m, n\}$ identify the indices for the impulse response data and range from $0, \ldots, N-1$. Note that a bold variable is used to denote a lifted system description.

$$\mathbf{P} = \begin{bmatrix} H_{0,0} & & 0 \\ \vdots & \ddots & \\ H_{N-1,0} & \cdots & H_{N-1,N-1} \end{bmatrix}. \tag{D.3}$$

For MIMO LTI systems, $H_{m,n}(k)$ contains the impulse response from each of the

q_i inputs to each of the q_o outputs and can be derived using the matrices in (D.1),

$$H_{m,n} : \begin{cases} D, & m = n \\ CA^{m-n-1}B, & m > n. \end{cases} \tag{D.4}$$

Given $H_{m,n}(k) \in \mathbb{R}^{q_o \times q_i}$, system $\mathbf{P} \in \mathbb{R}^{Nq_o \times Nq_i}$ is a lower triangular matrix with a block Toeplitz structure. While the results presented in this paper are for an LTI system, the same design process can be applied to LTV systems. In the case of LTV systems, $H_{m,n}$ is of the form,

$$H_{m,n} : \begin{cases} D(m), & m = n \\ C(m)A(m-1)A(m-2)\dots A(n+1)B(n), & m > n. \end{cases} \tag{D.5}$$

During trial j, system $\mathbf{P}$ maps the input signal u_j to the measured output signal y_j, i.e., $\mathbf{y}_j = \mathbf{P}\mathbf{u}_j$, with $\mathbf{u}_j$ and $\mathbf{y}_j$ defined in (D.6) and (D.7), respectively.

$$\mathbf{u}_j = \begin{bmatrix} u_j^T(0) & u_j^T(1) & \cdots & u_j^T(N-1) \end{bmatrix}^T \tag{D.6}$$

$$\mathbf{y}_j = \begin{bmatrix} y_j^T(0) & y_j^T(1) & \cdots & y_j^T(N-1) \end{bmatrix}^T \tag{D.7}$$

$$\text{with} \quad u_j^T(k) = \begin{bmatrix} u_j^1(k) & \cdots & u_j^{q_i}(k) \end{bmatrix}$$

$$\text{and} \quad y_j^T(k) = \begin{bmatrix} y_j^1(k) & \cdots & y_j^{q_o}(k) \end{bmatrix}$$

In this paper we adopt a widely used norm optimal ILC update law [89, 90]

$$\mathbf{u}_{j+1} = \mathbf{L}_\mathbf{u}\mathbf{u}_j + \mathbf{L}_\mathbf{e}\mathbf{e}_j \tag{D.8}$$

where the error signal is comprised of the individual error signals from each independent axis as shown in (D.9).

$$\begin{aligned} \mathbf{e}_j &= \begin{bmatrix} e_j^T(0) & e_j^T(1) & \cdots & e_j^T(N-1) \end{bmatrix}^T \\ e_j^T(k) &= \begin{bmatrix} e_j^1(k) & \cdots & e_j^{q_o}(k) \end{bmatrix} \end{aligned} \tag{D.9}$$

The error signal in (D.9) is defined as $\mathbf{e}_j = \mathbf{y}_r - \mathbf{y}_j$, where $\mathbf{y}_r$ is the reference signal and is assumed iteration invariant. In (D.8), $\mathbf{L_u}$ and $\mathbf{L_e}$ are solutions to a quadratic optimization problem detailed in Section D.4. These lifted matrices are generally non-causal, time-invariant linear operators on the control and error signals, respectively.

Previous work in [91] introduced time-varying designs for these filters to address particular challenges at specific time intervals. The objective of this work is to implement a time-varying ILC design which couples the output performance of two dissimilar systems in the norm optimal framework. The coupling of multiple dissimilar systems in the form of the output performance is presented in the following section.

D.3 Coupling of Multiple Dissimilar Systems

When combining multiple individual systems or axes, one may couple these axes through a common desired output. For MIMO systems which consist of two or more individual axes, an additional error signal known as the contour error can be defined, as illustrated by the 2D example in Fig. D.1. Contour errors, ε, for a general class of MIMO systems can be defined with respect to the individual error signals, $e_1, e_2, \ldots, e_{q_o}$, and trajectory dependent gains known as coupling gains [92,93], $c_1(k,\theta), c_2(k,\theta), \ldots, c_{q_o}(k,\theta)$, where k is the time interval from $k = 0, 1, \ldots, N-1$,

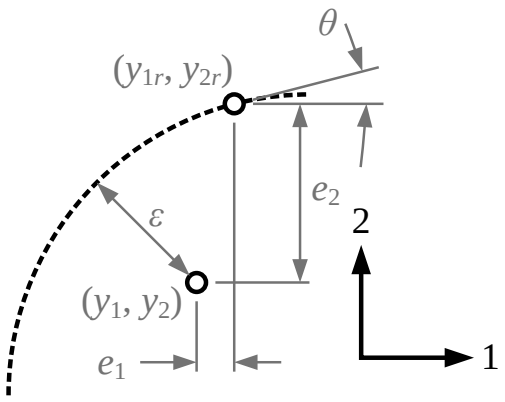


Figure D.1: 2D trajectory illustrating contour (ε) and individual errors (e_1, e_2) for two individual axes. These errors are defined with respect to the desired position (y_{1r}, y_{2r}) and the actual position (y_1, y_2) of a system in the ($\mathbf{Axis-1}, \mathbf{Axis-2}$) coordinate frame. Linearized coupling gains ($c_1(k,\theta), c_2(k,\theta)$) at point in time (k) with respect to the tangent angle (θ) can be used to simplify the derivation of the contour error.

θ is defined as the instantaneous angle of the reference trajectory with respect to the horizontal axis of the coordinate system [92], and $1, 2, \ldots, q_o$ are the individual outputs.

When the class of MIMO systems described in Section D.2 is comprised of dissimilar axes, an additional weighting component should be added to the definition of the contour error to account for variations between the individual systems such as time-constants, system resonances, and system bandwidths. Previous work in [83] presented a coupled learning controller which incorporated an additional weighting gain into the derivation of the contour error in order to compensate for dominant time constant dissimilarities between two systems. This gain was applied across all frequencies, thereby indiscriminately increasing the weighting applied to the error signals at all frequencies. In this paper, we extend the idea of additional weighting through the introduction of a weighting filter. The weighting filter is used to compensate for dynamic inconsistencies across a range of frequencies when combining dynamically diverse systems. The weighting filters are derived from the relationship between the fast and slow systems in order to compensate for specific dynamic

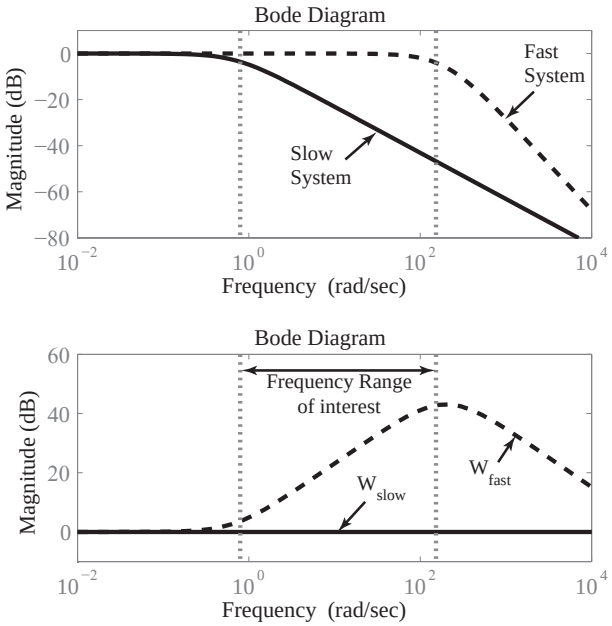


Figure D.2: Subplot 1: Example complementary sensitivity plot for a dynamically slow and fast system. Note the frequencies of interest lie between the cutoff frequencies of the two systems, respectively. In this frequency range the fast system can compensate for performance limitations from the slower system. Subplot 2: Weighting filters designed using the fast and slow system from subplot 1. Note that W_{fast} is calculated by dividing the complementary sensitivity of the fast system by the complementary sensitivity of the slow system. W_{slow} is set to one to reflect no additional weighting on the slow system.

differences between the two systems.

Consider the complementary sensitivity plots of two dynamically diverse stable systems, subplot 1 of Fig. D.2. Both systems can easily handle low frequency signals, while the high frequency response following the cutoff frequency of the faster system is unimportant since it can be categorized as either unattainable reference trajectories or noise. The frequency range of interest lies *between* the cutoff frequencies of the two systems. In this range, the low bandwidth of the slow system can be compensated for by the additional tracking capabilities of the fast system by coupling the two systems through the contour error. An important criteria for coupling dynamically diverse systems is signal equivalence. This requires that the signals are modified in

order to balance the dynamics between the two systems. While a bandpass filter may target the frequency range of interest, the amplitude and shape of the filter should compensate for dynamic differences between the two systems. A more direct method for designing an appropriate weighting filter comes from comparing the dynamics of the two systems directly.

Dividing the complementary sensitivity of the fast system by the complementary sensitivity of the slow system results in a filter that maintains low frequency performance, amplifies signals in the frequency range of interest, and minimizes high frequency signals, as illustrated in subplot 2 of Fig. D.2. Amplifying the errors in this frequency range forces the faster system to respond, thereby relinquishing some of the performance strain from the slower system. Note that care must be taken to ensure minimal amplification of the high frequency signals. A low-pass filter may be added to the ratio of complementary sensitivities to ensure the filter attenuates high frequency noise. A general definition of the weighting filters for two SISO systems is provided in (D.10), where T represents the complimentary sensitivity of a system. A typical low-pass filter is provided in (D.11).

$$W_{fast} = \frac{T_{fast}}{T_{slow}} \cdot F_{lowpass}, W_{slow} = 1. \tag{D.10}$$

$$F_{lowpass} = \frac{k}{z - \alpha_1}. \tag{D.11}$$

Mathematically, for the two axes, represented as individual systems in Fig. D.3, the modified contour error can be defined as,

$$\varepsilon(k) = W_{fast}(\mathsf{q}) \cdot c_1(k, \theta) \cdot e_1(k) + W_{slow}(\mathsf{q}) \cdot c_2(k, \theta) \cdot e_2(k) \tag{D.12}$$

$$\varepsilon(k) = C_Q(\mathsf{q}, k, \theta) \cdot e(k), \tag{D.13}$$

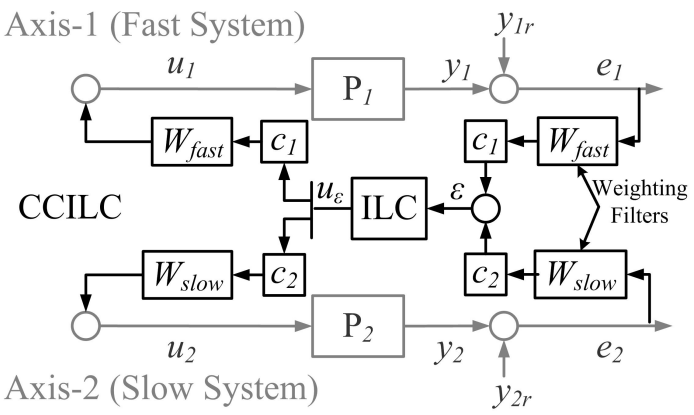


Figure D.3: Block diagram of a two-axis MIMO system in which the two independent SISO axes are coupled together via CCILC. $P_{(\cdot)}$ represents the plant sensitivity function defined as $\frac{G_{(\cdot)}}{1+G_{(\cdot)}k_{(\cdot)}}$, where $G_{(\cdot)}$ is the open-loop axis model and $k_{(\cdot)}$ is a feedback controller used to stabilize the open-loop axis. Note that $P_{(\cdot)}$ is different from the complementary sensitivity function, $T_{(\cdot)}$, defined as $\frac{G_{(\cdot)}\cdot k_{(\cdot)}}{1+G_{(\cdot)}k_{(\cdot)}}$. In this example, the fast and slow system descriptions are associated with Axis-1 and Axis-2, respectively.

where q is the backwards time shift operator defined as $\mathsf{q}y(k) \equiv y(k-1)$ and $W_{fast}(\mathsf{q})$ and $W_{slow}(\mathsf{q})$ are the filters given in (D.10). Note that Axis-1 is assumed to be the fast system, while Axis-2 is defined as the slow system, respectively. Equation (D.13) illustrates that the weighting filters and coupling gains are combined into a single variable, $C_Q(\mathsf{q}, k, \theta)$. Linearized coupling gains $c_1(k, \theta)$ and $c_2(k, \theta)$ have the following format

$$c_1(k, \theta) = -\sin\theta(k); c_2(k, \theta) = \cos\theta(k), \tag{D.14}$$

Note that the use of trajectory-dependent coupling gains leads to a time-varying controller. Figure D.3 provides a block diagram representation of two individual axes coupled together through CCILC.

The generalized structure for the norm optimal controller is given in the following section.

D.4 Norm Optimal ILC

The norm optimal algorithm is designed to minimize a quadratic optimization problem [94–96],

$$\mathcal{J} = \mathbf{e}_{j+1}^T \mathbf{Q} \mathbf{e}_{j+1} + \mathbf{u}_{j+1}^T \mathbf{S} \mathbf{u}_{j+1} + (\mathbf{u}_{j+1} - \mathbf{u}_j)^T \mathbf{R} (\mathbf{u}_{j+1} - \mathbf{u}_j). \tag{D.15}$$

where $(\mathbf{Q}, \mathbf{R}, \mathbf{S})$ are symmetric, positive definite real-valued matrices of appropriate dimension and $\mathbf{P}^T\mathbf{Q}\mathbf{P} + \mathbf{S} + \mathbf{R}$ is positive definite. Applying the substitution $\mathbf{e}_{j+1} = \mathbf{e}_j - \mathbf{P}(\mathbf{u}_{j+1} - \mathbf{u}_j)$, differentiating $\mathcal{J}$ with respect to $\mathbf{u}_{j+1}$, setting the result to zero, and rearranging the solution, yields the general norm optimal controller,

$$\begin{aligned} \mathbf{u}_{j+1} &= \mathbf{L}_\mathbf{u} \mathbf{u}_j + \mathbf{L}_\mathbf{e} \mathbf{e}_j \\ \mathbf{L}_\mathbf{u} &= (\mathbf{P}^T\mathbf{Q}\mathbf{P} + \mathbf{S} + \mathbf{R})^{-1}(\mathbf{P}^T\mathbf{Q}\mathbf{P} + \mathbf{R}) \\ \mathbf{L}_\mathbf{e} &= (\mathbf{P}^T\mathbf{Q}\mathbf{P} + \mathbf{S} + \mathbf{R})^{-1}\mathbf{P}^T\mathbf{Q}. \end{aligned} \tag{D.16}$$

For many designs, $(\mathbf{Q}, \mathbf{R}, \mathbf{S}) \triangleq (q\mathbf{I}, s\mathbf{I}, r\mathbf{I})$, with q, s, r real-valued positive scalars. In [91], a novel time-varying design for the $\mathbf{Q}$ weighting matrix was introduced,

$$\mathbf{Q}^{tv} = \mathbf{\Sigma}_Q \cdot [\mathbf{\Gamma 1}_Q + \mathbf{\Gamma 2}_Q \cdot \mathbf{C}_Q^T \mathbf{C}_Q] \tag{D.17}$$

where the $\mathbf{C}_Q$ matrix contains the terms used to define coupling between the individual error signals of the MIMO system, $\mathbf{\Gamma 1}_Q$ and $\mathbf{\Gamma 2}_Q$ refer to the weighting matrices applied to the coupled or individual error signals, and $\mathbf{\Sigma}_Q$ determines the overall weighting on the error signal compared to the control and change in control signals.

The coupling matrix $\mathbf{C}_Q$ is derived from the definition of the contour error given in Eq. (D.13). Applying the lifted approach to Eq. (D.13) and writing the term

$\mathbf{C}_Q$ as the lifted form of $C_Q(\mathsf{q}, k, \theta)$, the coupling of the error terms is represented by the convolution matrix $\mathbf{C}_Q$ using the combined impulse response data of the weighting filters, $\{W_{fast}, W_{slow}\}$, and the coupling gains, $\{c_1, c_2\}$ for the two SISO system example in Fig. D.3, respectively.

$$\mathbf{C}_Q = \begin{bmatrix} C_{0,0} & & 0 \\ \vdots & \ddots & \\ C_{N-1,0} & \cdots & C_{N-1,N-1} \end{bmatrix}. \tag{D.18}$$

For MIMO systems comprised of two dynamically different SISO systems, $C_{m,n}$ contains the impulse response data of the weighting filters combined with the coupling gains in a two element vector format. Define W_{fast} and W_{slow} with the real-valued matrices, $\{A_{W_{fast}}, B_{W_{fast}}, C_{W_{fast}}, D_{W_{fast}}\}$ and $\{A_{W_{slow}}, B_{W_{slow}}, C_{W_{slow}}, D_{W_{slow}}\}$, respectively. Using these matrices, along with the vector descriptions of the coupling gains, $\mathbf{c_1} = [c_1(0,\theta) \cdots c_1(N-1,\theta)]$ and $\mathbf{c_2} = [c_2(0,\theta) \cdots c_2(N-1,\theta)]$, $C_{m,n}$ can be defined as,

$$C_{m,n} : \begin{cases} \begin{bmatrix} D_{W_{fast}}\mathbf{c_1}_m & D_{W_{slow}}\mathbf{c_2}_m \end{bmatrix}, & m = n \\ \begin{bmatrix} C_{W_{fast}} A_{W_{fast}}^{m-n-1} B_{W_{fast}} \mathbf{c_1}_n & C_{W_{slow}} A_{W_{slow}}^{m-n-1} B_{W_{slow}} \mathbf{c_2}_n \end{bmatrix}, & m > n. \end{cases} \tag{D.19}$$

The matrices $\mathbf{\Gamma 1}_Q$ and $\mathbf{\Gamma 2}_Q$ refer to the amount of weighting applied to the coupled or individual signals, respectively. These matrices are of the form provided in (D.20) and (D.21), where the inner block diagonal matrices are shown for a 2 DOF system.

$$\boldsymbol{\Gamma 1}_Q = \begin{bmatrix} \begin{bmatrix} \gamma(1) & 0 \\ 0 & \gamma(1) \end{bmatrix} & & 0 \\ & \ddots & \\ 0 & & \begin{bmatrix} \gamma(N) & 0 \\ 0 & \gamma(N) \end{bmatrix} \end{bmatrix}, \tag{D.20}$$

$$\boldsymbol{\Gamma 2}_Q = \begin{bmatrix} \begin{bmatrix} 1-\gamma(1) & 0 \\ 0 & 1-\gamma(1) \end{bmatrix} & & 0 \\ & \ddots & \\ 0 & & \begin{bmatrix} 1-\gamma(N) & 0 \\ 0 & 1-\gamma(N) \end{bmatrix} \end{bmatrix}. \tag{D.21}$$

As can be seen from (D.20) and (D.21), the individual elements in $\boldsymbol{\Gamma 1}_Q$ and $\boldsymbol{\Gamma 2}_Q$ are related. Selecting ($\gamma(k) = 1$) refers to all of the weighting being applied to the individual signals (nominal ILC design), while ($\gamma(k) = 0$) results in only the coupled signals being weighted (CCILC design).

The gain matrix $\boldsymbol{\Sigma}_Q$ determines the overall weighting on the error signals with respect to the control signals and change in control signals and is of the form shown in (D.22). Note that the inner diagonal matrix is illustrated for a 2 DOF system.

$$\boldsymbol{\Sigma}_Q = \begin{bmatrix} \begin{bmatrix} \sigma_Q(1) & 0 \\ 0 & \sigma_Q(1) \end{bmatrix} & & 0 \\ & \ddots & \\ 0 & & \begin{bmatrix} \sigma_Q(N) & 0 \\ 0 & \sigma_Q(N) \end{bmatrix} \end{bmatrix}. \tag{D.22}$$

Recall from Section D.3 that the coupling gains are derived from the desired output trajectory, while the weighting filters are designed to compensate for dynamic differences between the axes. Using Eq. (D.17) and the more traditional format for $\mathbf{S}$ and $\mathbf{R}$, $(\mathbf{S}, \mathbf{R}) \triangleq (s\mathbf{I}, r\mathbf{I})$, a modified cost function can be determined.

$$\mathcal{J} = \mathbf{e}_{j+1}^T \mathbf{Q}^{tv} \mathbf{e}_{j+1} + \mathbf{u}_{j+1}^T \mathbf{S} \mathbf{u}_{j+1} + (\mathbf{u}_{j+1} - \mathbf{u}_j)^T \mathbf{R} (\mathbf{u}_{j+1} - \mathbf{u}_j) \tag{D.23}$$

An essential part of the design process involves determining weighting matrices for the cost function in Eq. (D.23). References [81, 91] present some guidelines for designing and tuning the matrices based on performance and robustness requirements. The work in this paper focuses on time-variation in the $\mathbf{Q}$ matrix primarily due to a coupled output objective defined as the contour error (D.13). For further examples of systems which implement time-varying weighting matrices see [91].

D.5 System Setup

In the next two sections we institute a change of variables where Axis-1 is a y-axis positioning system and Axis-2 is an extrusion system.

In order to explore the performance benefits of combining two dissimilar SISO sys-

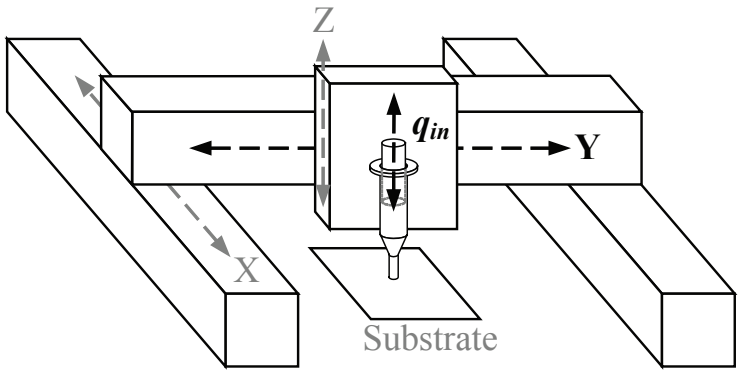


Figure D.4: Multi-axis robotic testbed with extrusion system included. Note that the example used in this paper only couples the extrusion system and the y-axis.

tems or axes into a MIMO format, time-varying (CCILC) and time-invariant (Nominal ILC) norm optimal learning controllers are implemented on the y-axis positioning and extrusion systems of a μRD system, Fig. D.4 and Fig. D.5 respectively. The primary objective in μRD is dimensional accuracy of the extruded build material. The extrusion and positioning systems are drastically different, with extrusion system performance measured in volume and positioning system performance measured in distance. The positioning system has a bandwidth that is more than 100 times faster than the extrusion system. Here, we show that the proposed control method exploits the disparity in axes performance, incongruently penalizing the fast positioning axis error in certain frequencies.

The input for the y-axis is amplifier current and the output is axis position, y_{out}. The input to the extrusion system is plunger displacement rate, q_{in}, and the output is build material volumetric flowrate, q_{out}.

Dynamic models of the two axes were developed in [36] and [55]. Numerical values for the y-axis plant model, G_y in Eq. (D.24), along with a stabilizing feedback controller, k_{Gy} in Eq. (D.25), can be found in the Appendix. The extrusion system, P_q presented in Eq. (D.26), is open-loop stable and therefore only requires open-loop input signals.

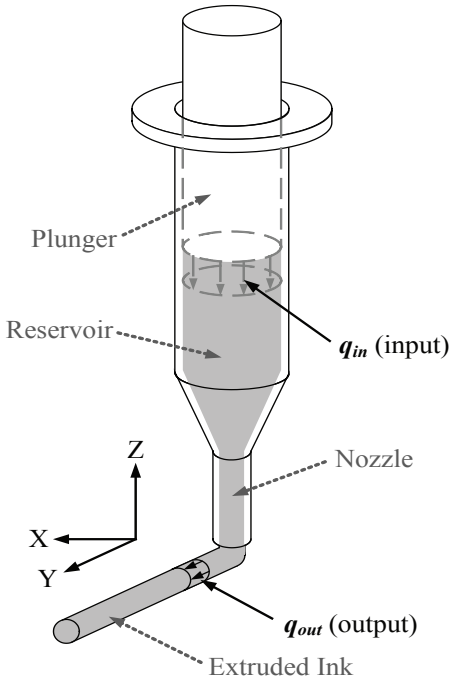


Figure D.5: Extrusion system for material deposition

$$G_y(z) = \frac{K(z+\alpha_1)(z^2-\alpha_2 z+\alpha_3)(z^2-\alpha_4 z+\alpha_5)}{(z-\beta_1)(z-1)(z^2-\beta_2 z+\beta_3)(z^2-\beta_4 z+\beta_5)}. \tag{D.24}$$

$$k_{Gy}(z) = \frac{K(z-\alpha_1)(z-\alpha_2)(z-\alpha_3)}{(z-\beta_1)(z-\beta_2)(z-\beta_3)}. \tag{D.25}$$

$$P_q(z) = \frac{0.00019766}{z-0.9998} \tag{D.26}$$

The MIMO system is subject to a combined trajectory which integrates material extrusion with linear stage positioning. Explicitly stated, the y-axis proceeds at a constant velocity while the extrusion system has a pulsed trajectory. The combination of these reference trajectories corresponds to the extrusion of a long cylinder of material deposited on a flat substrate, Fig. D.6. The primary objective is to achieve sharp and accurately placed transitions from no flowrate to a nominal flowrate with

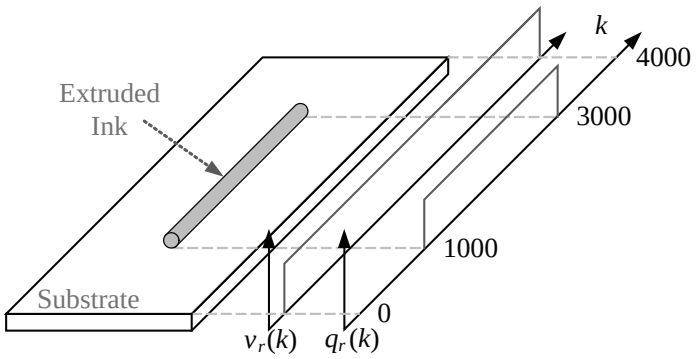


Figure D.6: Diagram of the desired fabricated structure and the corresponding reference trajectories. Reference trajectories for the two axes are the desired flowrate, q_r, and desired y-axis position, y_r. Position reference is shown in terms of axis velocity, $v_r(k) = (y_r(k) - y_r(k-1))/0.001$, where 0.001 is the sample time.

consistent nominal flowrate regulation.

Controller Design

The objective of this work is to pursue a primary performance objective by coupling two dissimilar axes through the desired output. The coupling of the output signals translates to a coupling of the error signals, as illustrated in the cost function of Eq. (D.23). The coupling between the signals results from the combination of coupling gains, $(c_y(k,\theta), c_q(k,\theta))$, and weighting filters, (W_{fast}, W_{slow}) in Fig. D.3. The coupling gains are derived from the reference trajectory (Fig. D.6) using the definition provided in Eq. (D.14). The trajectories in Fig. D.6 present an interesting challenge that we have addressed. Here, $c_q(k,\theta) = 1$ and $c_y(k,\theta) = 0$ for all $k = 0, 1, ...N-1$ with the exception of the start ($k = 1000$) and stop ($k = 3000$) locations of the q_r desired flowrate; this correlates to a single sample number for each location. In order to force additional compensatory movement from the y-axis, c_y must have a nonzero value for longer than one sample point. Therefore, the lifted time vectors of the coupling gains are filtered using a Gaussian filter with a bandwidth of 3 Hz. The resulting vectors are illustrated in Fig. D.7. This unique problem will be seen with

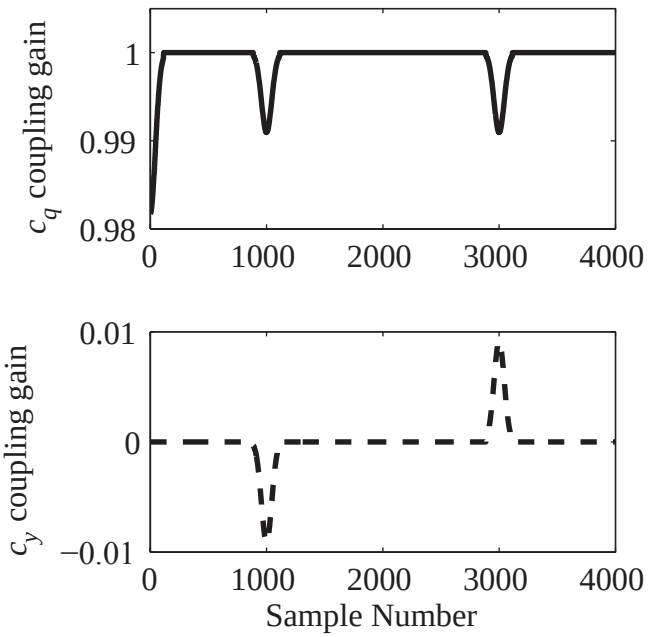


Figure D.7: Coupling gains used in the derivation of the contour error. Note that the vectors have been filtered using a Gaussian filter with a 3 Hz bandwidth in order to ensure the ability to force compensatory action from the y-axis.

any reference trajectory that contains discontinuities.

For simplicity, the weighting filters have been redefined as $W_y \triangleq W_{fast}$ and $W_q \triangleq W_{slow}$, respectively. The weighting filters, W_y and W_q (Fig. D.8), for this example are of the form,

$$W_y = \frac{T_y}{P_q} \cdot F_{lowpass}; W_q = 1, \tag{D.27}$$

$$\text{with} \qquad F_{lowpass} = \frac{0.3297}{z - 0.6703}, \tag{D.28}$$

where P_q is the complimentary sensitivity (open-loop stable model) of the slower extrusion axis and $T_y = \frac{G_y k_{Gy}}{1+G_y k_{Gy}}$ is the complimentary sensitivity for the faster y-positioning axis, respectively. Note that for this example, open-loop system stability results in P_q replacing T_{slow} in the calculation of W_y, (D.10). The ratio of the two

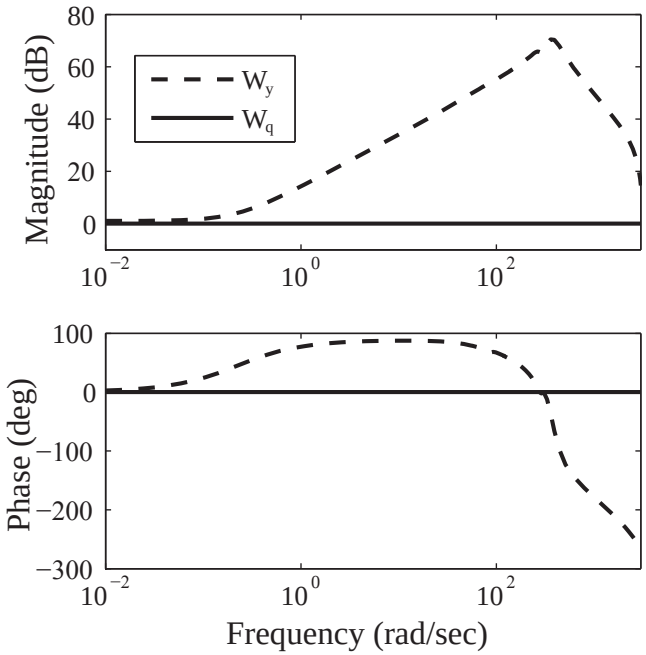


Figure D.8: Weighting filters used to compensate for the dissimilar dynamics between the positioning stage and the extrusion system. Note that only the fast system requires a weighting filter, W_y, as demonstrated by a weighting filter of 1 for the slow system, W_q.

models is used as a filter which weights the faster axis more heavily during the frequency range between the cutoff frequencies of the slow and fast axes. A lowpass filter combined with this ratio minimizes the amplification of any high frequency signals. The shape of this weighting filter forces the faster system to assume some of the performance load for the slower system, while maintaining robustness in the presence of high frequency noise or disturbances.

Learning filters of the form described in (D.16), with $\mathbf{Q}$ replaced by the time-varying weighting matrix of the form in (D.17), were designed using the methodology detailed in [91]. Heuristic tuning of the $\mathbf{S}$ and $\mathbf{R}$ weighting matrix gains resulted in the constant gain values ($s_y = 0.01, s_q = 0.001, r_y = 0.02, r_q = 0.01$) for the nominal ILC controllers and ($s_y = 0.01, s_q = 0.0005, r_y = 0.02, r_q = 0.01$) for the coupled CCILC controllers. The weighting gains for $\mathbf{Q}^{tv}$ are ($\gamma(k) = 1, 1 - \gamma(k) = 0, \sigma_Q(k) = 2$)

for nominal ILC (individual axis) control and $(\gamma(k) = 0, 1 - \gamma(k) = 1, \sigma_Q(k) = 2)$ for CCILC (coupled axis) control, for all $k = 0, 1, \ldots, N-1$. The values for $\sigma_Q(k)$ were chosen to maximize the system performance, while ensuring convergence and robustness.

Extrusion Trials

As a complement to the experimental data, each controller is evaluated visually in a manufacturing motivated example. Advanced architecture structures built by μRD require two materials to be directly opposed to each other to create distinct material, and hence physical property, domains. Here we evaluate the potential of Nominal ILC and CCILC to create these features by testing a butt-weld of two line segments five times for each controller; essentially two segments of the shape in Fig. D.6 oriented end-to-end. Ideally, each material transition should have minimal overlap so that the material property change is abrupt and should maintain a constant line width. This simple test uses one material and the evaluation metric is the regulation of line width. A multi-material example would simply require changing materials and identifying material specific Nominal ILC and CCILC signals for the y-axis and extrusion axis; displayed previously in the extrusion system axis in [97, 98].

D.6 Results

The generalized CCILC controller introduced in Section D.4 is applied to the μRD system. Figures D.9, D.10 and D.11 display signals for iteration 15. The performance of the combined system is hindered by the extrusion system performance, in which plunger displacement rate is limited to $\pm$ 20 mm^3/sec to minimize actuator wear. When applying ILC to the extrusion axis and y-axis independently, the coupled out-

put of the MIMO system poorly approximates the reference signal, Fig. D.11. Figure D.11 also shows the performance for the y-axis controlled by feedback and the extrusion axis in open-loop. Labeled as 'standard' in the figure, this control methodology is the common practice in μRD and yields poor flowrate modulation.

The CCILC approach penalizes the contribution of the y-axis to the contour error, yielding a coupled control signal that modifies the y-axis trajectory, Fig. D.9, to compensate for poor extrusion system performance. The feedforward input to the extrusion system remains relatively unchanged, Fig. D.10, thereby maintaining sub-threshold actuation inputs. Qualitatively, the y-axis briefly slows down and then accelerates into the desired position of the flowrate pulse and dwells momentarily to accumulate material volume, Fig. D.9. Then the y-axis, driven by the feedback controller, accelerates out of the dwell to minimize its individual axis tracking. This coupled axis behavior is intuitive in that it spatially positions the extrusion system in the correct location for the flowrate profile that is achievable by the extrusion system. Similar axes behavior has been designed on a similar system via *ad hoc* reference shaping [61]; however the method presented here achieves axes coordination automatically. The coordination of axes leads to a 14% average reduction in the root mean squared (RMS) tracking of the converged contour error, as compared to ILC applied to each system independently.

The primary objectives of the μRD process are the sharpness of the flowrate pulse and constant flowrate, which is analogous to accurate material starting and stopping, and constant diameter of the extruded material. Figure D.11 shows a contour tracking plot. Here CCILC yields the quickest transition from zero flowrate to a nominal flowrate and the longest duration of constant flowrate. The transition sharpness and increased duration of constant flowrate of the CCILC system are illustrated in the experimental images provided in the right-hand-side of Fig. D.11.

Lastly, CCILC and Nominal ILC are evaluated through a series of extrusion trials,

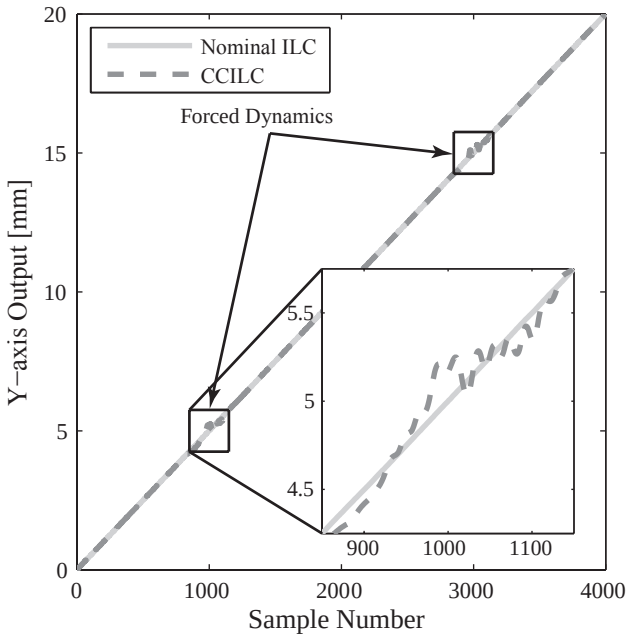


Figure D.9: Y-axis output for the Nominal ILC and CCILC cases. Axes coupling forces additional dynamics in the response to compensate for extrusion system inadequacies.

detailed in subsection D.5. The improved material flowrate transition sharpness seen in Fig. D.11 is realized in the ability to adjoin materials with minimal material overlap, Fig. D.12. For the ILC case, the inability to precisely transition from zero flowrate to the nominal flowrate leads to material overlap at the line segment abutment and therefore swelling and poor line width regulation. CCILC improves the flowrate performance and the two line segments can be placed neatly opposed to each other without material overlap. The implications of this improved CCILC performance are the ability to more closely achieve the ideal structure discussed in Section D.5; namely discrete divisions of materials in advanced architecture structures without a significant overlap section.

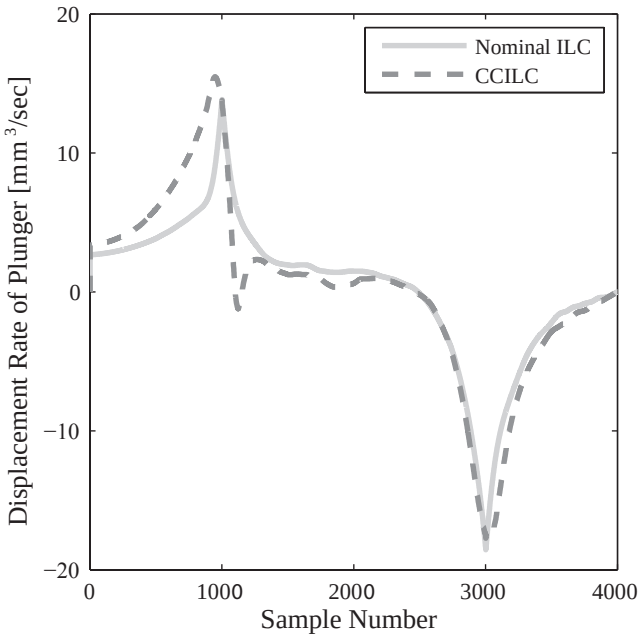


Figure D.10: Input signals for the extrusion system for the Nominal ILC and CCILC cases. Despite fairly consistent input signals that are within the input signal constraints, the CCILC controller results in much sharper material start and stop features as illustrated in Fig. D.11

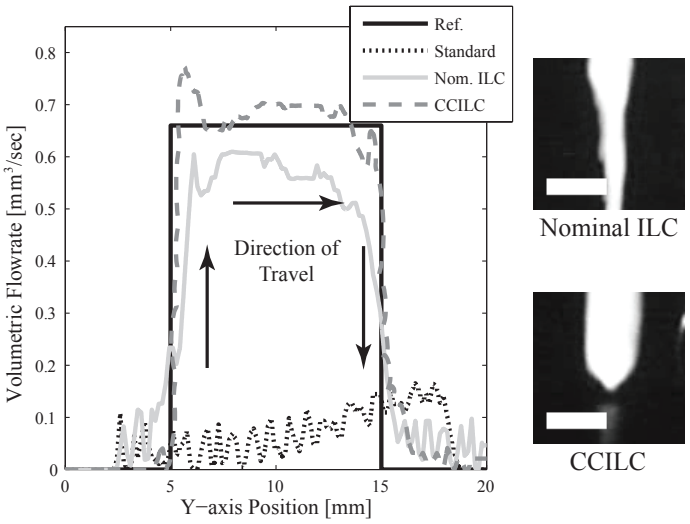


Figure D.11: Contour plot of the output signals for the μRD system, along with experimental images of the start positions (k=1000) for the Nominal ILC and CCILC cases. Note the improved sharpness of the deposited material for the CCILC case. The scale bar is 0.5 mm.

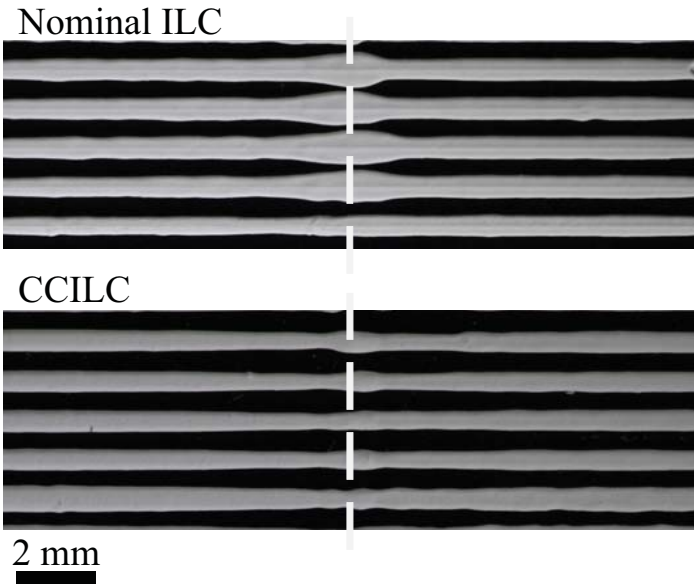


Figure D.12: Extrusion trials of the Nominal ILC and CCILC depositing two adjoined lines of material. Adjoining location denoted by white dashed line. CCILC minimizes material overlap by depositing a structure with sharp material starts and stops.

D.7 Concluding Remarks and Future Directions

In this paper, we investigate the coupling of dynamically dissimilar axes in manufacturing systems with a coupled primary objective. The key contributions in this work include 1) the introduction of a coupled output objective for dissimilar systems that incorporates the dynamic differences of the systems into the derivation of the desired objective, 2) the development of a novel framework for designing learning controllers which minimize a coupled objective for dissimilar systems, and 3) validation of this controller through experimental testing.

The traditional CCILC structure was adapted to include weighting filters that penalize contributions to a coupled objective, defined as the contour error, within a certain frequency range. This framework engages the underutilized high performance axis to assist low performance axes. In order to demonstrate the potential performance improvements obtained by coupling the output of the two dissimilar axes, a CCILC controller was applied experimentally to a μRD system. This MIMO

system consists of a positioning system and an extrusion system that is constrained by actuation limits. The generalized CCILC approach transfers actuation load from the extrusion system to the underutilized positioning system, thereby modifying the trajectory of the positioning system to compensate for extrusion system inadequacies. The experimental results display an average reduction of 14% in the RMS of the contour error using CCILC as compared to ILC designs. Extrusion trials illustrate that the additional material extrusion accuracy realized using CCILC leads to improved performance when joining multiple segments of material. Future work will investigate how well this improved extrusion capability translates to more advanced multi-material structures, as well as explore additional design approaches for optimizing the extrusion performance of the combined system.

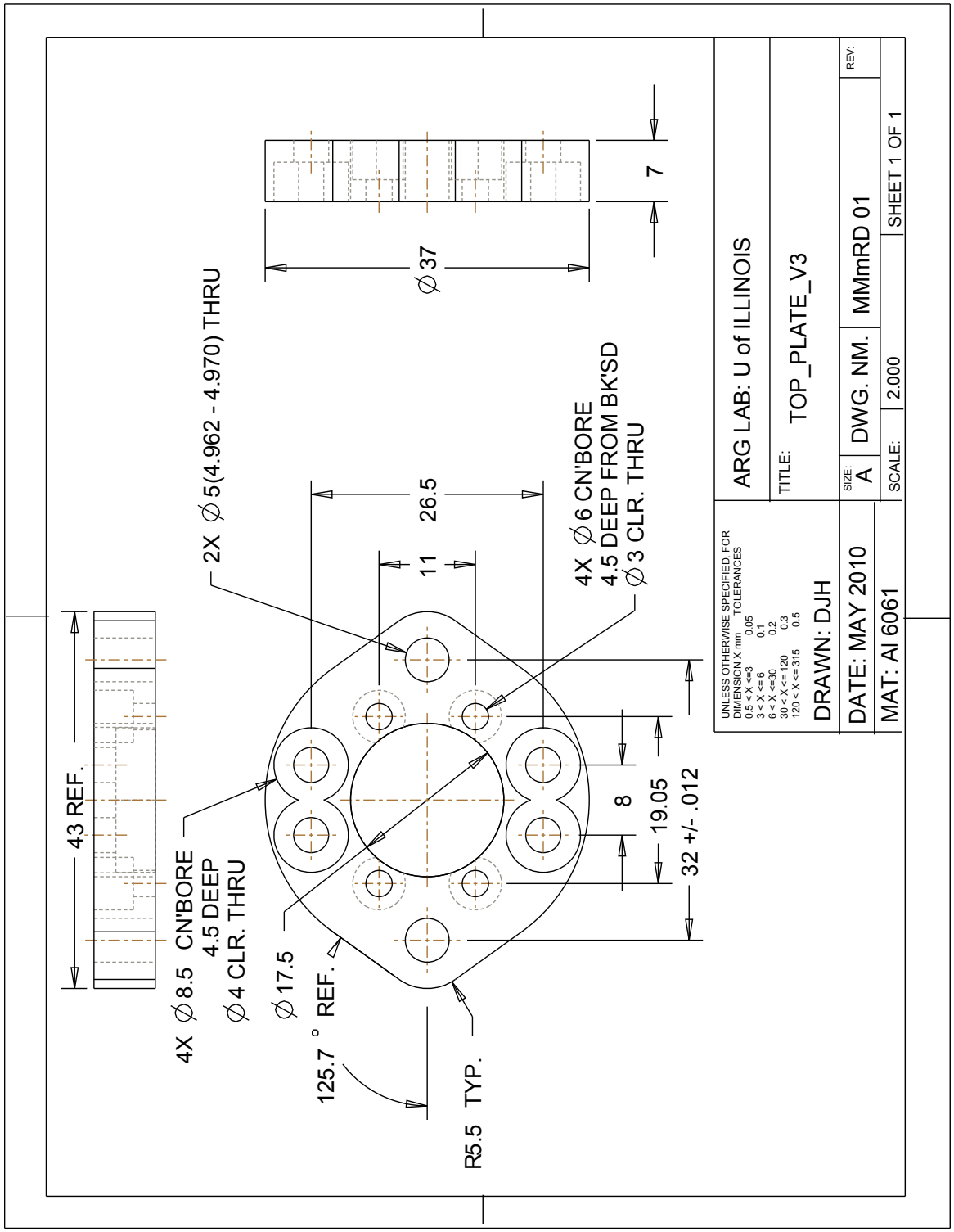
D.8 Acknowledgments

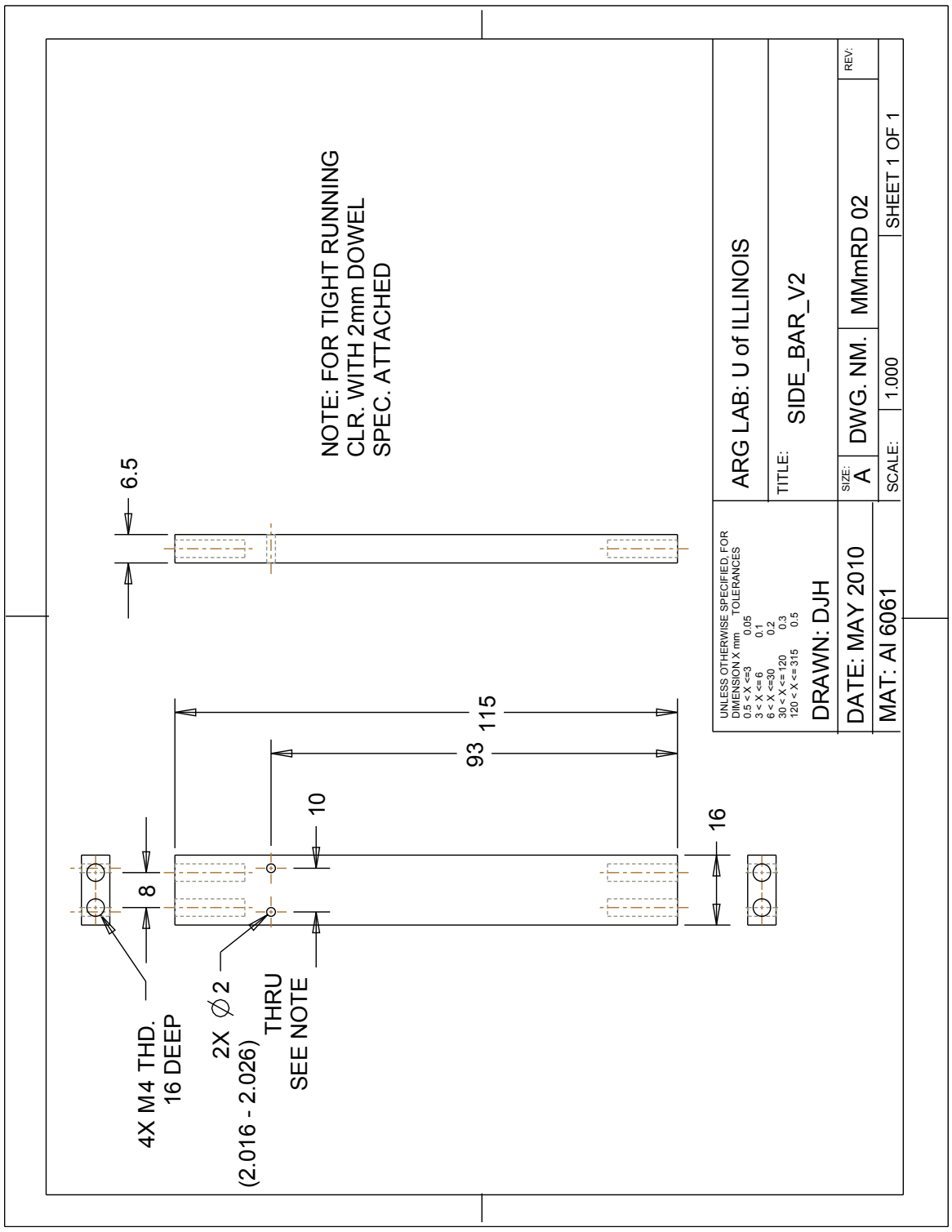
This work was supported by the National Science Foundation (NSF) Nano-CEMMS Center under Award DMI 0328162, CMMI 0749028, and NSF Award DMI 0140466, CMMI 09-00184 ARRA.

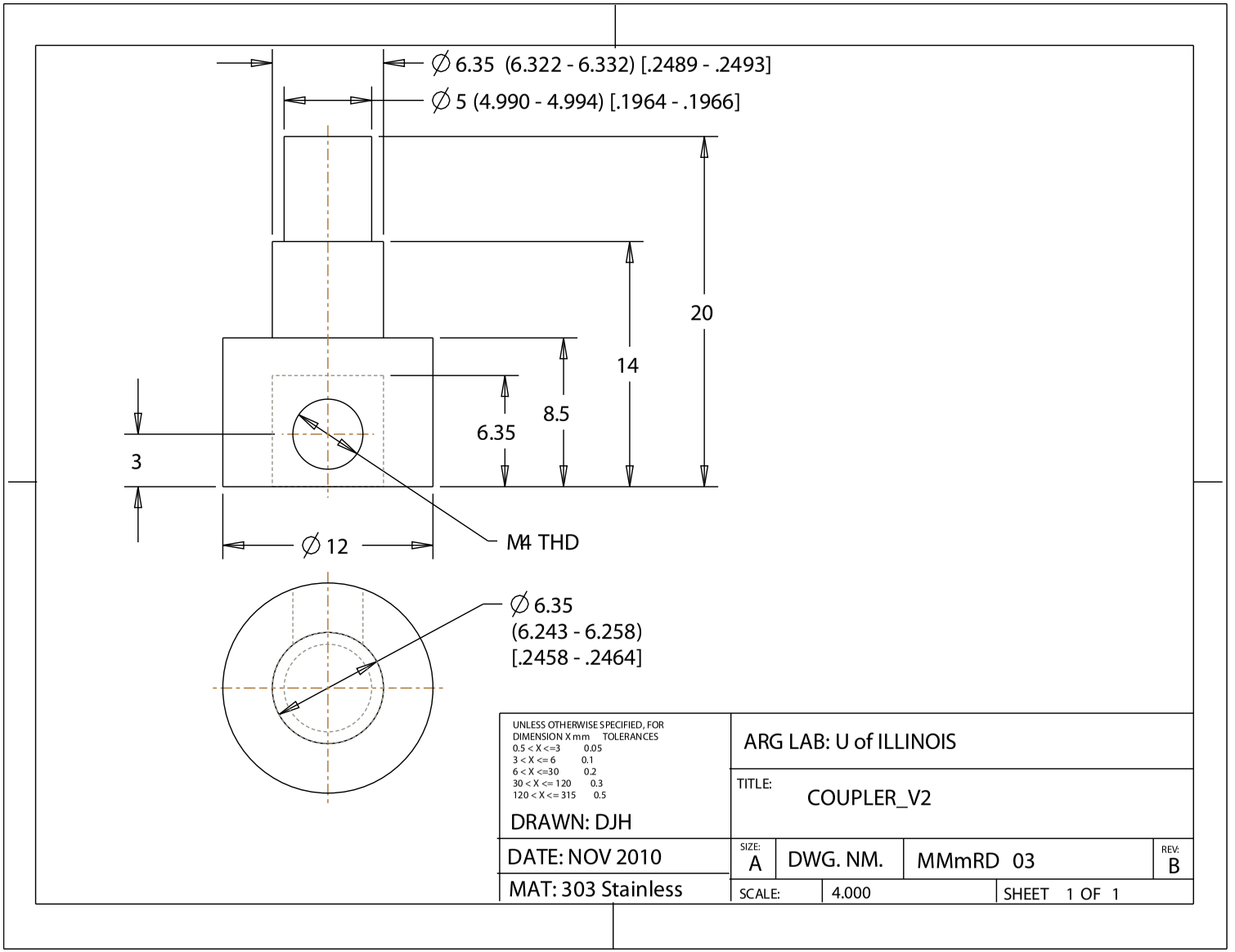
Appendix E

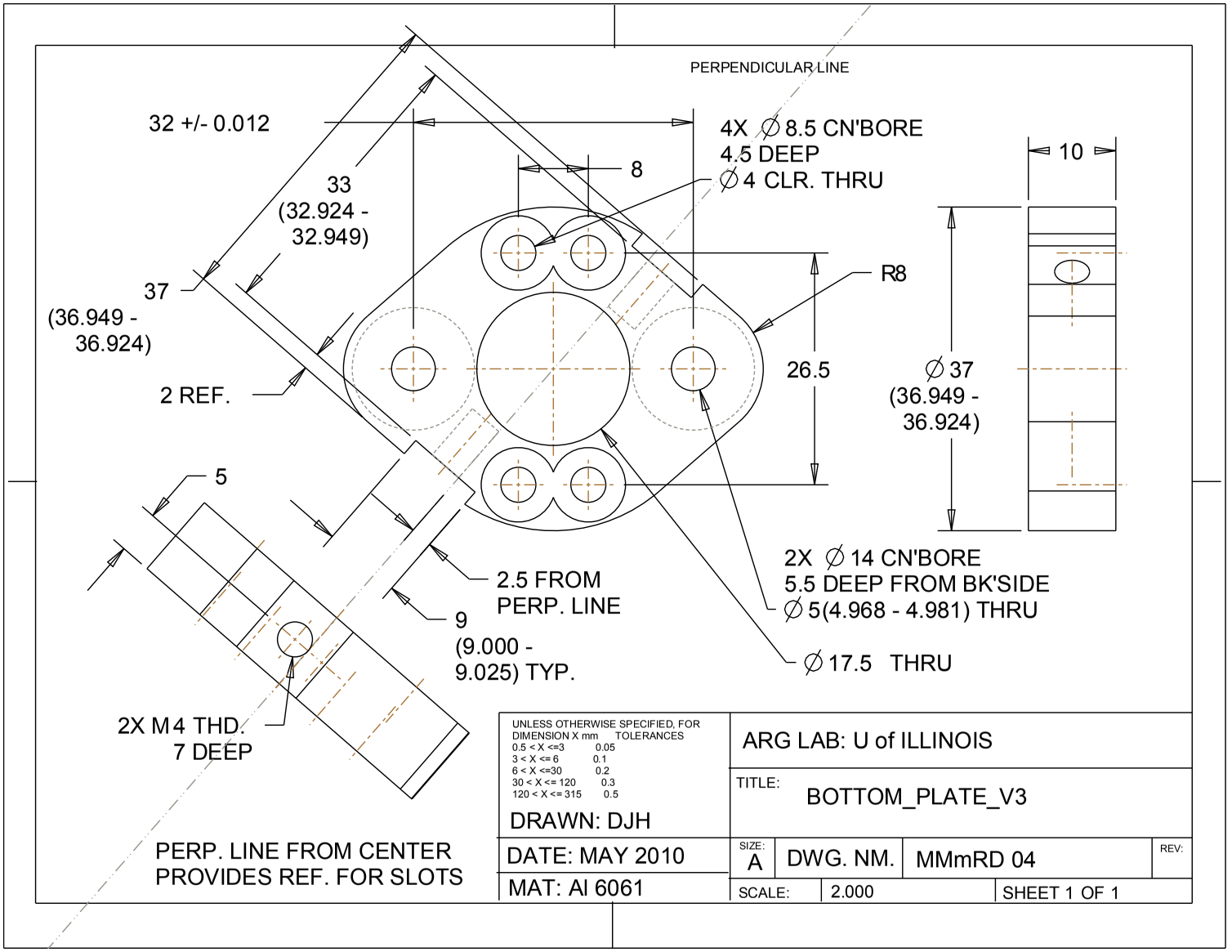
Multi-Material Deposition System

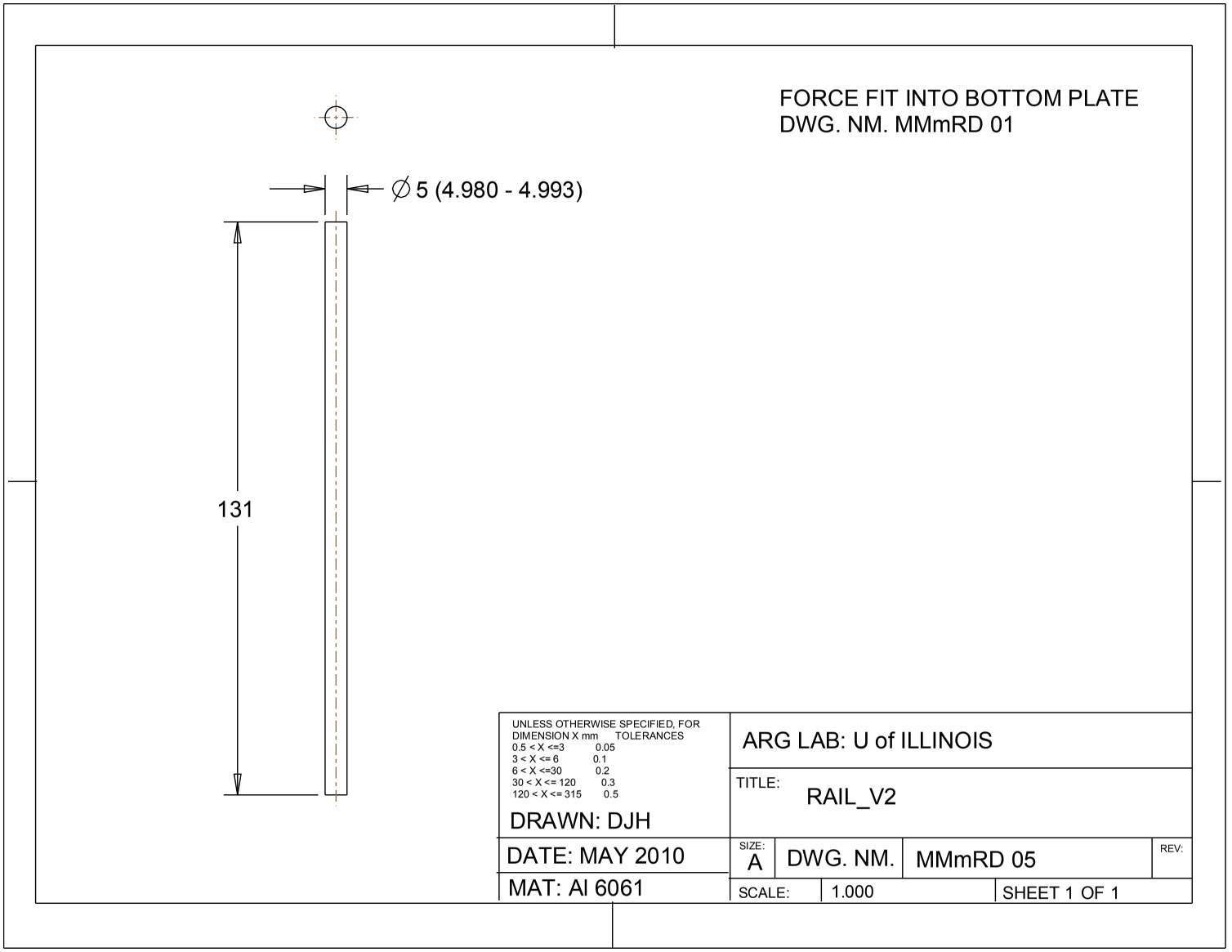
Appendix E provides the complete set of the engineering prints and wiring diagrams for the design of the Multi-Material Deposition System.

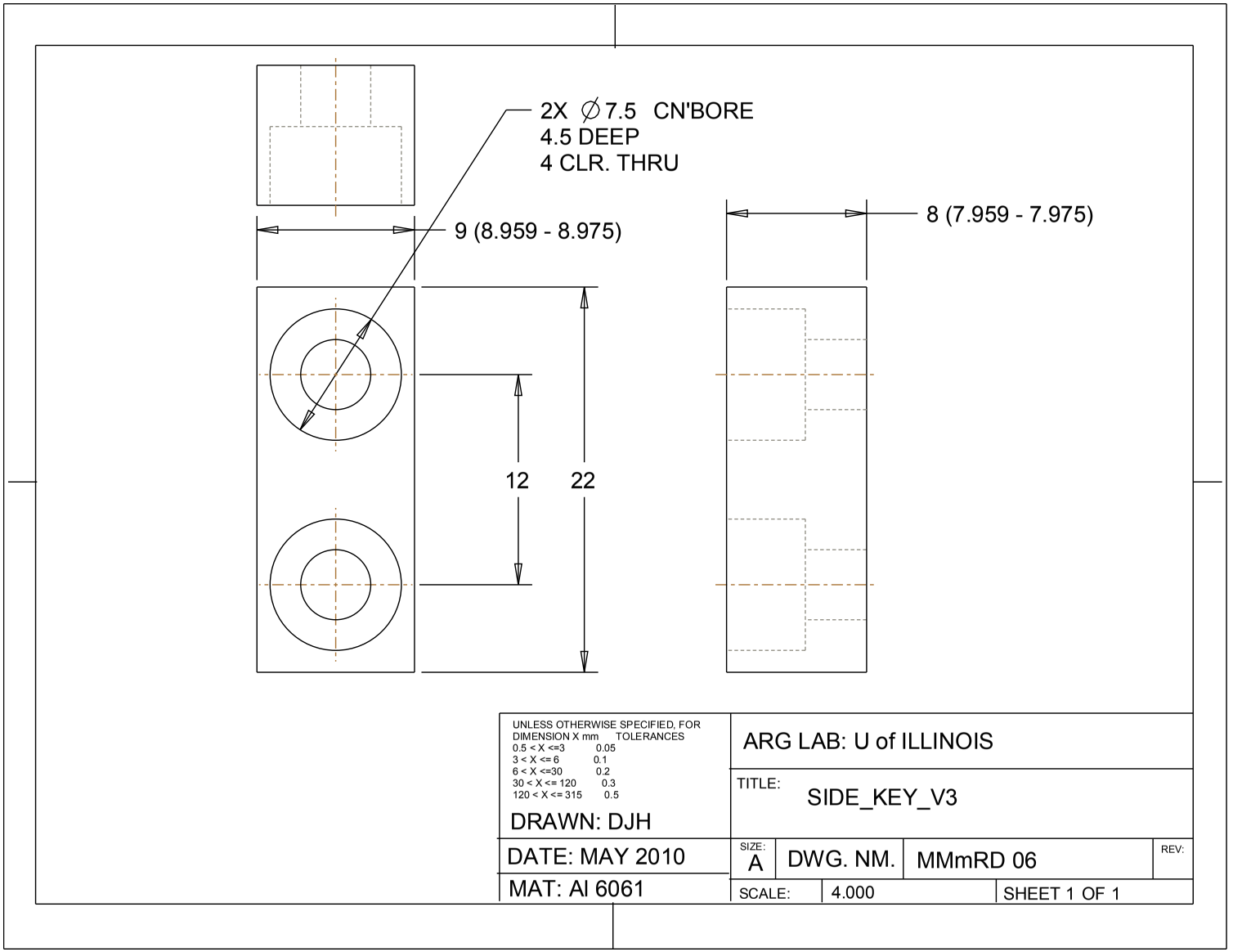

43 REF.
4X ∅8.5 CN'BORE
4.5 DEEP
∅4 CLR. THRU
∅17.5
125.7 ° REF.
R5.5 TYP.
2X ∅5(4.962 - 4.970) THRU
11
26.5
4X ∅6 CN'BORE
4.5 DEEP FROM BK'SD
∅3 CLR. THRU
8
19.05
32 +/- .012
∅37
7
UNLESS OTHERWISE SPECIFIED, FOR
DIMENSION X mm TOLERANCES
0.5 < X <=3 0.05
3 < X <= 6 0.1
6 < X <=30 0.2
30 < X <= 120 0.3
120 < X <= 315 0.5
DRAWN: DJH
DATE: MAY 2010
MAT: Al 6061
ARG LAB: U of ILLINOIS
TITLE:
TOP_PLATE_V3
SIZE:
A
DWG. NM.
MMmRD 01
REV:
SCALE:
2.000
SHEET 1 OF 1

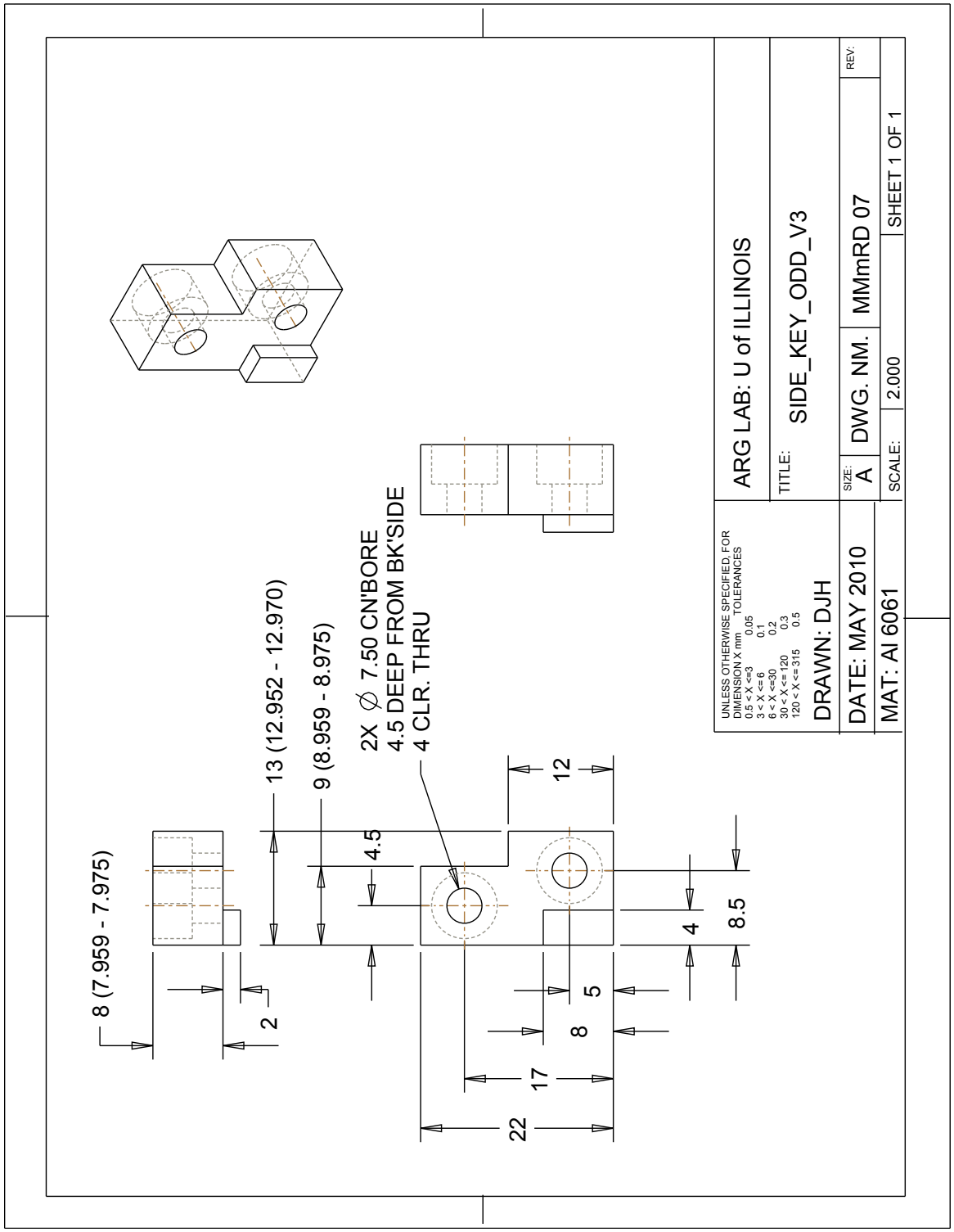

NOTE: FOR TIGHT RUNNING
CLR. WITH 2mm DOWEL
SPEC. ATTACHED
6.5
115
93
10
16
8
4X M4 THD.
16 DEEP
2X ∅2
(2.016 - 2.026)
THRU
SEE NOTE
UNLESS OTHERWISE SPECIFIED, FOR
DIMENSION X mm TOLERANCES
0.5 < X <=3 0.05
3 < X <= 6 0.1
6 < X <=30 0.2
30 < X <= 120 0.3
120 < X <= 315 0.5
DRAWN: DJH
DATE: MAY 2010
MAT: Al 6061
ARG LAB: U of ILLINOIS
TITLE:
SIDE_BAR_V2
SIZE:
A
DWG. NM.
MMmRD 02
REV:
SCALE:
1.000
SHEET 1 OF 1

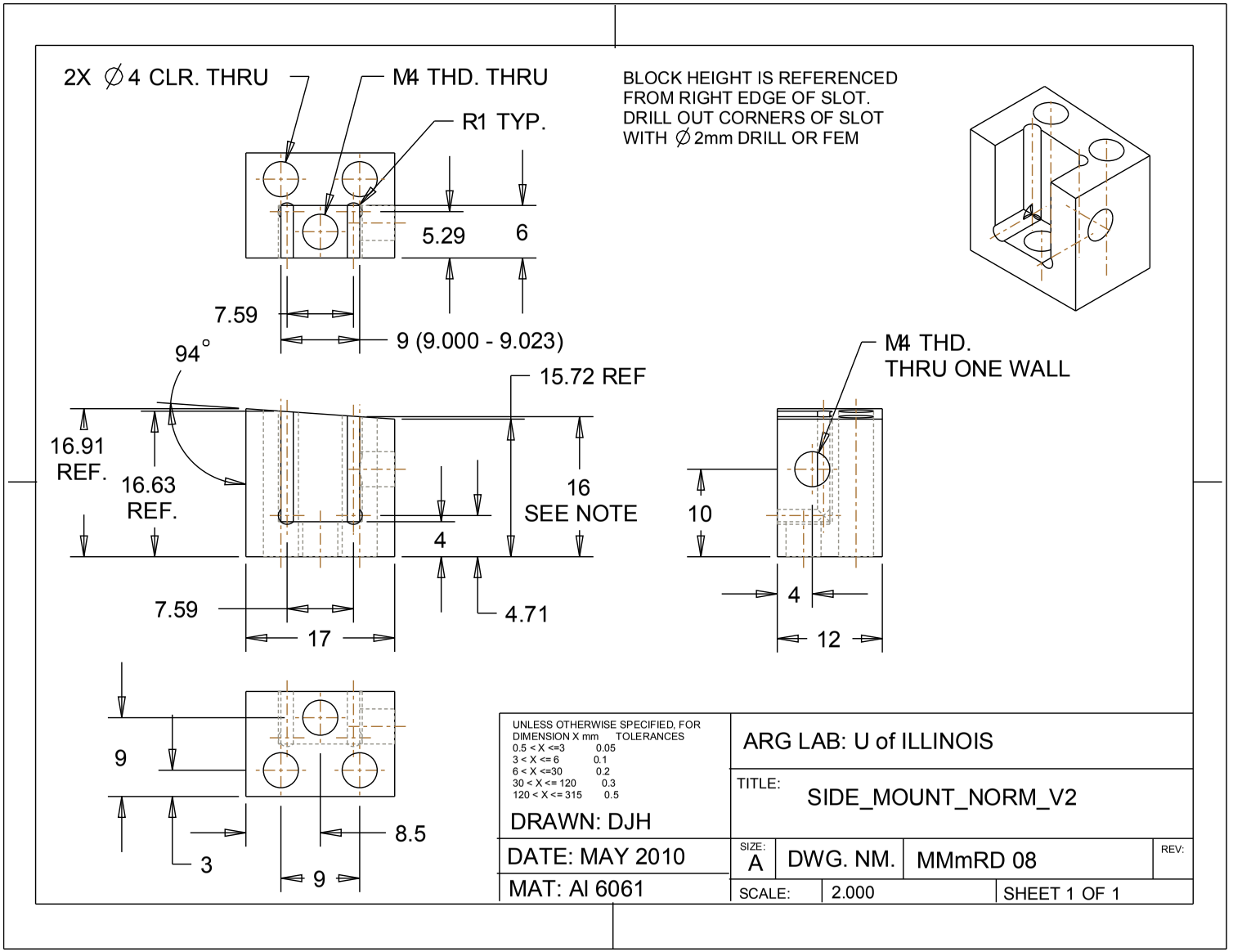

∅ 6.35 (6.322 - 6.332) [.2489 - .2493]
∅ 5 (4.990 - 4.994) [.1964 - .1966]
20
14
8.5
6.35
3
∅ 12
M4 THD
∅ 6.35
(6.243 - 6.258)
[.2458 - .2464]
UNLESS OTHERWISE SPECIFIED, FOR
DIMENSION X mm TOLERANCES
0.5 < X <=3 0.05
3 < X <= 6 0.1
6 < X <=30 0.2
30 < X <= 120 0.3
120 < X <= 315 0.5
DRAWN: DJH
DATE: NOV 2010
MAT: 303 Stainless
ARG LAB: U of ILLINOIS
TITLE:
COUPLER_V2
SIZE:
A
DWG. NM.
MMmRD 03
REV:
B
SCALE:
4.000
SHEET 1 OF 1

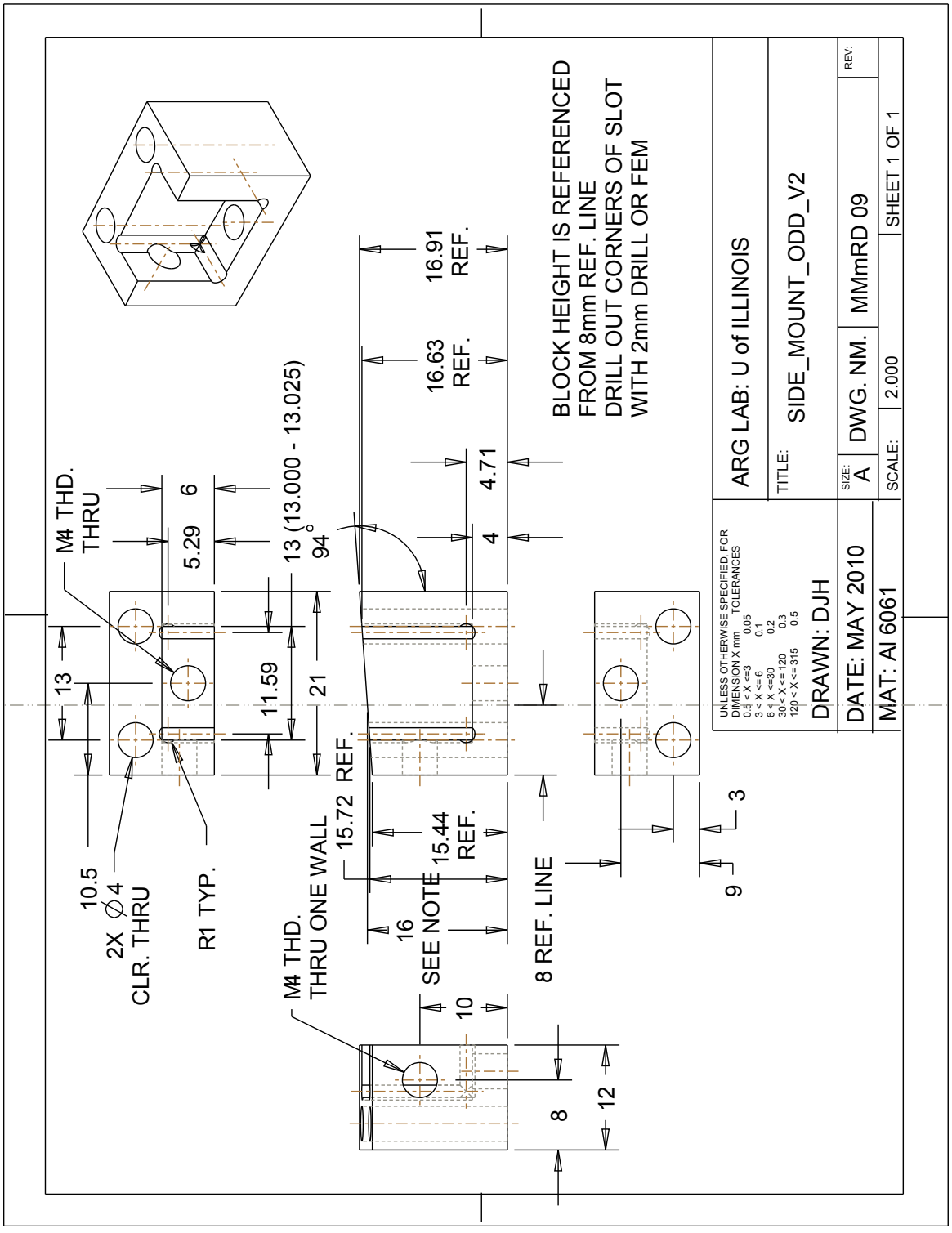

PERPENDICULAR LINE
32 +/- 0.012
33
(32.924 -
32.949)
37
(36.949 -
36.924)
2 REF.
4X ∅8.5 CN'BORE
4.5 DEEP
∅4 CLR. THRU
8
R8
26.5
10
∅37
(36.949 -
36.924)
5
2.5 FROM
PERP. LINE
9
(9.000 -
9.025) TYP.
2X ∅14 CN'BORE
5.5 DEEP FROM BK'SIDE
∅5(4.968 - 4.981) THRU
∅17.5 THRU
2X M4 THD.
7 DEEP
PERP. LINE FROM CENTER
PROVIDES REF. FOR SLOTS
UNLESS OTHERWISE SPECIFIED, FOR
DIMENSION X mm TOLERANCES
0.5 < X <=3 0.05
3 < X <= 6 0.1
6 < X <=30 0.2
30 < X <= 120 0.3
120 < X <= 315 0.5
DRAWN: DJH
DATE: MAY 2010
MAT: Al 6061
ARG LAB: U of ILLINOIS
TITLE:
BOTTOM_PLATE_V3
SIZE:
A
DWG. NM.
MMmRD 04
REV:
SCALE:
2.000
SHEET 1 OF 1

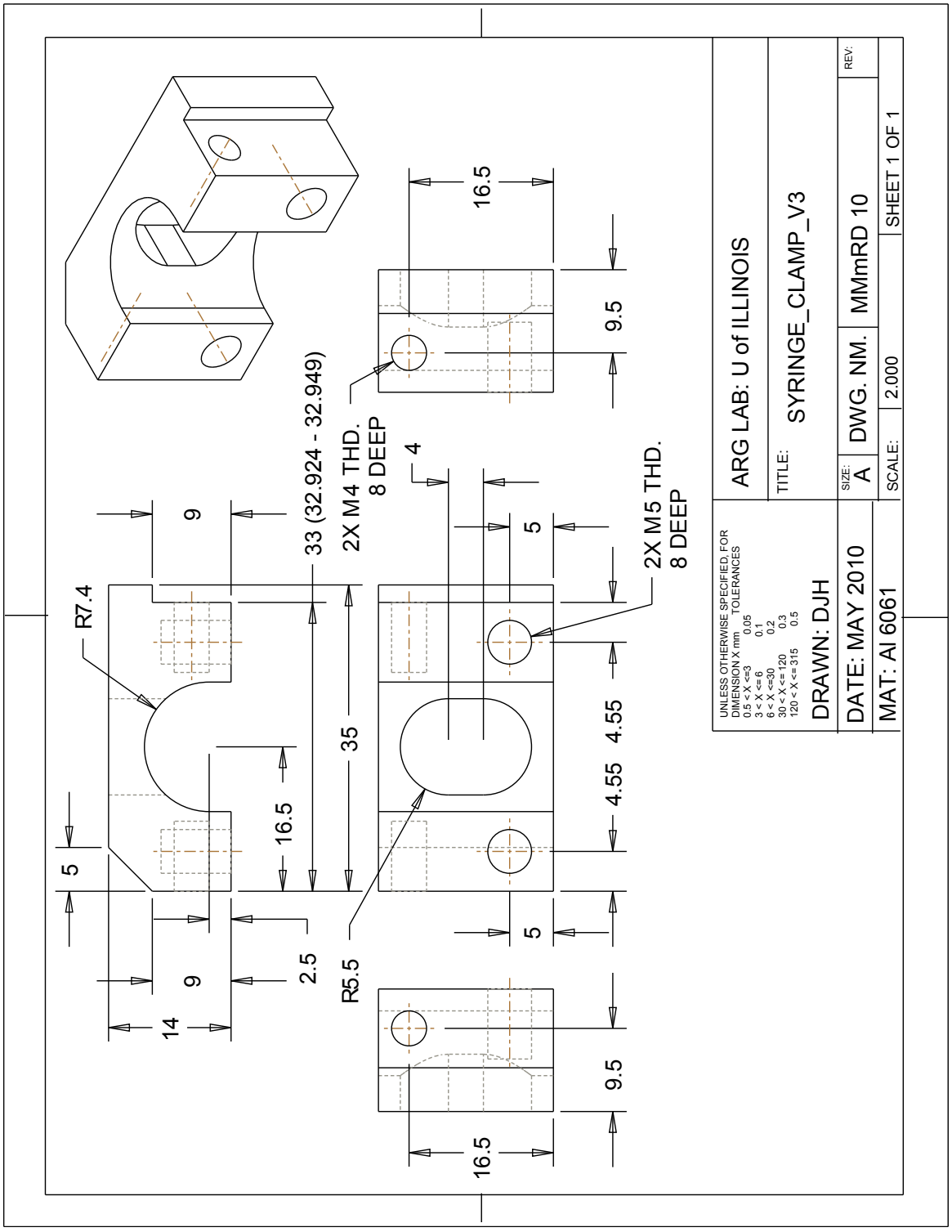

FORCE FIT INTO BOTTOM PLATE
DWG. NM. MMmRD 01
∅ 5 (4.980 - 4.993)
131
UNLESS OTHERWISE SPECIFIED, FOR
DIMENSION X mm TOLERANCES
0.5 < X <=3 0.05
3 < X <= 6 0.1
6 < X <=30 0.2
30 < X <= 120 0.3
120 < X <= 315 0.5
DRAWN: DJH
DATE: MAY 2010
MAT: Al 6061
ARG LAB: U of ILLINOIS
TITLE:
RAIL_V2
SIZE:
A
DWG. NM.
MMmRD 05
REV:
SCALE:
1.000
SHEET 1 OF 1

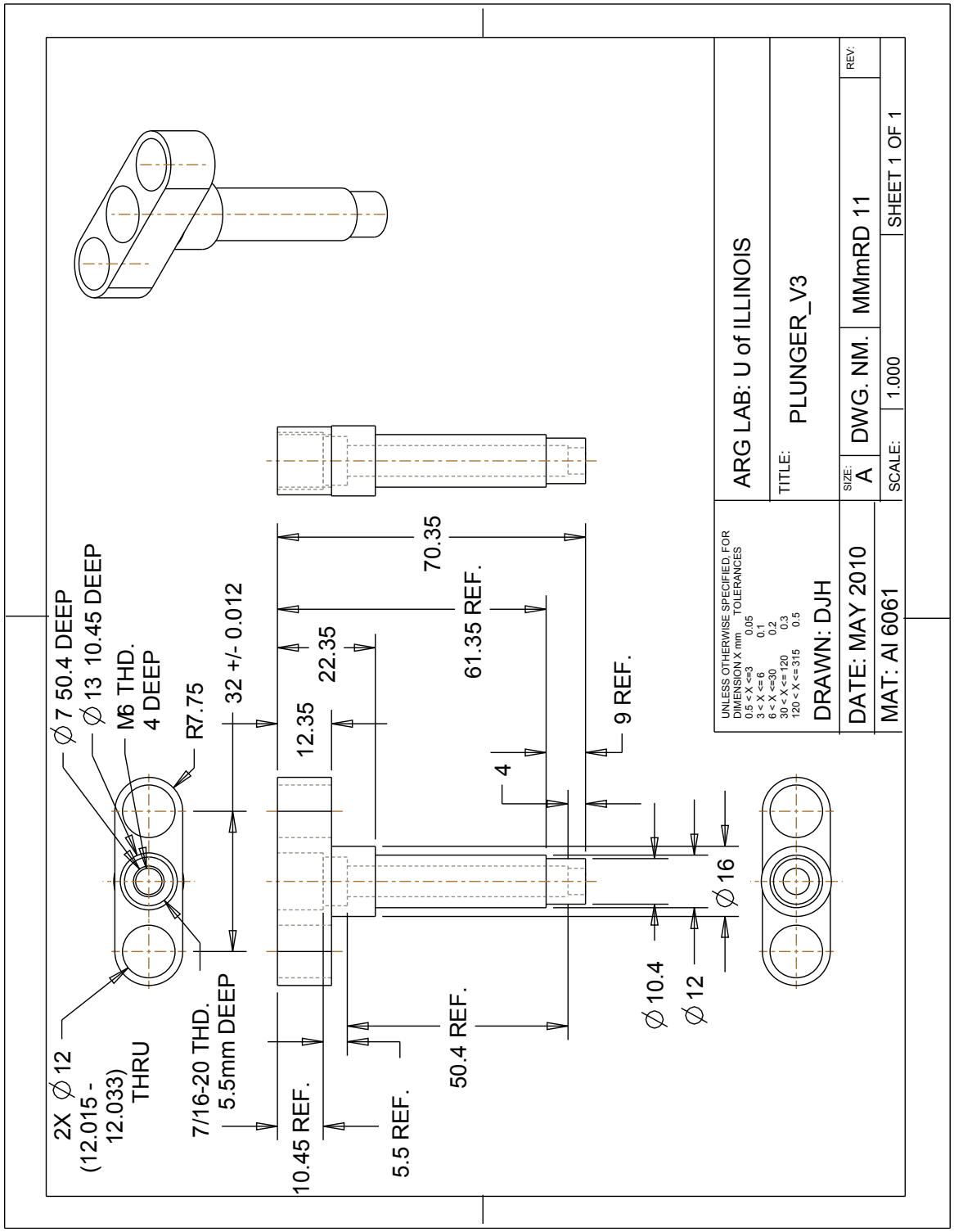

2X ∅7.5 CN'BORE
4.5 DEEP
4 CLR. THRU
9 (8.959 - 8.975)
8 (7.959 - 7.975)
12
22
UNLESS OTHERWISE SPECIFIED, FOR
DIMENSION X mm TOLERANCES
0.5 < X <=3 0.05
3 < X <= 6 0.1
6 < X <=30 0.2
30 < X <= 120 0.3
120 < X <= 315 0.5
DRAWN: DJH
DATE: MAY 2010
MAT: Al 6061
ARG LAB: U of ILLINOIS
TITLE:
SIDE_KEY_V3
SIZE:
A
DWG. NM.
MMmRD 06
REV:
SCALE:
4.000
SHEET 1 OF 1

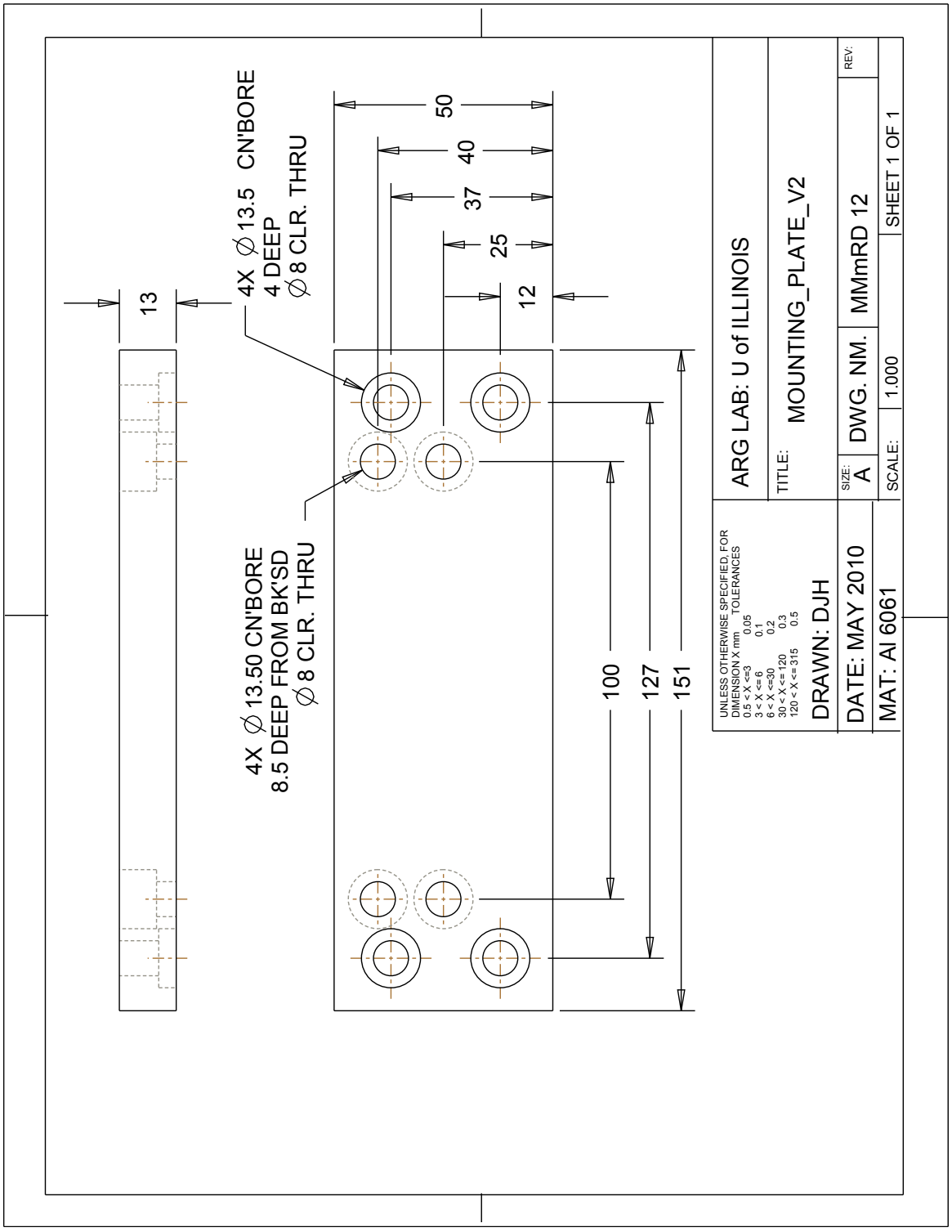

8 (7.959 - 7.975)
2
13 (12.952 - 12.970)
9 (8.959 - 8.975)
4.5
2X ∅ 7.50 CN'BORE
4.5 DEEP FROM BK'SIDE
4 CLR. THRU
12
4
8.5
5
8
17
22
UNLESS OTHERWISE SPECIFIED, FOR
DIMENSION X mm TOLERANCES
0.5 < X <=3 0.05
3 < X <= 6 0.1
6 < X <=30 0.2
30 < X <= 120 0.3
120 < X <= 315 0.5
DRAWN: DJH
DATE: MAY 2010
MAT: Al 6061
ARG LAB: U of ILLINOIS
TITLE:
SIDE_KEY_ODD_V3
SIZE:
A
DWG. NM.
MMmRD 07
REV:
SCALE:
2.000
SHEET 1 OF 1

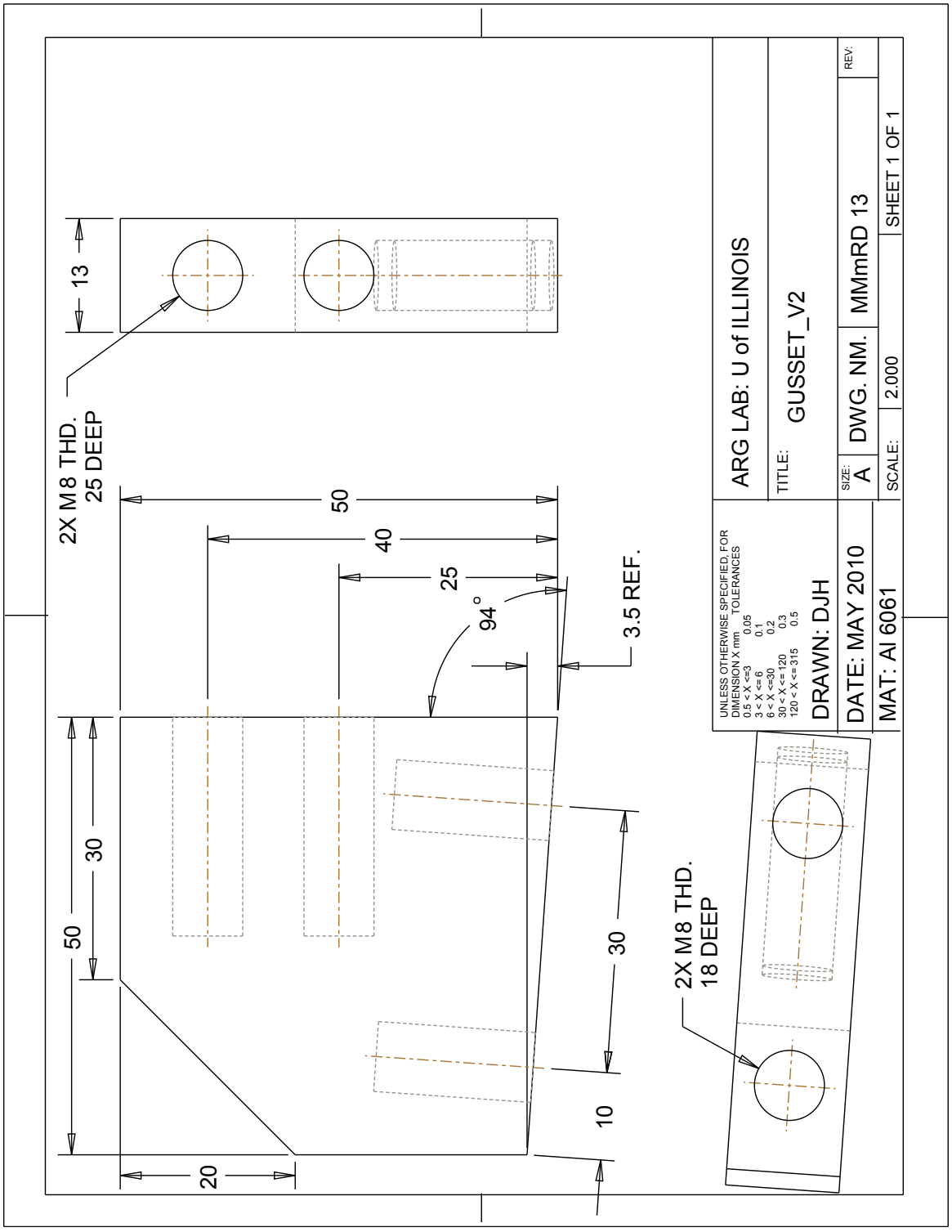

2X ∅4 CLR. THRU
M4 THD. THRU
R1 TYP.
BLOCK HEIGHT IS REFERENCED FROM RIGHT EDGE OF SLOT. DRILL OUT CORNERS OF SLOT WITH ∅2mm DRILL OR FEM
5.29
6
7.59
9 (9.000 - 9.023)
94°
15.72 REF
16.91 REF.
16.63 REF.
16 SEE NOTE
4
7.59
17
4.71
M4 THD. THRU ONE WALL
10
4
12
9
8.5
3
9
UNLESS OTHERWISE SPECIFIED, FOR DIMENSION X mm TOLERANCES
0.5 < X <=3 0.05
3 < X <= 6 0.1
6 < X <=30 0.2
30 < X <= 120 0.3
120 < X <= 315 0.5
DRAWN: DJH
DATE: MAY 2010
MAT: Al 6061
ARG LAB: U of ILLINOIS
TITLE: SIDE_MOUNT_NORM_V2
SIZE: A
DWG. NM. MMmRD 08
REV:
SCALE: 2.000
SHEET 1 OF 1

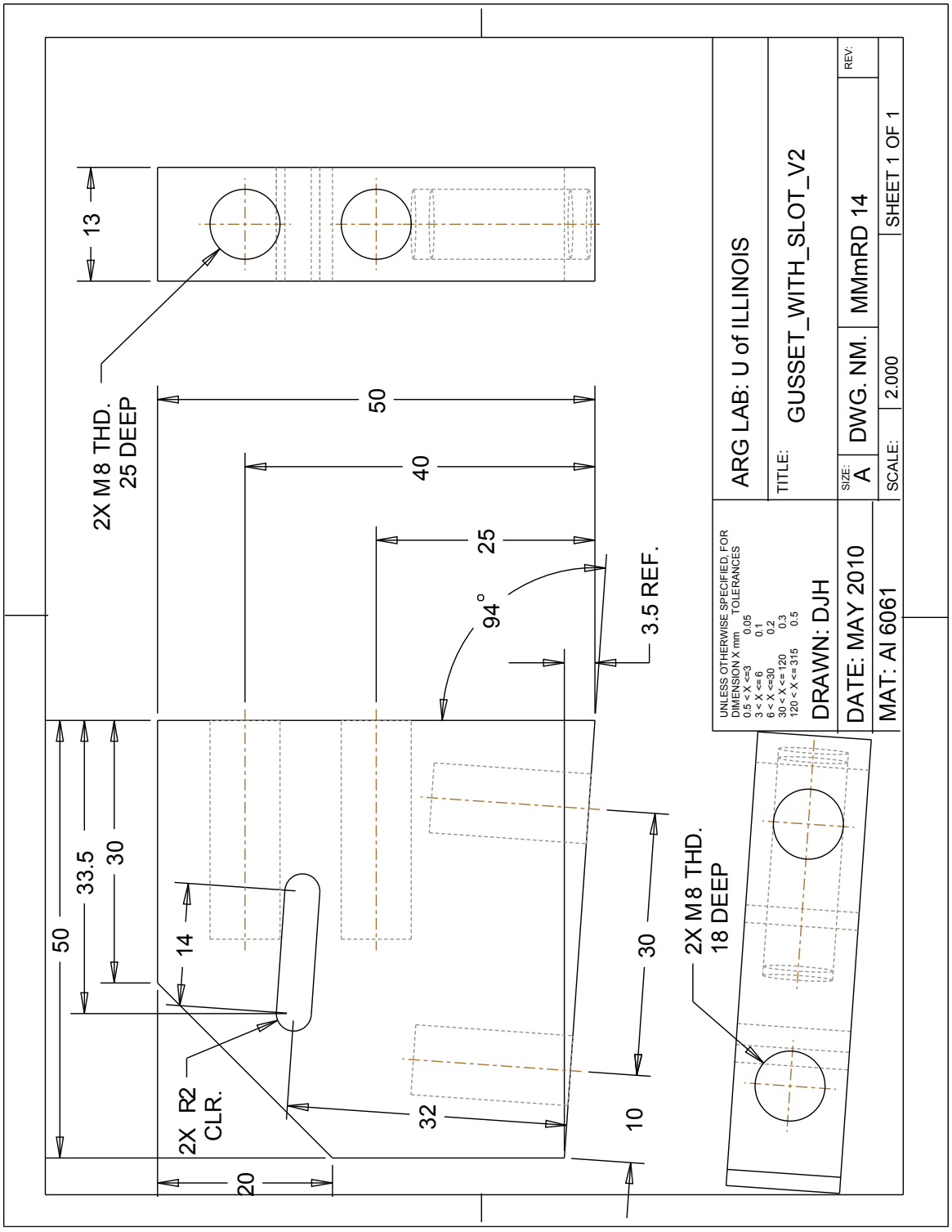

M4 THD.
THRU
13
5.29
6
13 (13.000 - 13.025)
94°
11.59
21
10.5
2X ∅4
CLR. THRU
R1 TYP.
M4 THD.
THRU ONE WALL
15.72 REF.
16
SEE NOTE
15.44
REF.
10
8
12
8 REF. LINE
16.63
REF.
16.91
REF.
4.71
4
3
9
BLOCK HEIGHT IS REFERENCED
FROM 8mm REF. LINE
DRILL OUT CORNERS OF SLOT
WITH 2mm DRILL OR FEM
UNLESS OTHERWISE SPECIFIED, FOR
DIMENSION X mm TOLERANCES
0.5 < X <=3 0.05
3 < X <= 6 0.1
6 < X <=30 0.2
30 < X <= 120 0.3
120 < X <= 315 0.5
DRAWN: DJH
DATE: MAY 2010
MAT: Al 6061
ARG LAB: U of ILLINOIS
TITLE:
SIDE_MOUNT_ODD_V2
SIZE:
A
DWG. NM.
MMmRD 09
REV:
SCALE:
2.000
SHEET 1 OF 1

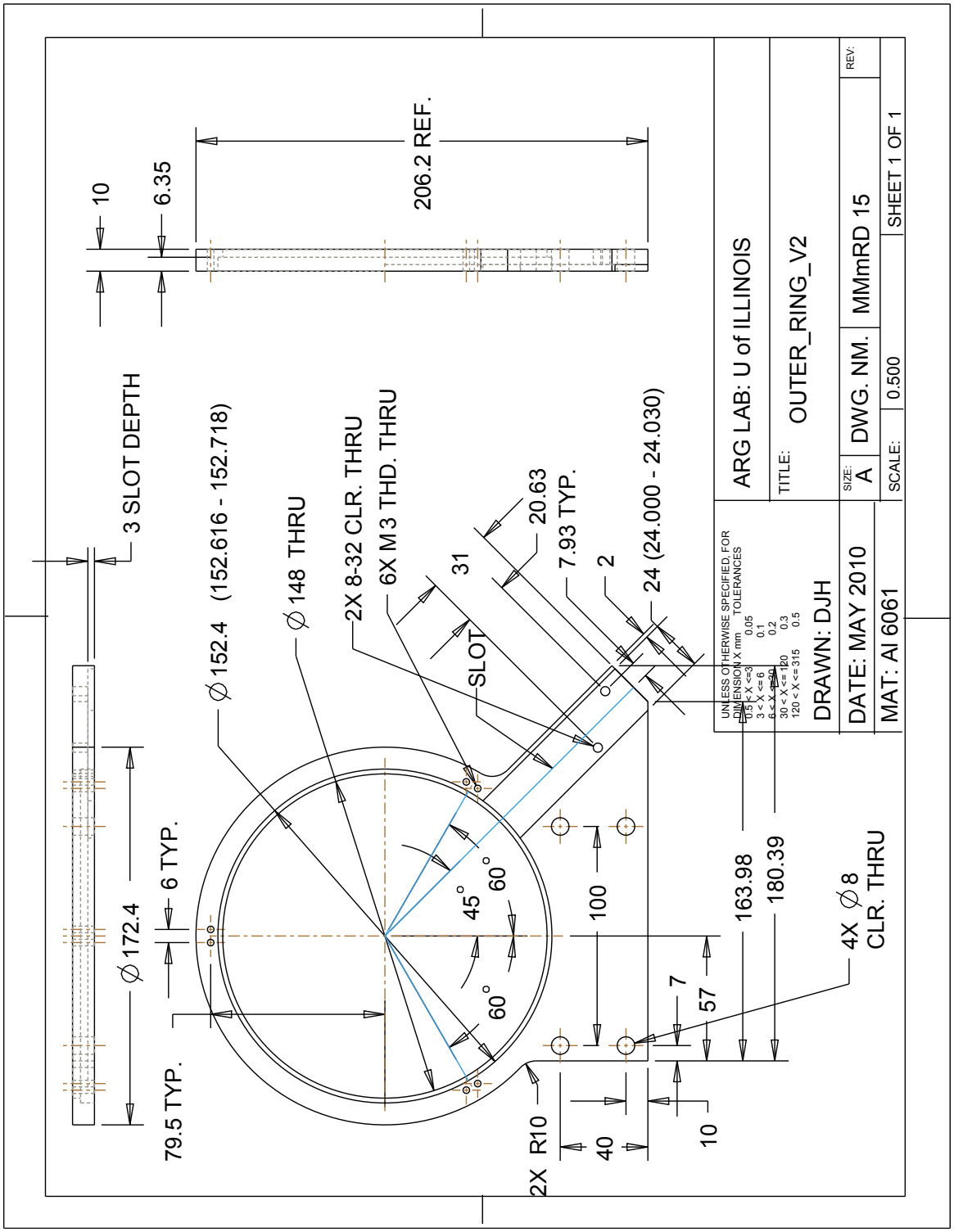

R7.4
9
33 (32.924 - 32.949)
2X M4 THD.
8 DEEP
4
5
2X M5 THD.
8 DEEP
16.5
9.5
5
16.5
35
4.55
4.55
5
2.5
R5.5
9
14
9.5
16.5
UNLESS OTHERWISE SPECIFIED, FOR
DIMENSION X mm TOLERANCES
0.5 < X <=3 0.05
3 < X <= 6 0.1
6 < X <=30 0.2
30 < X <= 120 0.3
120 < X <= 315 0.5
ARG LAB: U of ILLINOIS
TITLE:
SYRINGE_CLAMP_V3
DRAWN: DJH
SIZE:
A
DWG. NM.
MMmRD 10
REV:
DATE: MAY 2010
SCALE:
2.000
SHEET 1 OF 1
MAT: AI 6061

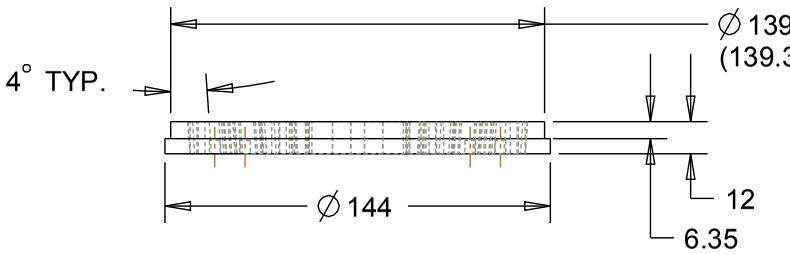

2X ∅12
(12.015 -
12.033)
THRU
7/16-20 THD.
5.5mm DEEP
∅7 50.4 DEEP
∅13 10.45 DEEP
M6 THD.
4 DEEP
R7.75
32 +/- 0.012
10.45 REF.
5.5 REF.
12.35
22.35
50.4 REF.
70.35
61.35 REF.
4
9 REF.
∅10.4
∅12
∅16
UNLESS OTHERWISE SPECIFIED, FOR
DIMENSION X mm TOLERANCES
0.5 < X <=3 0.05
3 < X <= 6 0.1
6 < X <=30 0.2
30 < X <= 120 0.3
120 < X <= 315 0.5
DRAWN: DJH
DATE: MAY 2010
MAT: Al 6061
ARG LAB: U of ILLINOIS
TITLE:
PLUNGER_V3
SIZE:
A
DWG. NM.
MMmRD 11
REV:
SCALE:
1.000
SHEET 1 OF 1

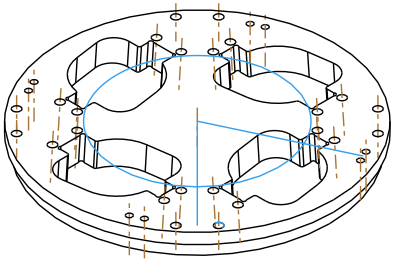

13
4X ∅13.5 CN'BORE
4 DEEP
∅8 CLR. THRU
4X ∅13.50 CN'BORE
8.5 DEEP FROM BK'SD
∅8 CLR. THRU
50
40
37
25
12
100
127
151
UNLESS OTHERWISE SPECIFIED, FOR
DIMENSION X mm TOLERANCES
0.5 < X <=3 0.05
3 < X <= 6 0.1
6 < X <=30 0.2
30 < X <= 120 0.3
120 < X <= 315 0.5
DRAWN: DJH
DATE: MAY 2010
MAT: Al 6061
ARG LAB: U of ILLINOIS
TITLE:
MOUNTING_PLATE_V2
SIZE:
A
DWG. NM.
MMmRD 12
REV:
SCALE:
1.000
SHEET 1 OF 1

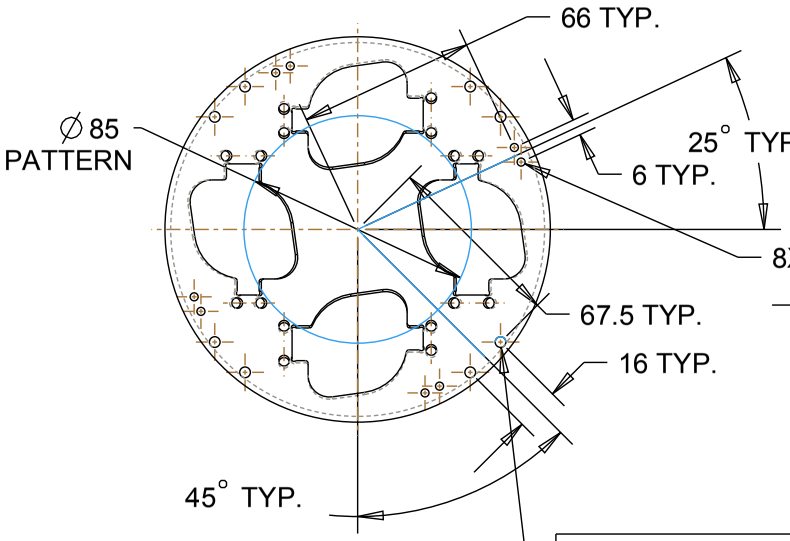

2X M8 THD.
25 DEEP
13
50
40
25
94°
3.5 REF.
50
30
20
10
30
2X M8 THD.
18 DEEP
UNLESS OTHERWISE SPECIFIED, FOR
DIMENSION X mm TOLERANCES
0.5 < X <=3 0.05
3 < X <= 6 0.1
6 < X <=30 0.2
30 < X <= 120 0.3
120 < X <= 315 0.5
DRAWN: DJH
DATE: MAY 2010
MAT: Al 6061
ARG LAB: U of ILLINOIS
TITLE:
GUSSET_V2
SIZE:
A
DWG. NM.
MMmRD 13
REV:
SCALE:
2.000
SHEET 1 OF 1


13
2X M8 THD.
25 DEEP
50
40
25
94°
3.5 REF.
50
33.5
30
14
2X R2
CLR.
32
30
10
20
2X M8 THD.
18 DEEP
UNLESS OTHERWISE SPECIFIED, FOR
DIMENSION X mm TOLERANCES
0.5 < X <=3 0.05
3 < X <= 6 0.1
6 < X <=30 0.2
30 < X <= 120 0.3
120 < X <= 315 0.5
DRAWN: DJH
DATE: MAY 2010
MAT: AI 6061
ARG LAB: U of ILLINOIS
TITLE:
GUSSET_WITH_SLOT_V2
SIZE:
A
DWG. NM.
MMmRD 14
REV:
SCALE:
2.000
SHEET 1 OF 1


10
6.35
206.2 REF.
3 SLOT DEPTH
Ø152.4 (152.616 - 152.718)
Ø148 THRU
2X 8-32 CLR. THRU
6X M3 THD. THRU
31
SLOT
20.63
7.93 TYP.
2
24 (24.000 - 24.030)
Ø172.4
6 TYP.
79.5 TYP.
45°
60°
60°
100
163.98
180.39
4X Ø8
CLR. THRU
7
57
2X R10
40
10
UNLESS OTHERWISE SPECIFIED, FOR
DIMENSION X mm TOLERANCES
0.5 < X <=3 0.05
3 < X <= 6 0.1
6 < X <= 30 0.2
30 < X <= 120 0.3
120 < X <= 315 0.5
ARG LAB: U of ILLINOIS
TITLE: OUTER_RING_V2
DRAWN: DJH
SIZE: A
DWG. NM. MMmRD 15
REV:
DATE: MAY 2010
SCALE: 0.500
SHEET 1 OF 1
MAT: Al 6061

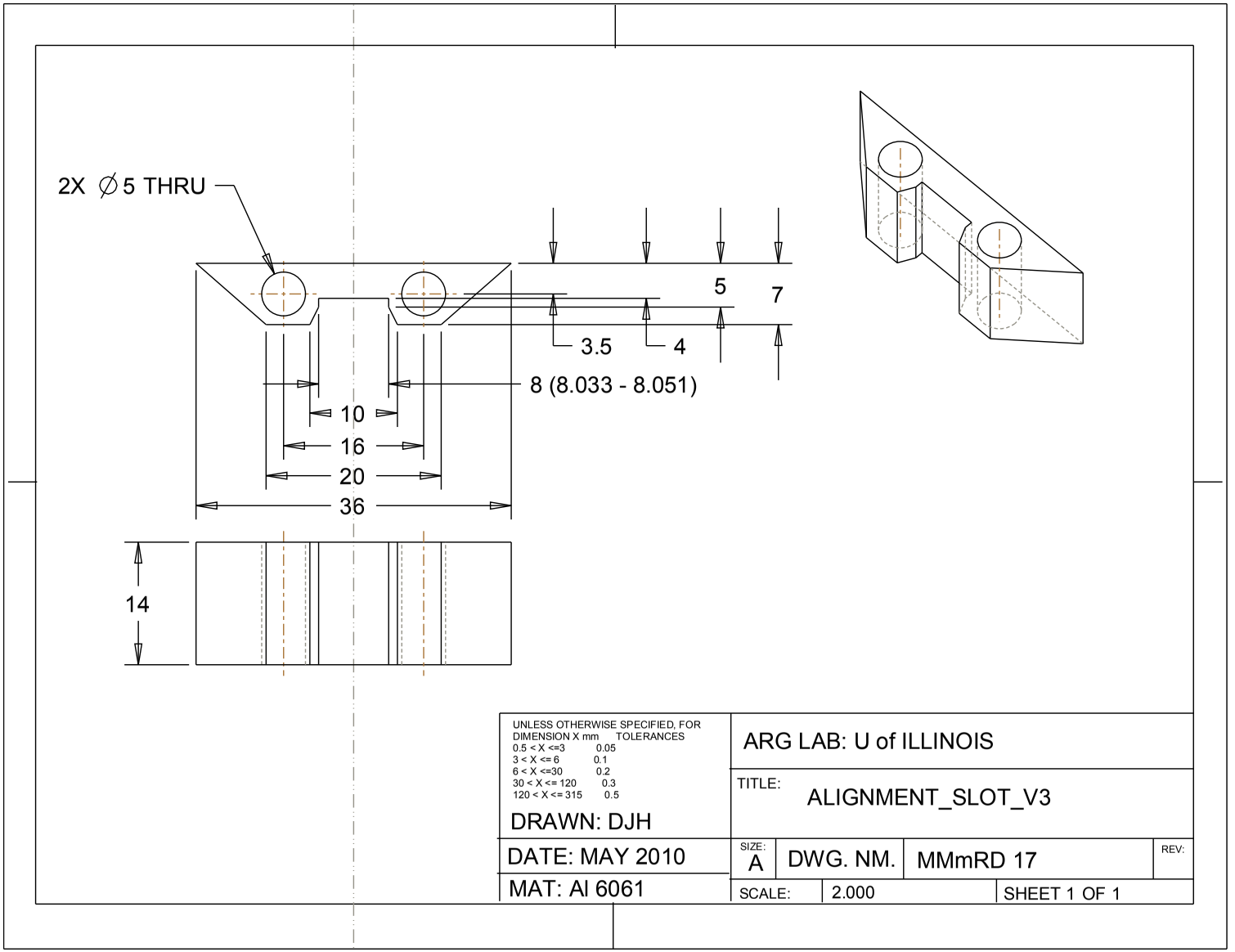






SEE SHT. 2 FOR PATTERN DETAIL.
PATTERN CENTER POINT IS ON 85mm CIRCLE. PATTER ANGLED 4deg TOWARDS TURRET CENTER. RECT. SLOTS PARALLEL WITH TURRET CL.

UNLESS OTHERWISE SPECIFIED, FOR DIMENSION X mm TOLERANCES 0.5 < X <=3 0.05 3 < X <= 6 0.1 6 < X <=30 0.2 30 < X <= 120 0.3 120 < X <= 315 0.5	ARG LAB: U of ILLINOIS			
DRAWN: DJH	TITLE: TURRET_V2			
DATE: MAY 2010	SIZE: A	DWG. NM.	MMmRD 16	REV:
MAT: Al 6061	SCALE:	0.500	SHEET 1 OF 2	



UNLESS OTHERWISE SPECIFIED, FOR DIMENSION X mm TOLERANCES
0.5 < X <=3 0.05
3 < X <= 6 0.1
6 < X <=30 0.2
30 < X <= 120 0.3
120 < X <= 315 0.5

DRAWN: DJH

DATE: MAY 2010

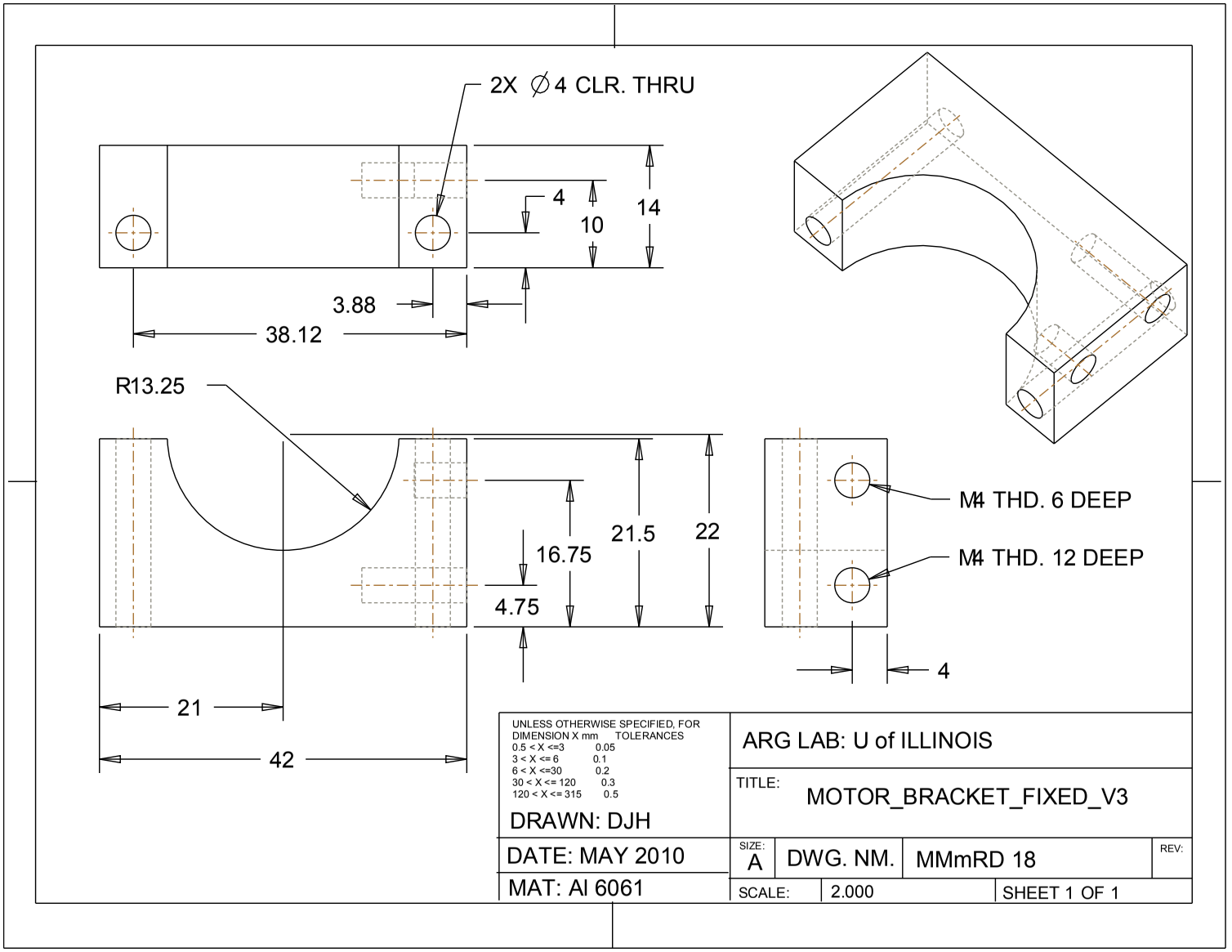
MAT: Al 6061

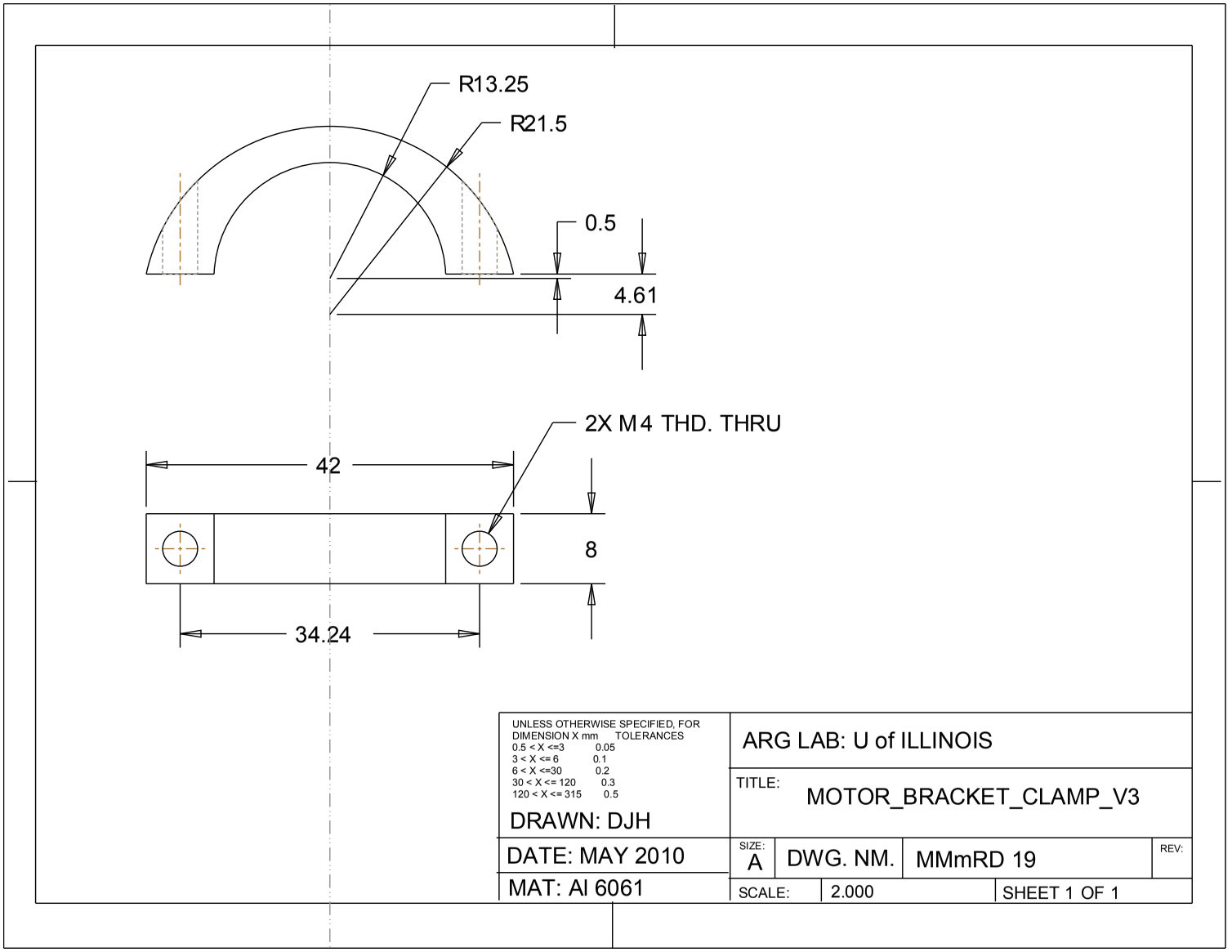
ARG LAB: U of ILLINOIS

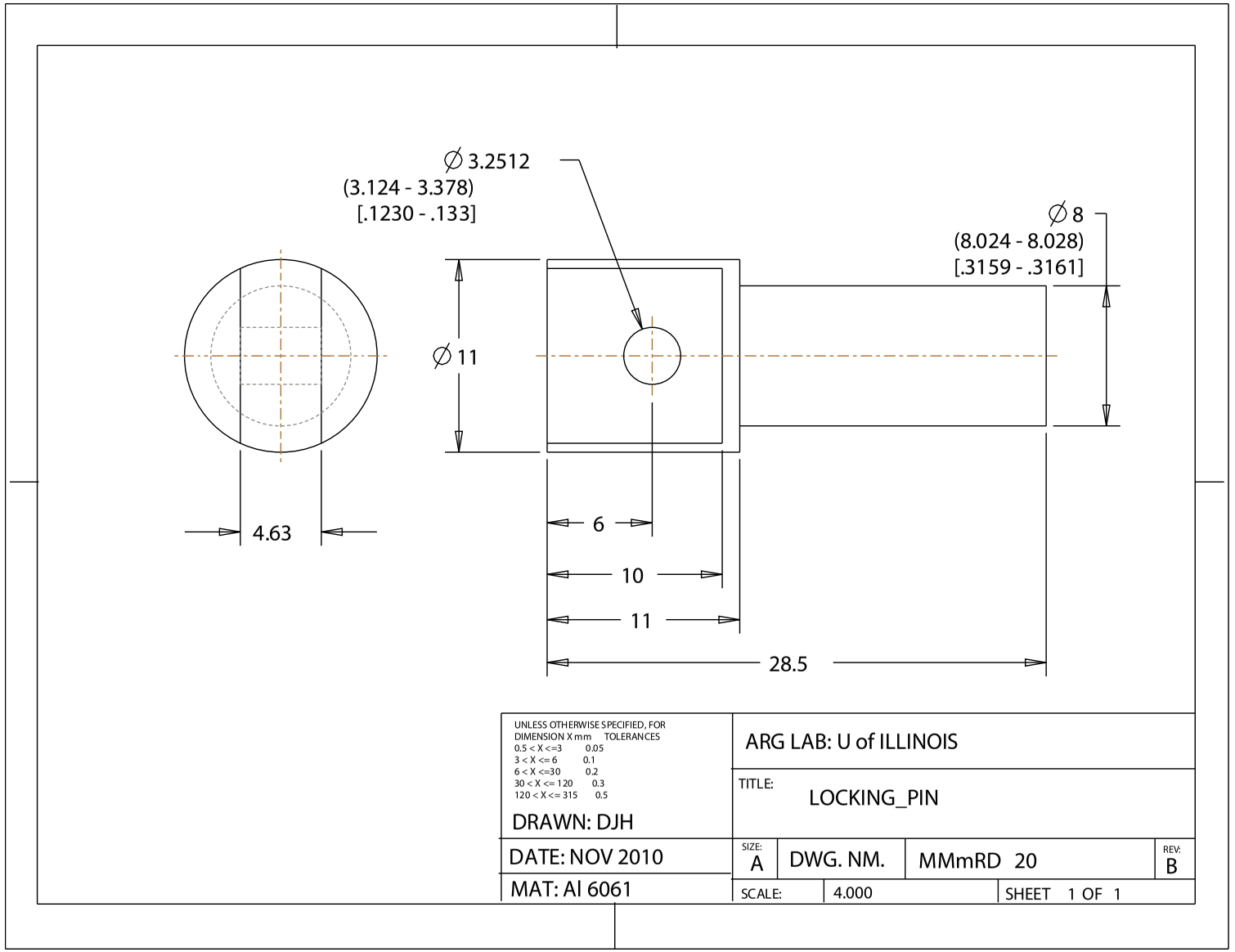
TITLE: ALIGNMENT_SLOT_V3

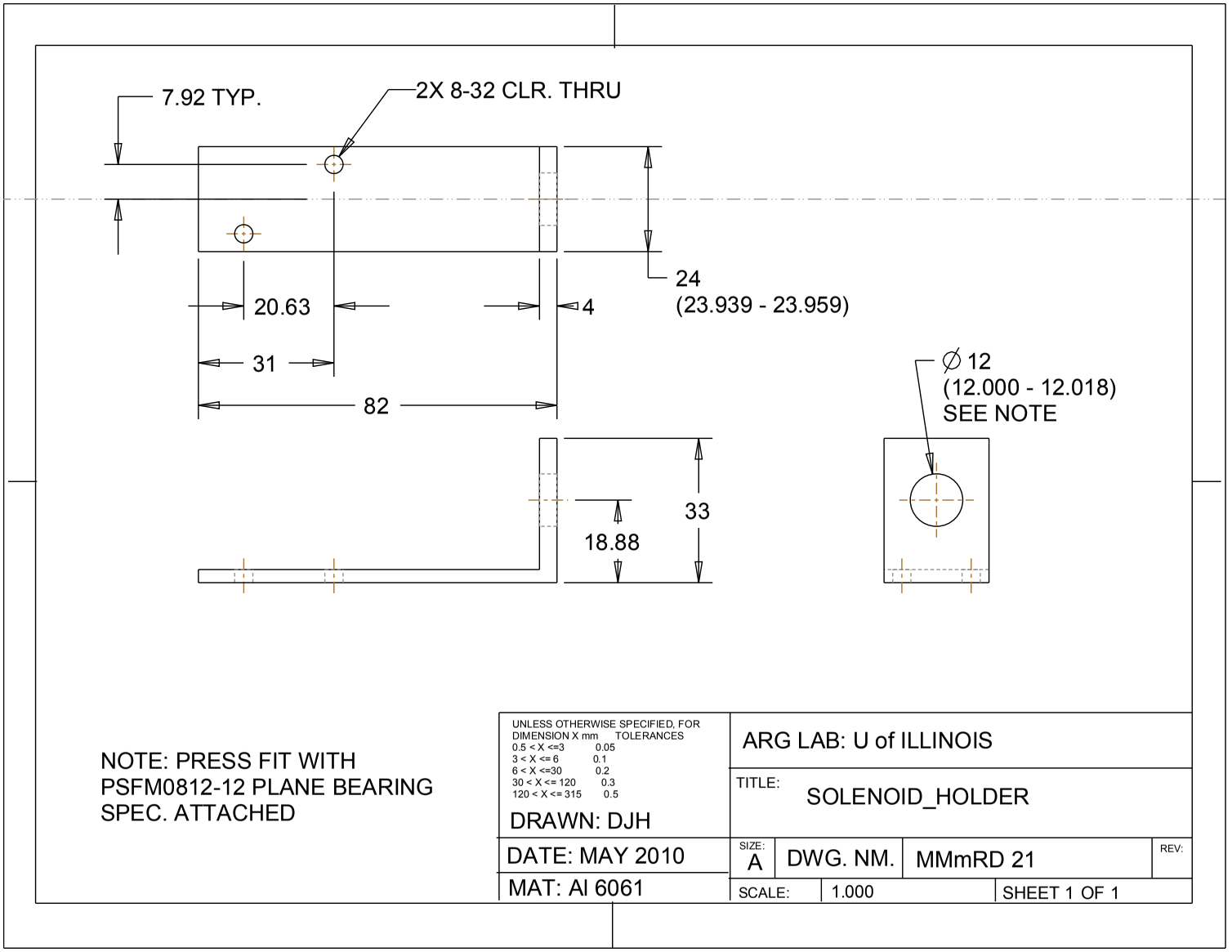
SIZE:	DWG. NM.	REV:
A	MMmRD 17	

SCALE: 2.000 SHEET 1 OF 1


2X ∅4 CLR. THRU
4
10
14
3.88
38.12
R13.25
16.75
21.5
22
4.75
21
42
M4 THD. 6 DEEP
M4 THD. 12 DEEP
4
UNLESS OTHERWISE SPECIFIED, FOR
DIMENSION X mm TOLERANCES
0.5 < X <=3 0.05
3 < X <= 6 0.1
6 < X <=30 0.2
30 < X <= 120 0.3
120 < X <= 315 0.5
DRAWN: DJH
DATE: MAY 2010
MAT: Al 6061
ARG LAB: U of ILLINOIS
TITLE:
MOTOR_BRACKET_FIXED_V3
SIZE:
A
DWG. NM.
MMmRD 18
REV:
SCALE:
2.000
SHEET 1 OF 1

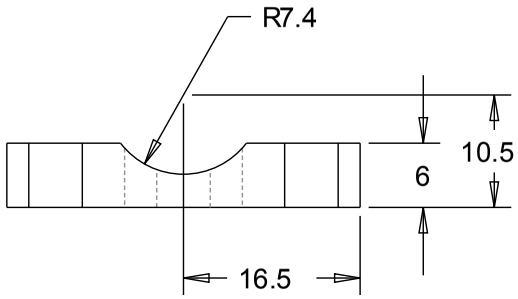

R13.25
R21.5
0.5
4.61
2X M4 THD. THRU
42
8
34.24
UNLESS OTHERWISE SPECIFIED, FOR
DIMENSION X mm TOLERANCES
0.5 < X <=3 0.05
3 < X <= 6 0.1
6 < X <=30 0.2
30 < X <= 120 0.3
120 < X <= 315 0.5
DRAWN: DJH
DATE: MAY 2010
MAT: Al 6061
ARG LAB: U of ILLINOIS
TITLE:
MOTOR_BRACKET_CLAMP_V3
SIZE:
A
DWG. NM.
MMmRD 19
REV:
SCALE:
2.000
SHEET 1 OF 1

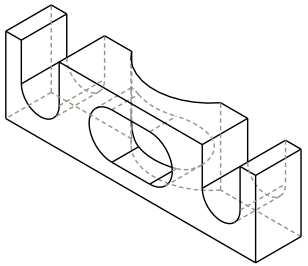

∅ 3.2512
(3.124 - 3.378)
[.1230 - .133]
∅ 8
(8.024 - 8.028)
[.3159 - .3161]
∅ 11
4.63
6
10
11
28.5
UNLESS OTHERWISE SPECIFIED, FOR
DIMENSION X mm TOLERANCES
0.5 < X <=3 0.05
3 < X <= 6 0.1
6 < X <=30 0.2
30 < X <= 120 0.3
120 < X <= 315 0.5
DRAWN: DJH
DATE: NOV 2010
MAT: Al 6061
ARG LAB: U of ILLINOIS
TITLE:
LOCKING_PIN
SIZE:
A
DWG. NM.
MMmRD 20
REV:
B
SCALE:
4.000
SHEET 1 OF 1

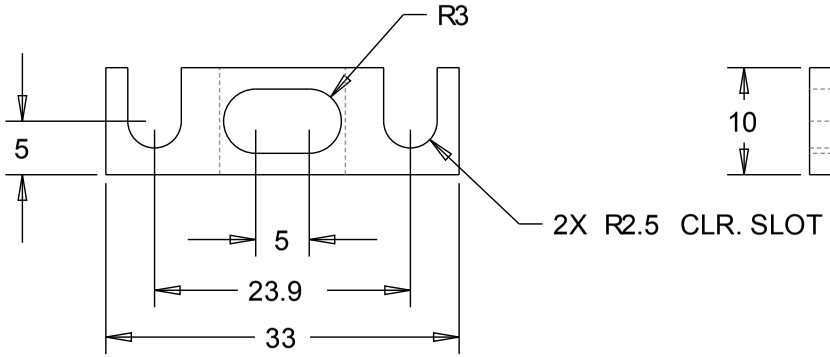


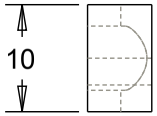
NOTE: PRESS FIT WITH
PSFM0812-12 PLANE BEARING
SPEC. ATTACHED

UNLESS OTHERWISE SPECIFIED, FOR DIMENSION X mm TOLERANCES 0.5 < X <=3 0.05 3 < X <= 6 0.1 6 < X <=30 0.2 30 < X <= 120 0.3 120 < X <= 315 0.5	ARG LAB: U of ILLINOIS
DRAWN: DJH	TITLE: SOLENOID_HOLDER
DATE: MAY 2010	SIZE: A / DWG. NM. MMmRD 21 / REV:
MAT: Al 6061	SCALE: 1.000 / SHEET 1 OF 1

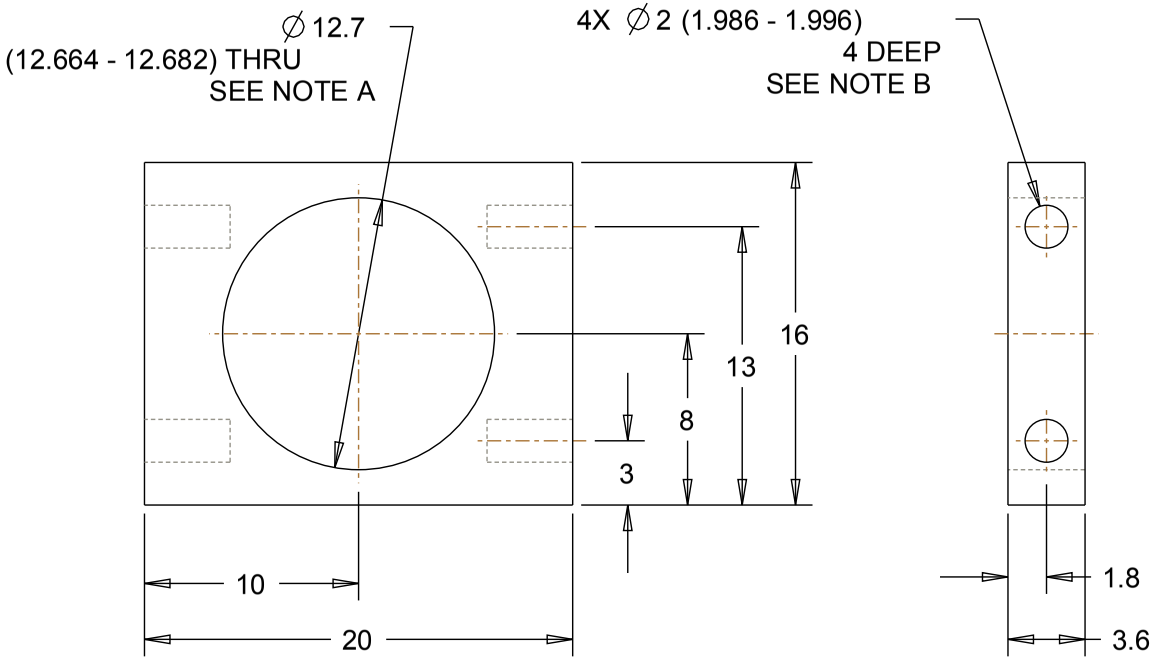








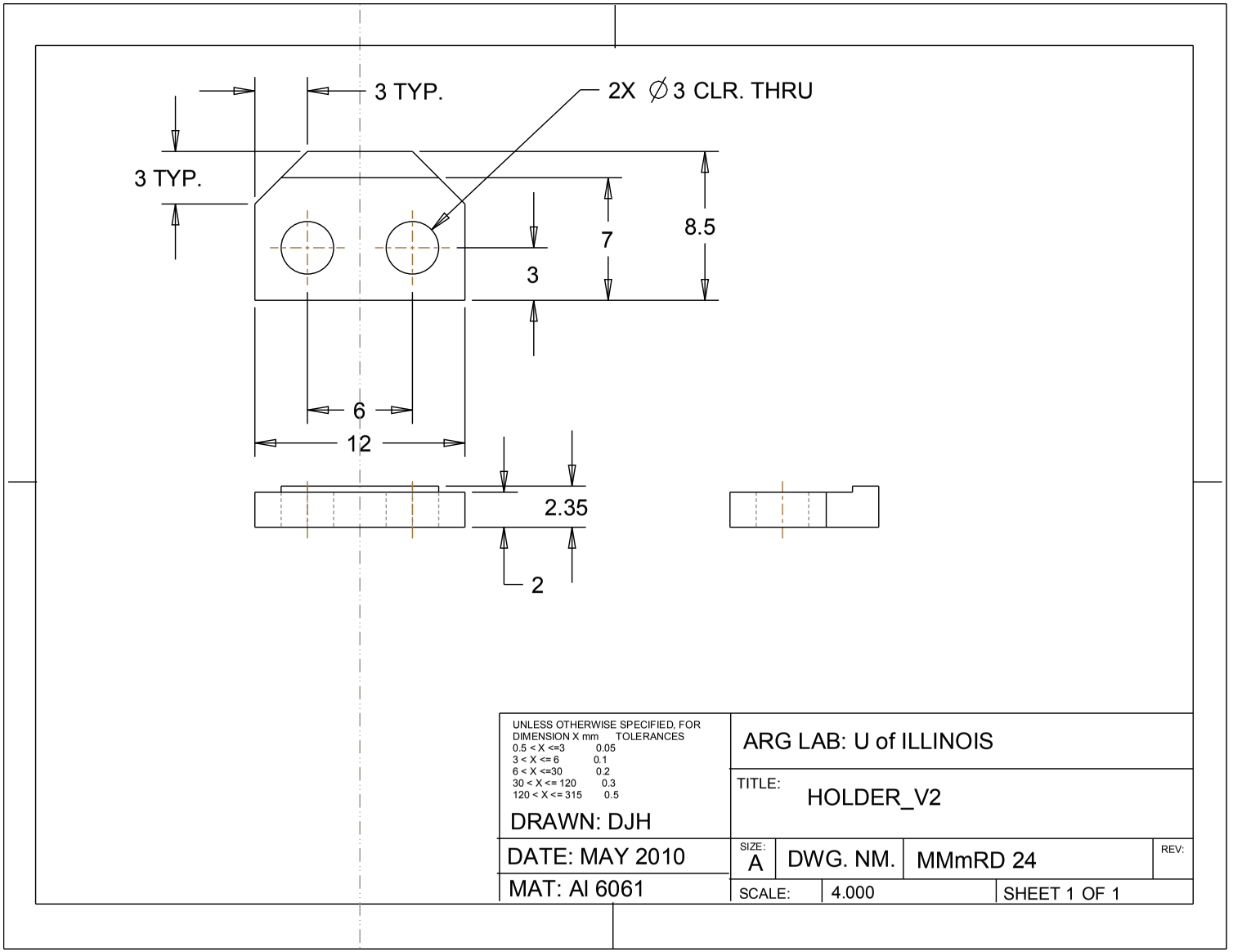
UNLESS OTHERWISE SPECIFIED, FOR DIMENSION X mm TOLERANCES 0.5 < X <=3 0.05 3 < X <= 6 0.1 6 < X <=30 0.2 30 < X <= 120 0.3 120 < X <= 315 0.5	ARG LAB: U of ILLINOIS
DRAWN: DJH	TITLE: SYRINGE_CLAMP_FRONT_V4
DATE: MAY 2010	SIZE: A / DWG. NM. MMmRD 22 / REV:
MAT: Al 6061	SCALE: 2.000 / SHEET 1 OF 1

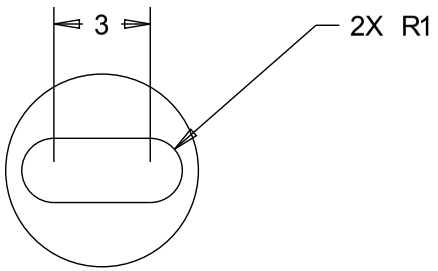


NOTE A: FOR PRESS FIT WITH 0.5in BALL BEARING SPEC. ATTACHED

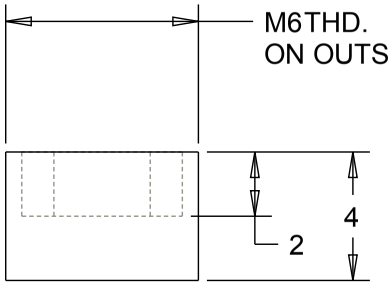
NOTE B: FOR PRESS FIT WITH ∅2mm DOWEL PIN SPEC. ATTACHED

UNLESS OTHERWISE SPECIFIED, FOR DIMENSION X mm TOLERANCES			
0.5 < X <=3	0.05		
3 < X <= 6	0.1		
6 < X <=30	0.2		
30 < X <= 120	0.3		
120 < X <= 315	0.5		
DRAWN: DJH		ARG LAB: U of ILLINOIS	
		TITLE: BEARING_HOLDER	
DATE: MAY 2010		SIZE: A	DWG. NM. MMmRD 23	REV:
MAT: Al 6061		SCALE: 4.000	SHEET 1 OF 1

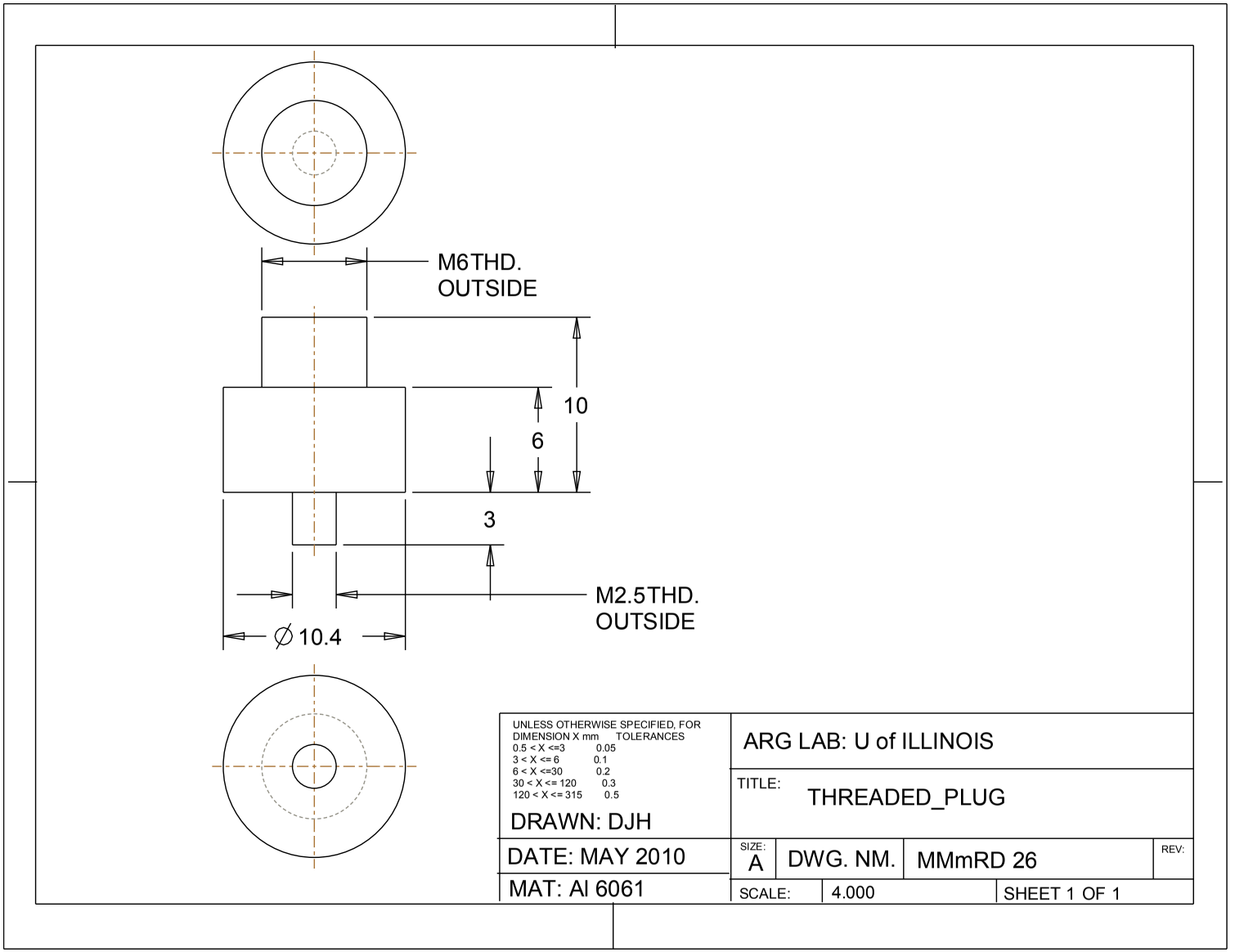

3 TYP.
2X ∅3 CLR. THRU
3 TYP.
7
8.5
3
6
12
2.35
2
UNLESS OTHERWISE SPECIFIED, FOR
DIMENSION X mm TOLERANCES
0.5 < X <=3 0.05
3 < X <= 6 0.1
6 < X <=30 0.2
30 < X <= 120 0.3
120 < X <= 315 0.5
DRAWN: DJH
DATE: MAY 2010
MAT: Al 6061
ARG LAB: U of ILLINOIS
TITLE:
HOLDER_V2
SIZE:
A
DWG. NM.
MMmRD 24
REV:
SCALE:
4.000
SHEET 1 OF 1

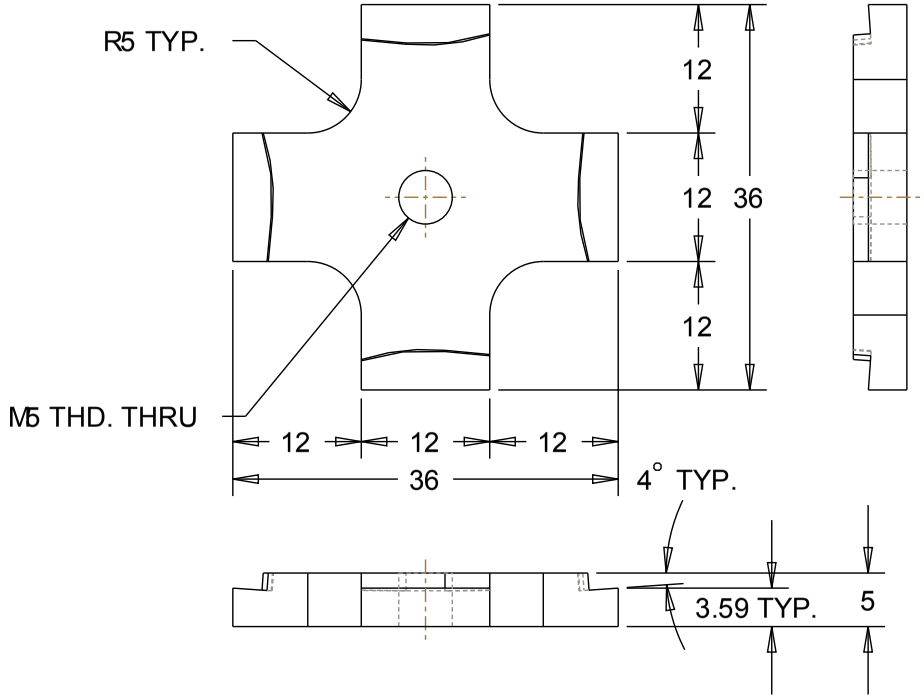


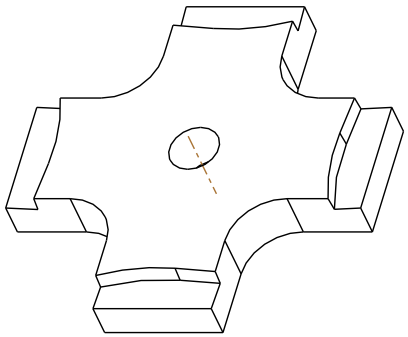
SLOT DIMENSIONS NOT CRITICAL
FOR USE WITH FLAT
SCREW DRIVER



UNLESS OTHERWISE SPECIFIED, FOR DIMENSION X mm TOLERANCES 0.5 < X <=3 0.05 3 < X <= 6 0.1 6 < X <=30 0.2 30 < X <= 120 0.3 120 < X <= 315 0.5	ARG LAB: U of ILLINOIS			
DRAWN: DJH	TITLE: SIMPLE_PLUG_V2			
DATE: MAY 2010	SIZE: A	DWG. NM.	MMmRD 25	REV:
MAT: Al 6061	SCALE:	6.000	SHEET 1 OF 1	


M6THD.
OUTSIDE
10
6
3
M2.5THD.
OUTSIDE
∅10.4
UNLESS OTHERWISE SPECIFIED, FOR
DIMENSION X mm TOLERANCES
0.5 < X <=3 0.05
3 < X <= 6 0.1
6 < X <=30 0.2
30 < X <= 120 0.3
120 < X <= 315 0.5
DRAWN: DJH
DATE: MAY 2010
MAT: Al 6061
ARG LAB: U of ILLINOIS
TITLE:
THREADED_PLUG
SIZE:
A
DWG. NM.
MMmRD 26
REV:
SCALE:
4.000
SHEET 1 OF 1

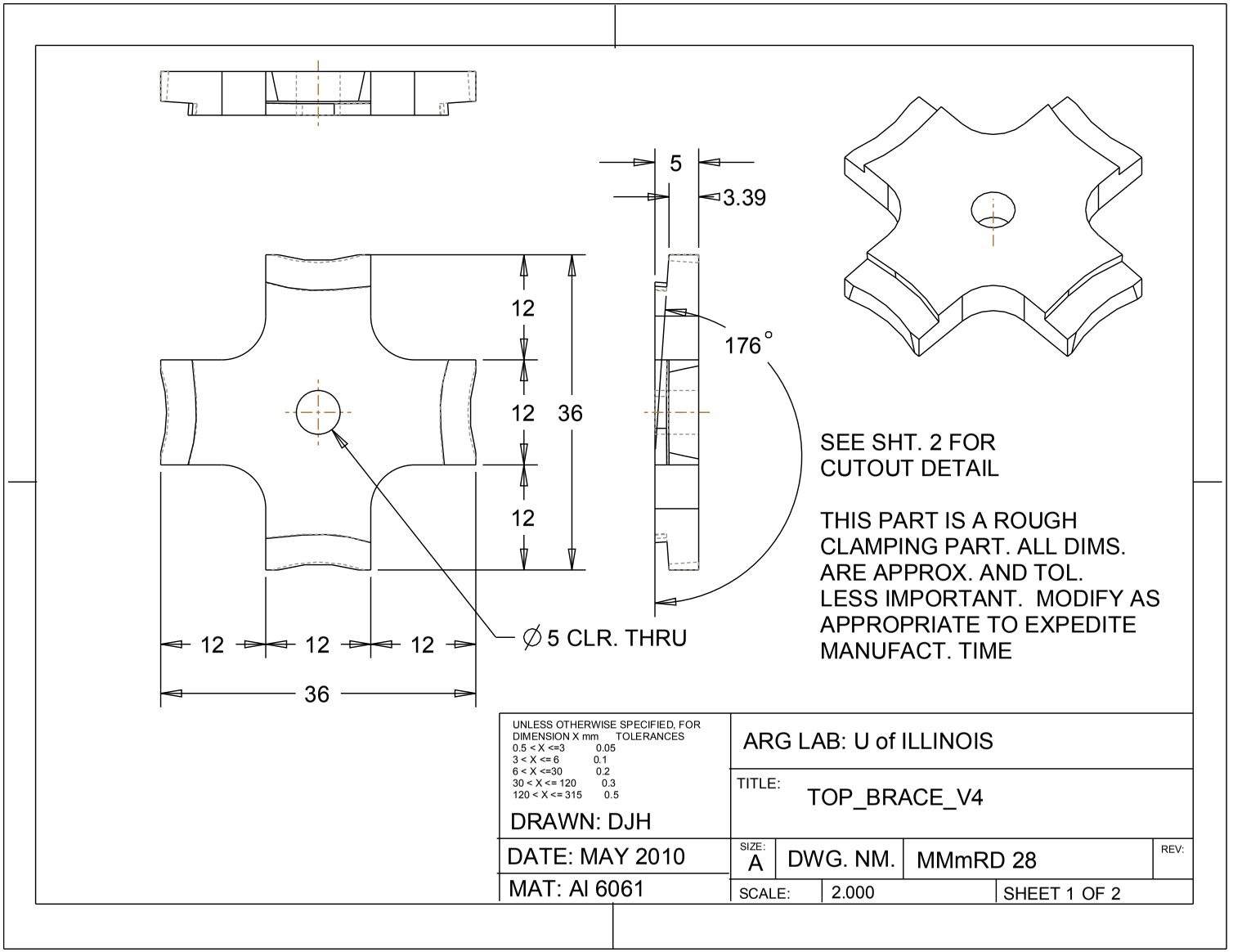


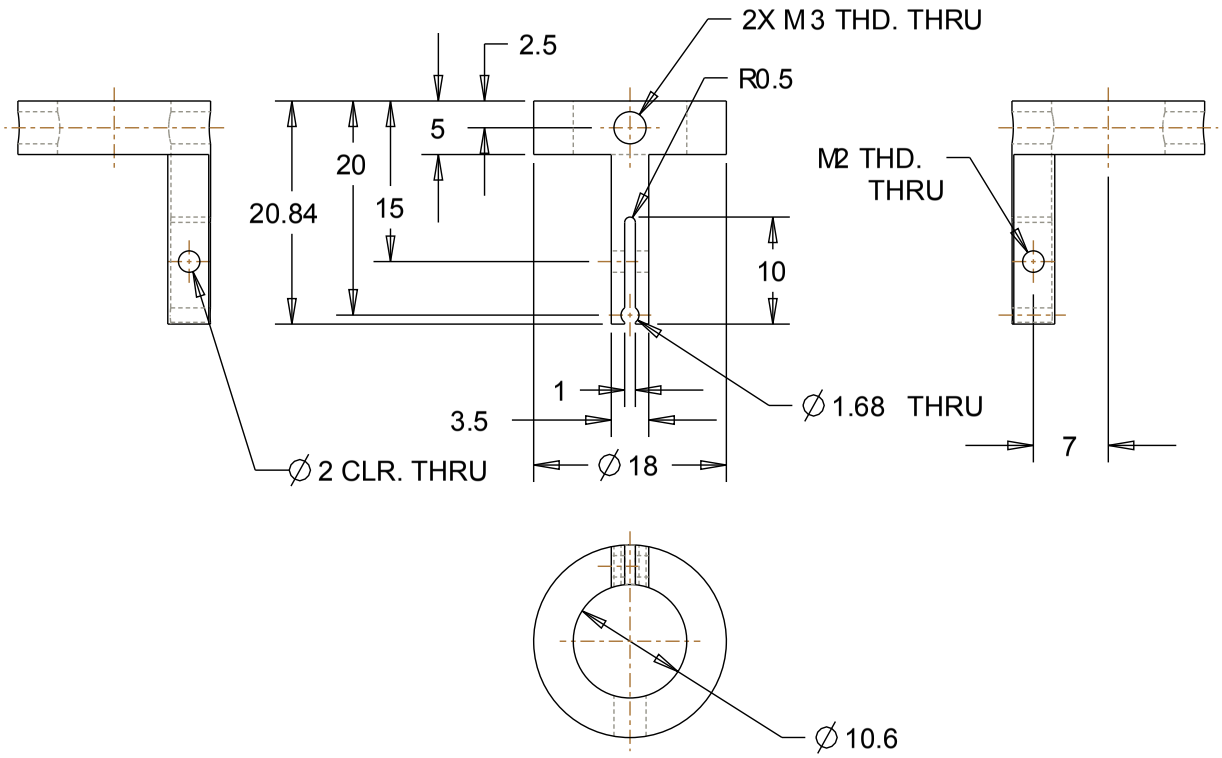


SEE SHT. 2 FOR
CUTOUT DETAIL

THIS PART IS A ROUGH CLAMPING PART. ALL DIMS. ARE APPROX. AND TOL. LESS IMPORTANT. MODIFY AS APPROPRIATE TO EXPEDITE MANUFACT. TIME

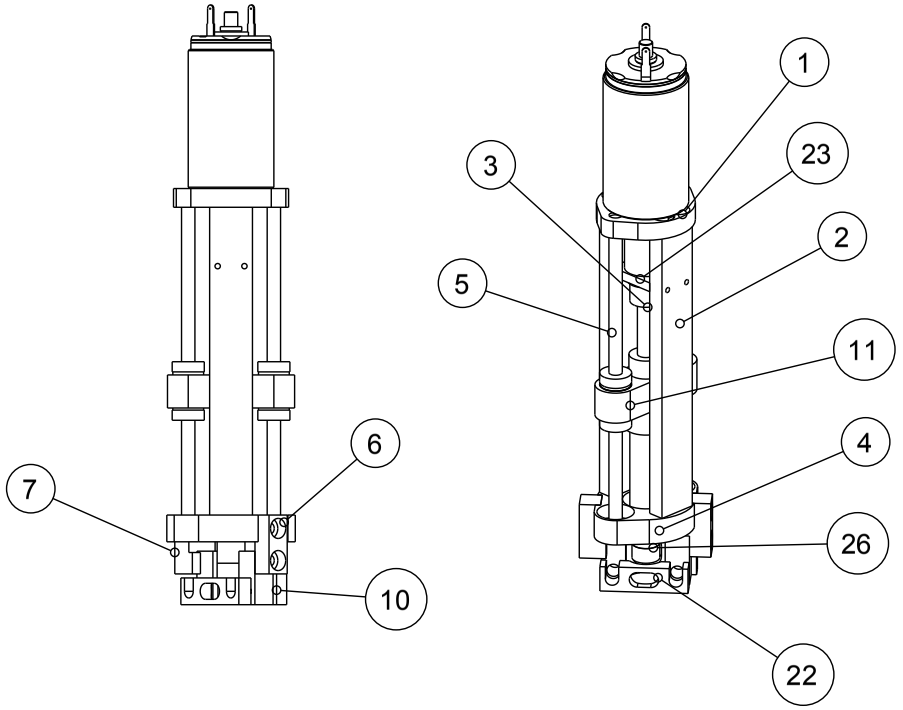
UNLESS OTHERWISE SPECIFIED, FOR DIMENSION X mm TOLERANCES 0.5 < X <=3 0.05 3 < X <= 6 0.1 6 < X <=30 0.2 30 < X <= 120 0.3 120 < X <= 315 0.5	ARG LAB: U of ILLINOIS
DRAWN: DJH	TITLE: BOTTOM_BRACE_V4
DATE: MAY 2010	SIZE: A | DWG. NM. MMmRD 27 | REV:
MAT: Al 6061	SCALE: 2.000 | SHEET 1 OF 2


5
3.39
12
12
12
36
176°
∅ 5 CLR. THRU
12
12
12
36
SEE SHT. 2 FOR
CUTOUT DETAIL
THIS PART IS A ROUGH
CLAMPING PART. ALL DIMS.
ARE APPROX. AND TOL.
LESS IMPORTANT. MODIFY AS
APPROPRIATE TO EXPEDITE
MANUFACT. TIME
UNLESS OTHERWISE SPECIFIED, FOR
DIMENSION X mm TOLERANCES
0.5 < X <=3 0.05
3 < X <= 6 0.1
6 < X <=30 0.2
30 < X <= 120 0.3
120 < X <= 315 0.5
DRAWN: DJH
DATE: MAY 2010
MAT: Al 6061
ARG LAB: U of ILLINOIS
TITLE:
TOP_BRACE_V4
SIZE:
A
DWG. NM.
MMmRD 28
REV:
SCALE:
2.000
SHEET 1 OF 2

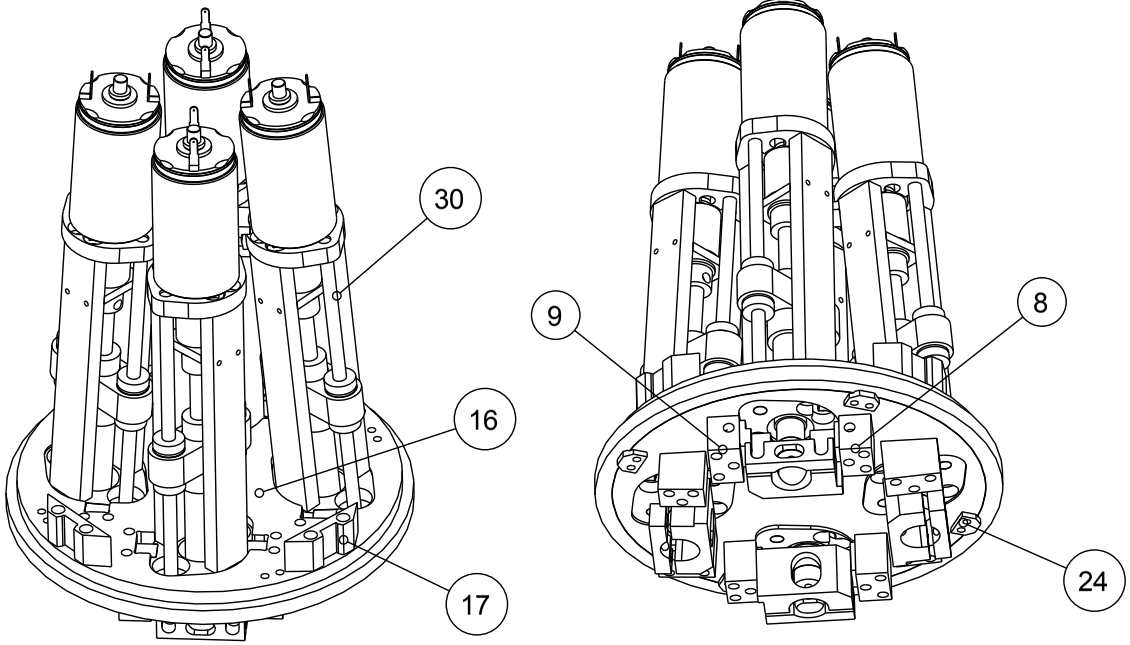


FAB. WITH ANY CONDUCTIVE MATERIAL.

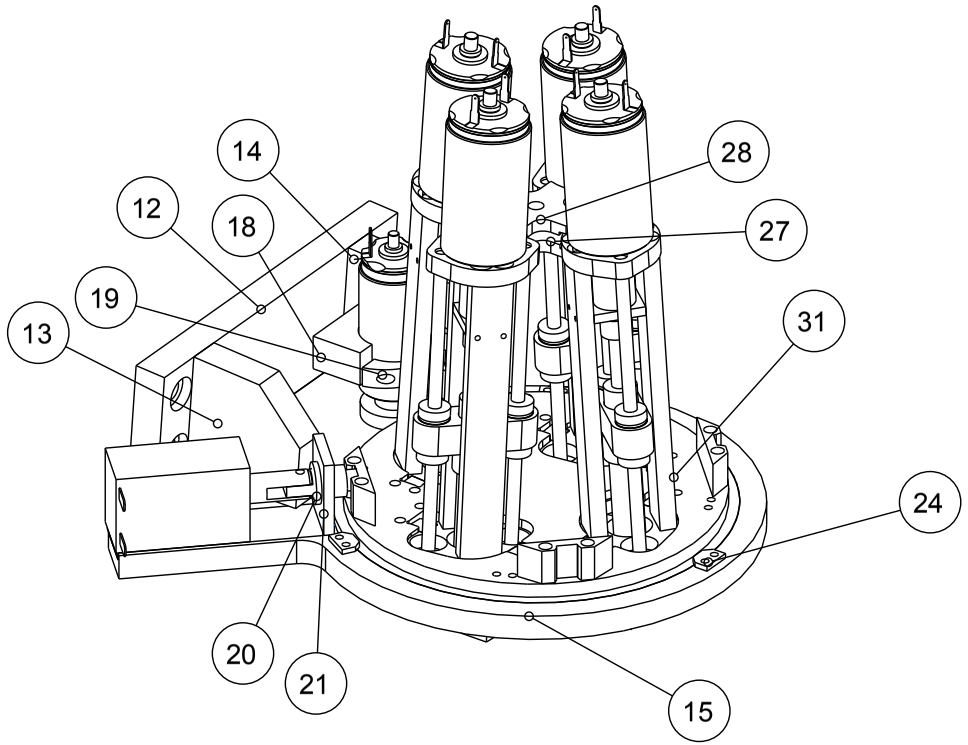
UNLESS OTHERWISE SPECIFIED, FOR DIMENSION X mm TOLERANCES 0.5 < X <=3 0.05 3 < X <= 6 0.1 6 < X <=30 0.2 30 < X <= 120 0.3 120 < X <= 315 0.5	ARG LAB: U of ILLINOIS
DRAWN: DJH	TITLE: CONTACT_V1
DATE: MAY 2010	SIZE: A | DWG. NM. MMmRD 29 | REV:
MAT: Al 6061	SCALE: 2.000 | SHEET 1 OF 1



UNLESS OTHERWISE SPECIFIED, FOR DIMENSION X mm TOLERANCES 0.5 < X <=3 0.05 3 < X <= 6 0.1 6 < X <=30 0.2 30 < X <= 120 0.3 120 < X <= 315 0.5	ARG LAB: U of ILLINOIS			
DRAWN: DJH	TITLE: EXTRUSION_ASSEMBLY			
DATE: MAY 2010	SIZE: A	DWG. NM.	MMmRD 30	REV:
MAT: Al 6061	SCALE:	0.500	SHEET 1 OF 1	



UNLESS OTHERWISE SPECIFIED, FOR DIMENSION X mm TOLERANCES 0.5 < X <=3 0.05 3 < X <= 6 0.1 6 < X <=30 0.2 30 < X <= 120 0.3 120 < X <= 315 0.5	ARG LAB: U of ILLINOIS			
DRAWN: DJH	TITLE: ROTARY			
DATE: MAY 2010	SIZE: A	DWG. NM.	MMmRD 31	REV:
MAT: Al 6061	SCALE:	0.500	SHEET 1 OF 1	



UNLESS OTHERWISE SPECIFIED, FOR DIMENSION X mm TOLERANCES 0.5 < X <=3 0.05 3 < X <= 6 0.1 6 < X <=30 0.2 30 < X <= 120 0.3 120 < X <= 315 0.5	ARG LAB: U of ILLINOIS			
DRAWN: DJH	TITLE: COMPLETE			
DATE: MAY 2010	SIZE: A	DWG. NM.	MMmRD 32	REV:
MAT: Al 6061	SCALE:	0.500	SHEET 1 OF 2	

ITEM	QUANTITY	MAT.	DWG. TYPE
MMmRD 01	4	ALUM.	PART
MMmRD 02	8	ALUM.	PART
MMmRD 03	4	ALUM.	PART
MMmRD 04	4	ALUM.	PART
MMmRD 05	8	ALUM.	PART
MMmRD 06	4	ALUM.	PART
MMmRD 07	4	ALUM.	PART
MMmRD 08	4	ALUM.	PART
MMmRD 09	4	ALUM.	PART
MMmRD 10	4	ALUM.	PART
MMmRD 11	4	ALUM.	PART
MMmRD 12	1	ALUM.	PART
MMmRD 13	1	ALUM.	PART
MMmRD 14	1	ALUM.	PART
MMmRD 15	1	ALUM.	PART
MMmRD 16	1	ALUM.	PART
MMmRD 17	4	ALUM.	PART
MMmRD 18	1	ALUM.	PART
MMmRD 19	1	ALUM.	PART
MMmRD 20	1	ALUM.	PART
MMmRD 21	1	ALUM.	PART
MMmRD 22	4	ALUM.	PART
MMmRD 23	4	ALUM.	PART
MMmRD 24	7	ALUM.	PART
MMmRD 25	4	ALUM.	PART
MMmRD 26	4	ALUM.	PART
MMmRD 27	1	ALUM.	PART
MMmRD 28	1	ALUM.	PART
MMmRD 29	4	ALUM.	PART
MMmRD 30	4	-	ASSEMBLY
MMmRD 31	1	-	ASSEMBLY
MMmRD 32	1	-	ASSEMBLY

UNLESS OTHERWISE SPECIFIED, FOR
DIMENSION X mm TOLERANCES
0.5 < X <=3 0.05
3 < X <= 6 0.1
6 < X <=30 0.2
30 < X <= 120 0.3
120 < X <= 315 0.5

DRAWN: DJH

DATE: MAY 2010

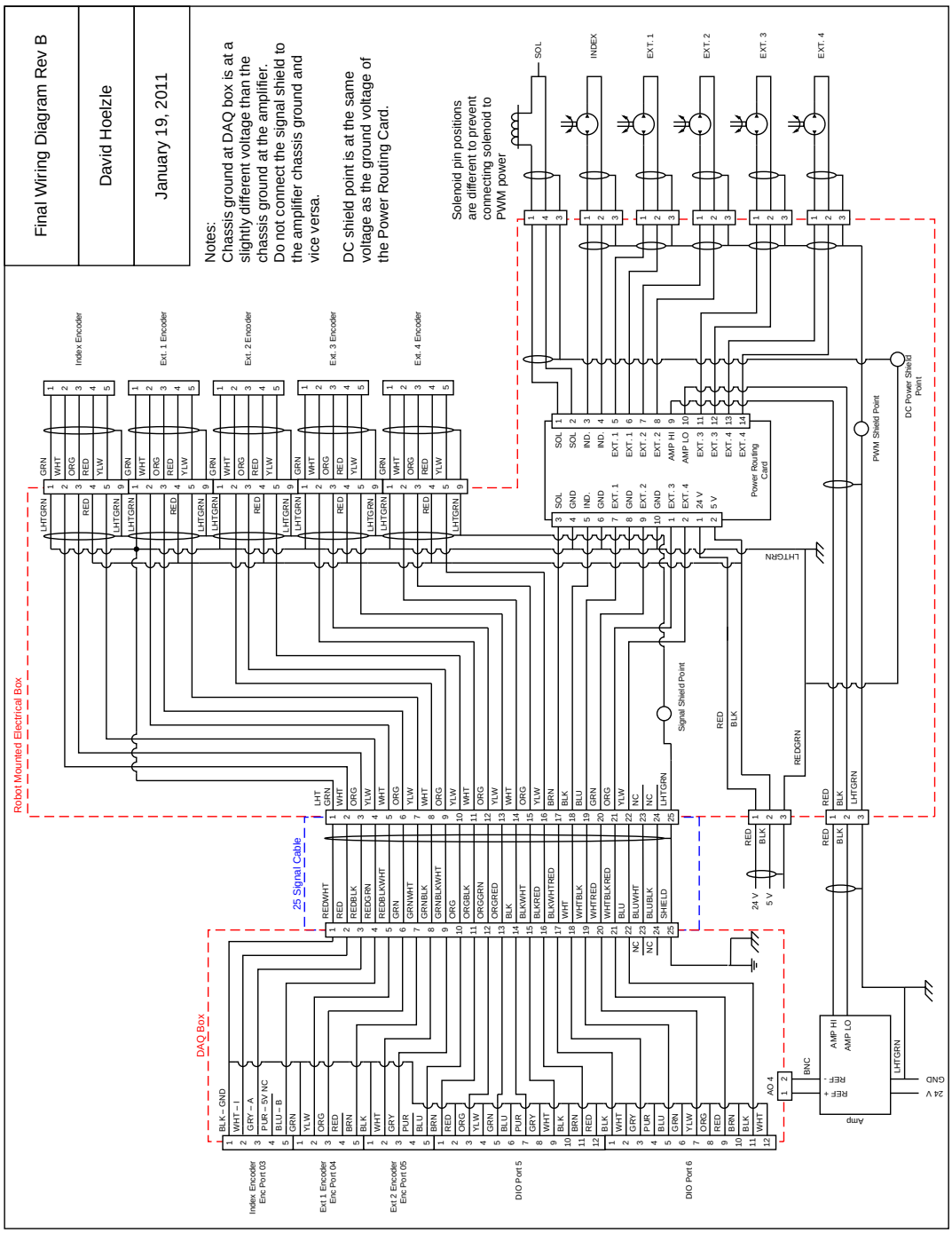
MAT: Al 6061

ARG LAB: U of ILLINOIS

TITLE: BILL_OF_MATERIALS

SIZE: A | DWG. NM. MMmRD 33 | REV:

SCALE: 1.000 | SHEET 1 OF 1

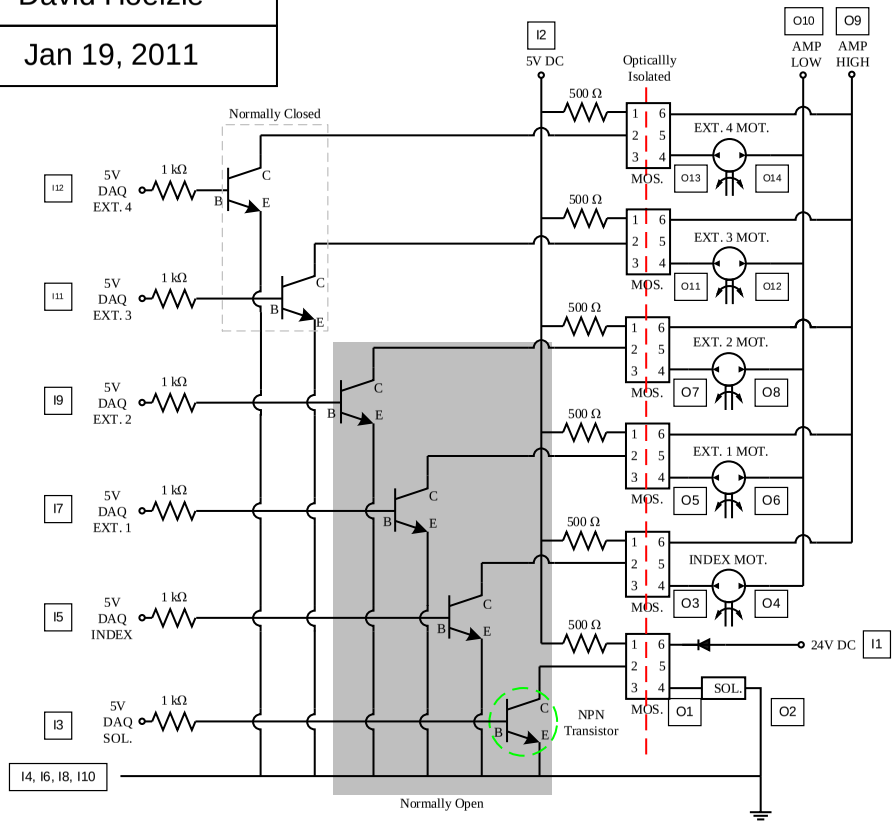

Final Wiring Diagram Rev B
David Hoelzle
January 19, 2011
Notes:
Chassis ground at DAQ box is at a slightly different voltage than the chassis ground at the amplifier. Do not connect the signal shield to the amplifier chassis ground and vice versa.
DC shield point is at the same voltage as the ground voltage of the Power Routing Card.
Solenoid pin positions are different to prevent connecting solenoid to PWM power
Robot Mounted Electrical Box
DAQ Box
25 Signal Cable
Power Routing Card
Signal Shield Point
PWM Shield Point
DC Power Shield Point
Amp
Index Encoder
Ext. 1 Encoder
Ext. 2 Encoder
Ext. 3 Encoder
Ext. 4 Encoder
SOL
INDEX
EXT. 1
EXT. 2
EXT. 3
EXT. 4
Index Encoder Enc Port 03
Ext 1 Encoder Enc Port 04
Ext 2 Encoder Enc Port 05
DIO Port 5
DIO Port 6

Power Routing Card
Rev B

David Hoelzle

Jan 19, 2011

Notes:
Transistor: MPSA06
Motor MOSFETS: PVG612A
Solenoid MOSFETS: PVG612



Appendix F

Computer Code

Appendix F contains the complete set of `.m` files used to run BTILC on the μRD system. Original source files can be obtained by emailing David Hoelzle (hoelzle2@gmail.com) and requesting them. Additionally, certain `.mat` and `.mdl` files are too lengthy to include here or are not suitable for accurate dissemination of information through print and will have to be requested.

F.1 Task 1: Identification of system dynamics

Video_Processing_CF.m

```
% Syntax [TimeRef, Vdot, Error, minx, maxx, miny]=Video_Processing_
CF(filename,Check_Point)
function [TimeRef, Vdot1kHz, Error, min_x, max_x, min_y] = Video_
Processing_CF(filename, Check_Point, Iteration, min_x, max_x, min_y);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%
% This code takes the arguments of a video filename and the frame
% at which to check the video to specify a region of interest and
% returns vectors for time, a 1 kHz Vdot signal, a 1 kHz Error
% signal, and video dependent frequency signal for rod width
% Note: Camera mounted right side up
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%

% Defined Variables used as test variables in testing
% filename = 'Trial 1';
% % Start_Max = 1;      % Frames
% Check_Point = 25;    % Frames
% Iteration = 1;
% min_x = 1;
% max_x = 20;
% min_y = 200;
```

```
% Initialize movie
temp_mov = VideoReader(filename);

% Defined constants
HorCalibrate = 554.75/7;   %Horpixel/mm
VertCalibrate = 216.39/4;  %Vertpixel/mm
PI = 3.14159;              % Pi
vel = 3;                   % mm/s
h = 0.42;                  % mm
NozSpot = 188.96;          % distance in vertical pixels to nozzle
 center, changes at any movement of camera
FrameJump = 3;             % Frequency of Frames uploaded
FramesPerSecond = 29.97;    % Will be dependent on specific video
feed; change accordingly

load ReferenceSignal2.mat         % Pulse type input

clear Frame; clear mov; clear Ibw; clear Image_Gray; clear Seg_Image;
clear rod_width; clear TimeTemp; clear VCorrection; clear Time;

% Image based variables
ROIDist = vel*FrameJump/FramesPerSecond;           % mmV/Frame
DoubleROIPix = ROIDist*VertCalibrate;       % Vpix/Frame double format
ROIPix = round(ROIDist*VertCalibrate);      % Vpix/Frame
Pixel2Time = DoubleROIPix*FramesPerSecond/FrameJump; % Vpix/s

% This section displays 9 sequential images from the video file for the
% user to select the image where movement starts

manual = 0;                  % Default manual start finder
happy = 0;                   % Sometimes the region of interest is
 improperly
                             % selected, happy variable allows the
                             % code to be rerun with a different ROI
                             % without interrupting the higher level
                             % code ILC_Implement_MI.
while (happy ~= 1)

    if (manual ~= 1)
        [Start_Frame] = Motion(temp_mov, 1, 12, 60);
        % Runs motion detection function
    end

    answer = 0;
    while ((answer ~= 1) && (manual == 1))
        Start_Guess = input('Guess Moving Frame: ');
        figure (1)
        for i = 1:9         % Plot 9 images
            subplot(3,3,i)
            imshow(read(temp_mov,Start_Guess+i-1))
            xlabel (Start_Guess+i-1)
        end
        answer = input('Correct Range? (1 = Yes, 0 = No): ');
        if (answer == 1)
            Start_Frame = input('Starting Image :');
        end
        close(1)
    end

    % Line Start
    Start_Line = round(15 * FramesPerSecond) + Start_Frame;
```

```
    % Line End
    End_Line = round(28 * FramesPerSecond) + Start_Frame;

    % reading one frame at a time and storing it in to array
    dummy = 0;
    for i=Start_Line:FrameJump:End_Line;    % Code only extracts
    data from every 3rd image
        dummy = dummy + 1;
%         mov=aviread(filename,i);
        Frame(:,:,:,dummy) = read(temp_mov,i);

    end

    clear xi; clear yi;

    % Select Region of interest

%     % imshow(Frame(:,:,:,Check_Point))
%
if (Iteration == 1)
    figure(1)
    [BW,xi,yi] = roipoly(Frame(:,:,:,Check_Point));
    % User selects region
    min_x = int16(min(xi));
    % of interest around
    max_x = int16(max(xi));
    % nozzle tip
    min_y = int16(min(yi));
    max_y = int16(max(yi));
    % Use this one for time shift calc
end

%     [BW,xi,yi] = roipoly(Frame(:,:,:,Check_Point));
%     min_x = 288;      % constants used for testing
%     max_x = 397;
%     min_y = 178;
%     max_y = 234;

    close all

    clear Seg_Image; clear Image_Gray; clear Ibw;

    rod_width(1:2) = 0;   % Initializations
    Time(1:2) = [-1,(double(Start_Line) - Start_Frame)/FramesPer
    Second - 15 - (-NozSpot+ROIPix-1+double(min_y))/double(Pixel
    2Time)-.001 + .3244/5];    % Makes rod_width at t0 to tstart
     = 0
    int = 0;

    % Final run through image for calculations
    for i = 1:round((End_Line - Start_Line)/FrameJump)

        Seg_Image(:,:,:,i) = Frame(min_y:min_y+ROIPix-1,min_x:max_x
        ,:,i);   % segment image to ROI
        Image_Gray(:,:,i) = .2989*Seg_Image(:,:,1,i)+.5870*Seg_Image
        (:,:,2,i)+.1140*Seg_Image(:,:,3,i); %convert to grayscale
        from standard values
%         level = graythresh(Image_Gray(:,:,i))-.05;
%Find threshold level
%         if (level < 0 )
 % level between 0 and 1
%             level = 0;
```

```
%         end
        level = 0.35;
        % empiracally determined level
        Ibw(:,:,i) = im2bw(Image_Gray(:,:,i),level);
        %Convert to BW binary image

        Size = length(rod_width);
         % For array indexing

        for j = 1:ROIPix            % Scan through image
            int = int + 1;
            rod_width(Size+j) = (1/HorCalibrate)*sum(Ibw(ROIPix
            -j+1,:,i));    %Calculates rod width in mm by summing
            along rows

            R = rod_width(Size+j)/2;
            if (R <= 0.5 * h)
                VCorrection(Size+j) = (PI)* R^2;      % If diameter
                 is less than fly height
            else
                theta = asin(.5*h/R);                   % If diameter
                is greater than fly height
                VCorrection(Size+j) = (2*theta*R^2+0.5*h^2*(1/tan
                (theta)));
            end

            % Calculation of time vector is tricky.  This equation
            % shifts time based on a equation that includes the
            % pixel2time ratio, the starting frame, and the time
            % shift brought on by the distance between the nozzle
            % tip and the ROI location
            Time(Size+j) = (int-1)*1/double(Pixel2Time) - 15 +
            (double(Start_Line) - Start_Frame)/FramesPerSecond -
            (-NozSpot+ROIPix-1+double(min_y))/double(Pixel2Time)
            + .3244/5; %Time Calc starting at 0, Added box location
             part, plus deceleration losses

        end

    end

    % the variable frequency VCorrection signal must be modified
    % to a 1kHz signal.  Here the interp1 function linearly inter
    % polates the variable frequency signal to provide evenly
    % spaced data points at 1kHz

    Time = sort(Time);
    % Prevents interp1 errors
    Vdot1kHz = interp1(Time,VCorrection,TimeRef,'linear');
    % mm^3/s at 1kHz: dim(TimeRef) < dim(Time)
    for n = 1:length(Vdot1kHz)
        if (isnan(Vdot1kHz(n)))
        % NaN's corrupt data.  Turn all Nan's to zero
            Vdot1kHz(n) = mean(Vdot1kHz(n-10:n-1));
        end
    end
    Error = Reference - Vdot1kHz(1:length(Reference));
    % Error signal.  mm^3/s

    %Plots
    figure(1)
```

```
    plot(Time,VCorrection,'o-', TimeRef, Vdot1kHz,'.-')
    xlabel ('Time (s)')
    ylabel ('Volumetric Flow Rate (mm^3/mm)')
    xlim ([0 14])

    figure (2)
    plot(TimeRef, Reference,'k-', TimeRef, Vdot1kHz, 'b.-',TimeRef,
    Error,'r.-')
    xlabel ('Time (s)')
    ylabel ('Output (mm^3/mm)')
    legend ('Reference', 'Q', 'Error')

    figure(3)
    plot(Time, rod_width,'.-')
    xlabel ('Time (s)')
    ylabel ('Rod Width (mm)')
    xlim ([0 14])

    happy = input('Happy with results? (0 = No, 1 = Yes): ');
    % Assures ROI and other things were properly chosen
    if (happy ~= 1)
        manual = input('Manual Start Finder? (0 = No, 1 = Yes): ');
        % Asks whether to use a manual or automatic Start Finder
    end
end
```

Motion.m

```
function [Start_Frame] = Motion(mov, thresh, minframe, maxframe)

clear Frame_Gray_new; clear Frame_Gray_old; clear image;
% filename = 'Attempt20NMPTrial 1.avi';

% minframe = 10;
% maxframe = 40;
% thresh = 1;
framenum = minframe;
movement = 0;

image = read(mov,framenum);
Frame_Gray_old = .2989*image(:,:,1) + .5870*image(:,:,2) +
 .1140*image(:,:,3);

while ((framenum <= maxframe) && (movement ~= 1))
    image = read(mov, framenum+1);
    Frame_Gray_new = .2989*image(:,:,1) + .5870*image(:,:,2) +
     .1140*image(:,:,3);
    Diff = Frame_Gray_new - Frame_Gray_old;
    MeanDiff = mean(mean(abs(Diff)));
    Data(framenum + 1 - minframe) = MeanDiff;
    if (MeanDiff > thresh)
        movement = 1;
        Start_Frame = framenum
    end

Frame_Gray_old = Frame_Gray_new;
    framenum = framenum + 1;

end
```

```
figure (99)
plot(minframe:framenum-1, Data, 'k-o', [minframe,framenum-1],
[thresh, thresh], 'k--')
xlabel ('Frame Number')
ylabel ('Frame Difference')
legend ('Difference', 'Threshold', 2)
```

Compile.m

```
% Compile and plot nominal response

clear
load ReferenceSignal2
load Data;

for i = 6:10
    if i == 1
        min_x = 1; max_x = 1; min_y = 1;
    end
    [Time, Vdot(i,:), Error(i,:), min_x, max_x, min_y] = Video_
    Processing_    CF(['Trial ' int2str(i) '.avi'], 25, i, min_x,
    max_x, min_y);
    save Data;
end


ErrorMean = mean(Error,1);
VdotMean = mean(Vdot, 1);
% rod_widthMean = mean(RW,1);

save Data;
```

DataCompile.m

```
% Compile and plot nominal response

% clear
load Data
load ReferenceSignal2

ErrorMean = mean(Error,1);
VdotMean = mean(Vdot, 1);
% rod_widthMean = mean(RW,1);

% figure (5)
% plot(TimeRef1', rod_widthMean, 'b-')
% xlabel ('Time (s)')
% ylabel ('Rod Width (mm)')

% Simulation

K = 1.04;
tau = 2.7;

sim NomSim

figure (4)
plot(Time, Reference, 'k--', Time, VdotMean', 'b-', SimTime, Output,
 'r-.')
```

```
xlabel ('Time (s)')
ylabel ('Normed Flowrate (mm^3/mm)')
legend ('Reference','Flowrate','Model')
xlim([0 14])
```

F.2 Task 2: Identification of basis task input signals (basis signals)

ILC_Implement_MI_TV.m

```
% Syntax: [Time, u, Error, Vdot1kHz, RMS, Max_Error, UCompile] =
ILC_Implement_MI_TV(P, Smoothing Bandwidth, Smoothing Order,
% Start Iteration, Iterations, Material Number);

function [Time, u, ErrorComp, Vdot1kHzComp, RMS, Max_Error] = ILC_
Implement_MI_TV(P, Q_Band, Order, Start_It, Iterate, MatNum);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%

% This function uses error signals from deposition trials to calculate a
% new u signal at every interation to interatively improve deposition
% performance.  The error signal is calculated by calling function
% Video_Processing_CF.m

% Input arguments are P gain, Smoothing Bandwidth, Smoothing Order,
% Start Iteration, number of iterations, Material Number

% Machine notes: Always use acceleration of 250 mm/s^2

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%

% Initial Parameters
% Start_It = 1;       % Starting Iteration in event of crash
% P = .25;              % Proportional Gain of Learning Filter
% Q_Band = 10;     % Hz, Bandwidth of Q-Filter
% Order = 2;          % Filter Order
% Iterate = 2;       % Number of Iterations
% MatNum = 1;

% Nominal Plant Dynamics
K = 1.04;                    % Parameters of nominal system
tau = 2.7;

num = K*[1/100 1];
den = [tau 1];

invsysd = c2d(tf(den,num),.001);

invdnum = cell2mat(invsysd.num);   % inverse discrete plant numerator
invdden = cell2mat(invsysd.den);      % inverse discrete plant denomi
nator
```

```
[b,a] = butter(Order, Q_Band/1000);

load ReferenceSignal.mat           % Nominal pulse-type input  1kHz

if (Start_It > 1)   % Recover Data from .mat file if computer crashed,
 if not starting at 1
    load ('DataSave.mat','Time','Vdot1kHzComp','ErrorComp','RMS',
    'Max_Error','u', 'min_x', 'max_x', 'min_y')
end

u(1,:) = Reference;      % Store 1st u(k) at 1kHz into memory
[ustart, usteady, uend, utail, UCompile] = usplice(u(1,:), 2, 10,
10.01, 14.2, MatNum);
Uodd = UCompile;
save(['UFile' int2str(1) '.mat'],'Uodd')

for j = Start_It:Iterate

    disp (['Hit any key to process Iteration ' int2str(j)]);
    pause;

    filename = ['It' int2str(j) '.avi'];

    if j == 1                                          % 1st iteration
     uses dummy ROI values
        min_x = 1; max_x = 1; min_y = 1;
    end

    [Time, Vdot1kHzComp(j,:), ErrorComp(j,:), min_x, max_x, min_y] =
    Video_Processing_CF(filename, 25, j, min_x, max_x, min_y);
    % Get exp. data

    % Learning algorithm  (Here we use model inversion)
    utemp(j+1,:) = u(j,:) + P*filter(invdnum,invdden,ErrorComp(j,:));

    % Lowpass Q-Filter
    u(j+1,:) = filtfilt(b,a,utemp(j+1,:));

    % TV-Filter
%     u(j+1,:) = TVfilter_test(utemp(j+1,:), filtfunc, SmoothBand,
 Order);

    [ustart, usteady, uend, utail, UCompile] = usplice(u(j+1,:), 2, 10,
     10.01, 14.2, MatNum);

    if (mod(j+1,2) == 0)                                % Need to
    alternate even and odd because S-function needs to be changed
        Ueven = UCompile;
        save(['UFile' int2str(j+1) '.mat'],'Ueven', '-v6')
    else
        Uodd = UCompile;
        save(['UFile' int2str(j+1) '.mat'],'Uodd', '-v6')
    end

    RMS(j) = norm(ErrorComp(j,:))/sqrt(length(ErrorComp(j,:)));
    % Trial stats
    Max_Error(j) = max(abs(ErrorComp(j,:)));

    save DataSave.mat

    figure(4)
    plot(Time,u(j+1,:),'r-', Time, u(j,:),'b-')
    xlabel ('Time (s)')
    ylabel ('u(k) (mm^3/mm)')
```

```
    legend ('j+1','j')
    figure(5)
    plot(1:j, RMS(1:j), 'k-o', 1:j, Max_Error(1:j), 'b--*')
    xlabel('Iteration')
    ylabel ('Error (mm^3/mm)')
    legend ('RMS', 'Max Error')
end
```

Video_Processing_CF.m

```
% Syntax [TimeRef,Vdot, Error,% rod_width]=Video_Processing_CF
(filename,Check_Point)
function [TimeRef, Vdot1kHz, Error, min_x, max_x, min_y] = Video_
Processing_CF(filename, Check_Point, Iteration, min_x, max_x,
min_y);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%

% This code takes the arguments of a video filename and the frame
% at which to check the video to specify a region of interest and
% returns vectors for time, a 1 kHz Vdot signal, a 1 kHz Error
% signal, and video dependent frequency signal for rod width

% Note: Camera right side up

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%


% Defined Variables used as test variables in testing
% filename = ['It' int2str(12) '.avi'];
% Start_Max = 1;      % Frames
% Check_Point = 25;    % Frames
% Iteration = 12;
% min_x = 1; max_x = 100; min_y = 100;

% Initialize movie
temp_mov = VideoReader(filename);

% Defined constants
HorCalibrate = 554.75/7;   %Horpixel/mm
VertCalibrate = 216.39/4;  %Vertpixel/mm
PI = 3.14159;              % Pi
vel = 3;                   % mm/s
h = 0.42;                  % mm
NozSpot = 188.96;          % distance in vertical pixels to nozzle
center, changes at any movement of camera
FrameJump = 3;             % Frequency of Frames uploaded
% Image information
FramesPerSecond = 29.97;    % Will be dependent on specific video
feed; change accordingly

load ReferenceSignal.mat         % Pulse type input

clear avi_info; clear Frame; clear mov; clear Ibw; clear Image_Gray;
clear Seg_Image; clear rod_width; clear TimeTemp; clear VCorrection;
clear Time;
```

```
% Image based variables
ROIDist = vel*FrameJump/FramesPerSecond;           % mmV/Frame
DoubleROIPix = ROIDist*VertCalibrate;       % Vpix/Frame double format
ROIPix = round(ROIDist*VertCalibrate);      % Vpix/Frame
Pixel2Time = DoubleROIPix*FramesPerSecond/FrameJump; % Vpix/s

% This section displays 9 sequential images from the video file for
% the user to select the image where movement starts

happy = 0;   hold = 0;      % Sometimes the region of interest is
 improperly
                            % selected or start frame auto-selected
                             wrong, happy
                            % and hold variables allows the code to
                            % be rerun with a different ROI and
                            start frame without
                            % interrupting the higher level code
                            % ILC_Implement_MI.
while (happy ~= 1)
    if (hold == 0)
        hold = 1;
        [Start_Frame] = Movement(temp_mov)
    else
        answer = 0; rod_width = 0;
        while (answer ~= 1)
            Start_Guess = input('Guess Moving Frame: ');
            figure (1)
            for i = 1:9         % Plot 9 images
                subplot(3,3,i)
                imshow(read(temp_mov,Start_Guess+i-1))
                xlabel (Start_Guess+i-1)
            end
            answer = input('Correct Range? (1 = Yes, 0 = No): ');
            if (answer == 1)
                Start_Frame = input('Starting Image :');
            end
            close(1)
        end
    end

    % Line Start
    Start_Line = round(2.2 * FramesPerSecond) + Start_Frame;

    % Line End
    End_Line = round(14.1 * FramesPerSecond) + Start_Frame;

    % reading one frame at a time and storing it in to array
    dummy = 0;
    for i=Start_Line:FrameJump:End_Line;    % Code only extracts data
     from every 3rd image
        dummy = dummy + 1;
%         mov=aviread(filename,i);
        Frame(:,:,:,dummy) = read(temp_mov,i);

    end

    clear xi; clear yi;

    if (Iteration == 1)
        % Select Region of interest
        figure(1)

        [BW,xi,yi] = roipoly(Frame(:,:,:,Check_Point));
```

```
        % User selects region
        min_x = int16(min(xi));
        % of interest around
        max_x = int16(max(xi));
        % nozzle tip
        min_y = int16(min(yi));
        % Use this one for time shift calc
    end

    close all

    clear Seg_Image; clear Image_Gray; clear Ibw;

    rod_width(1:2) = 0;   % Initializations
    Time(1:2) = [-1,(double(Start_Line) - Start_Frame)/FramesPerSecond
     - (-NozSpot+ROIPix-1+double(min_y))/double(Pixel2Time)-.001 +
       .3244/5];    % Makes rod_width at t0 to tstart = 0
    int = 0;

    % Final run through image for calculations
    for i = 1:round((End_Line - Start_Line)/FrameJump)

        Seg_Image(:,:,:,i) = Frame(min_y:min_y+ROIPix-1,min_x:max_x
        ,:,i);   % segment image to ROI
        Image_Gray(:,:,i) = .2989*Seg_Image(:,:,1,i)+.5870*Seg_Image
        (:,:,2,i)+.1140*Seg_Image(:,:,3,i); %convert to grayscale
        from standard values
%         level = graythresh(Image_Gray(:,:,i))-.05;
%Find threshold level
%         if (level < 0 )
% level between 0 and 1
%             level = 0;
%         end
        level = 0.30;
        % empiracally determined level
        Ibw(:,:,i) = im2bw(Image_Gray(:,:,i),level);
        %Convert to BW binary image

        Size = length(rod_width);
        % For array indexing

        for j = 1:ROIPix            % Scan through image
            int = int + 1;
            rod_width(Size+j) = (1/HorCalibrate)*sum(Ibw(ROIPix-j+
            1,:,i));    %Calculates rod width in mm by summing along
            rows

            R = rod_width(Size+j)/2;
            % Correction Factor based on thesis Section 5.2
            if (R <= 0.5 * h)
                VCorrection(Size+j) = (PI)* R^2;
                 % If diameter is less than fly height
            else
                theta = asin(.5*h/R);
                % If diameter is greater than fly height
                VCorrection(Size+j) = (2*theta*R^2+0.5*h^2*(1/tan
                (theta)));
            end

            % Calculation of time vector is tricky.  This equation
            % shifts time based on a equation that includes the
            % pixel2time ratio, the starting frame, and the time
```

```
            % shift brought on by the distance between the nozzle
            % tip and the ROI location
            Time(Size+j) = (int-1)*1/double(Pixel2Time) +
            (double(Start_Line) - Start_Frame)/FramesPerSecond -
            (-NozSpot+ROIPix-1+double(min_y))/double(Pixel2Time)
            + .3244/5; %Time Calc starting at 0, Added box location
             part, plus deceleration losses
        end
    end
    % the variable frequency VCorrection signal must be modified to
    % a 1kHz signal.  Here the interp1 function linearly interpolates
    % the variable frequency signal to provide evenly spaced data
    % points at 1kHz
    Time = sort(Time);
                % Prevents interp1 errors
    Vdot1kHz = interp1(Time,VCorrection,TimeRef,'linear');
               % mm^3/s at 1kHz: dim(TimeRef) < dim(Time)
    for n = 1:length(Vdot1kHz)
        if (isnan(Vdot1kHz(n)))
                % NaN's corrupt data.  Turn all Nan's to previous
                 values
            Vdot1kHz(n) = mean(Vdot1kHz(n-10:n-1));
        end
    end
    Error = Reference - Vdot1kHz(1:length(Reference));
              % Error signal.  mm^3/s

    %Plots
%     figure(1)
%     plot(Time,VCorrection,'o-', TimeRef, Vdot1kHz,'.-')
%     xlabel ('Time (s)')
%     ylabel ('Volumetric Flow Rate (mm^3/mm)')
%     xlim ([0 14])
    figure (2)
    plot(TimeRef, Reference,'k-', TimeRef, Vdot1kHz, 'b.-',TimeRef,
    Error,'r.-')
    xlabel ('Time (s)')
    ylabel ('Output (mm^3/mm)')
    legend ('Reference', 'Q', 'Error')
%     figure(3)
%     plot(Time, rod_width,'.-')
%     xlabel ('Time (s)')
%     ylabel ('Rod Width (mm)')
%     xlim ([0 14])
    happy = input('Happy with results? (0 = No, 1 = Yes): ');
     % Assures ROI and other things were properly chosen
end
```

Motion.m

```
% Optical Movement Detection
```

```
% Name: Seongsu Yun, Date: 07/30/09, Modified: 06/23/10 by
David Hoelzle
function [Start_Frame] = Movement(mov)

% video = 'It2.avi';
thresh = 25000000; % experimental threshold value
sum = 0; % set numerical summation of pictures
framenum = 11; % first frame number set
L1 = read(mov,framenum); % Load first frame
[m,n] = size(L1);

% Calculate movement of nozzle by use of correlation
while (sum < thresh)
    sum = 0; % reset sum
    framenum = framenum + 1; % for loading next frame
    L2 = read(mov,framenum); % Load new image
    for x1 = 1:n % experimental window size (x-axis)
        for y1=1:m % experimental window size (y-axis)
            sum = sum+(double(L1(y1,x1))-double(L2(y1,x1)))^2;
            % calculate error and summation
        end
    end
    L1 = L2; % Old image to New image (make program efficient)
    sum_(framenum - 10) = sum;
end

% Display result
framenum = framenum; % restore framenum value for print
Start_Frame = framenum;

plot(11:framenum, sum_, 'k-o', [11, framenum+10], [thresh, thresh],
'k--')
xlabel ('Frame Number')
ylabel ('Correlations')

% % Plot
% for i = 1:10
%     subplot(2,5,i)
%     image(frame2im(aviread(video,framenum+i-4)));
%     title(framenum+i-3)
% end
```

usplice.m

```
% Splice a u into the start, steady, and end sections
% syntax [ustart, usteady, uend, utail, Ucompile] = usplice(u,
startstart (s), startend (s), endstart (s), endend (s), Extruder
% Number);
function [ustart, usteady, uend, utail, UCompile] = usplice(u,
 startstart, startend, endstart, endend, ExtNum);

% load filtersig;
% u = filtfunc;

% ExtNum = 1;           % unitless

if (ExtNum < 5)           % Only 1 - 4 extruders allowed

%     startstart = 1.4;    % s
```

```
%     startend = 5;     % s
%     endstart = 5.001;       % s
%     endend = 14.2;        % s
    StartNom = 2.5;         % s
    EndNom = 11;           % s
    vel = 3;              % mm/s

    ustart = u(startstart*1000:startend*1000);
    usteady = mean(u(startend*1000:endstart*1000));
    uend = u(endstart*1000+1:endend*1000);
    utail = 0;

    UCompile(:,1:16) = zeros(100,16);
    UCompile(1,1+4*(ExtNum-1)) = (StartNom - startstart) * vel;
    % Start Lead
    UCompile(2,1+4*(ExtNum-1)) = length(ustart);
    % Start Length
    UCompile(3:length(ustart)+2,1+4*(ExtNum-1)) = ustart;
    % Start Trajectory
    UCompile(1,2+4*(ExtNum-1)) = usteady;
    % Steady State
    UCompile(1,3+4*(ExtNum-1)) = (EndNom - endstart) * vel;
    % End Lead
    UCompile(2,3+4*(ExtNum-1)) = length(uend);
    % End Length
    UCompile(3:length(uend)+2,3+4*(ExtNum-1)) = uend;
    % End Trajectory
    UCompile(1,4+4*(ExtNum-1)) = 0;
    % Dummy decel info
    UCompile(2,4+4*(ExtNum-1)) = 1;
    %
    UCompile(3:4,4+4*(ExtNum-1)) = usteady;

    %save ('UTraj.mat', 'UCompile')

    figure (300)
    plot(.001:.001:length(u)/1000,u,'k--',startstart:.001:
    startend, ustart,'g-',[startend+.001,endstart-.001],
    [usteady, usteady],'m-.',endstart+.001:.001:endend,uend,'r--')
    %axis([0 14 -30 30])
    xlabel ('Time (s)')
    ylabel ('Input (mm^3/mm)')
    legend ('Original','Start','Steady-State','Stop')

    % save uinfo.mat
else
    disp('Extruder Number must be (1-4)')
end
```

usplice_complete.m

```
% Splice a u into the start, steady, and end sections
% syntax [ustart, usteady, uend, utail, Ucompile] = usplice(u,
% startstart (s),
% startend (s), endstart (s), endend (s), Extruder Number);
function [ustart, usteady, uend, udecel, UCompile] = usplice_complete
```

```
(u, startstart, startend, endstart, endend, decelstart, decelend,
ExtNum);

% load filtersig;
% u = filtfunc;

% ExtNum = 1;           % unitless
ts = 0.001;

if (ExtNum < 5)           % Only 1 - 4 extruders allowed

%     startstart = 1.4;    % s
%     startend = 5;     % s
%     endstart = 5.001;       % s
%     endend = 14.2;        % s
    StartNom = 2.5;         % s
    EndNom = 11;           % s
    DecelNom = 7;           % s
    vel = 3;               % mm/s

    ustart = u(startstart*1000:startend*1000);
    usteady = mean(u([startend*1000:decelstart*1000, decelend*1000:
    endstart*1000]));
    uend = u(endstart*1000+1:endend*1000);
    udecel = u(decelstart*1000+1:decelend*1000);
    utail = 0;

    UCompile(:,1:16) = zeros(100,16);
    UCompile(1,1+4*(ExtNum-1)) = (StartNom - startstart) * vel;
    % Start Lead
    UCompile(2,1+4*(ExtNum-1)) = length(ustart);
    % Start Length
    UCompile(3:length(ustart)+2,1+4*(ExtNum-1)) = ustart;
    % Start Trajectory
    UCompile(1,2+4*(ExtNum-1)) = usteady;
    % Steady State
    UCompile(1,3+4*(ExtNum-1)) = (EndNom - endstart) * vel;
    % End Lead
    UCompile(2,3+4*(ExtNum-1)) = length(uend);
    % End Length
    UCompile(3:length(uend)+2,3+4*(ExtNum-1)) = uend;
    % End Trajectory
    UCompile(1,4+4*(ExtNum-1)) = (DecelNom - decelstart) * vel;
    % Decel Lead
    UCompile(2,4+4*(ExtNum-1)) = length(udecel);
    % Start Length                                      %
    UCompile(3:length(udecel)+2,4+4*(ExtNum-1)) = udecel;
    % Decel Trajectory

    %save ('UTraj.mat', 'UCompile')

    figure (300)
    plot((.001:.001:length(u)/1000)/ts,u,'k--',(startstart:.001:
    startend)/ts,ustart,'g-',[startend+.001,decelstart-.001]/ts,
    [usteady,usteady],'m-.',(endstart+.001:.001:endend)/ts,uend,
    'r--',(decelstart+.001:.001:decelend)/ts,udecel,'b--',
    [decelend+.001,endstart-.001]/ts,[usteady, usteady],'m-.',
    StartNom/ts, u(StartNom*1000), 'go', EndNom/ts, u(EndNom*1000),
    'ro', DecelNom/ts, u(DecelNom*1000), 'bo')
    %axis([0 14 -30 30])
```

```
    xlabel ('Time Index')
    ylabel ('Input (mm^3/mm)')
    legend ('Original','Start','Steady-State','Stop', 'Corner')
    % save uinfo.mat
else
    disp('Extruder Number must be (1-4)')
end
```

References

[1] J. Taboas, R. Maddox, P. Krebsbach, and S. Hollister, "Indiret solid free form fabrication of local and global porous, biomimetic and composite 3d polymer-ceramic scaffolds," *Biomaterials*, vol. 24, pp. 181–194, 2003.

[2] S. J. Kalita, S. Bose, H. L. Hosick, and Bandyopadhyay, "Development of controlled porosity polymer-ceramic composite scaffolds via fused deposition modeling," *Materials Science and Engineering: C*, vol. 23, no. 5, 2003.

[3] S. J. Hollister, "Porous scaffold design for tissue engineering," *Nature Materials*, vol. 4, pp. 518–590, 2005.

[4] A. Khoda, I. T. Ozbolat, and B. Koc, "Engineered tissue scaffolds with variational porous archictecture," *Journal of Biomechanical Engineering*, vol. 133, pp. 011 001–1, 2011.

[5] J. Stampfl, S. Baudis, C. Heller, R. Liska, A. Neumeister, R. Kling, A. Ostendorf, and M. Spitzbart, "Photopolymers with tunable mechanical properties processed by laser-based high-resolution stereolithography," *Journal of Micromechanics and Microengineering*, vol. 18, p. 125014, 2008.

[6] K. Cooper, *Rapid Prototyping Technology: Selection and Application*. Marcel Dekker, 2001.

[7] J. Cesarano III, J. G. Dellinger, M. P. Saavedra, D. D. Gill, R. D. Jamison, B. A. Grosser, J. M. Sinn-Hanlon, and M. S. Goldwasser, "Customization of load-bearing hydroxyapatite lattice scaffolds," *International Journal of Applied Ceramic Technology*, vol. 2, no. 3, pp. 212–220, 2005.

[8] S. K. Lan Levengood, S. J. Polak, M. B. Wheeler, A. J. Maki, S. G. Clark, R. D. Jamison, and A. J. Wagoner Johnson, "Multiscale osteointegration as a new paradigm for the design of calcium phosphate scaffolds for bone regeneration," *Biomaterials*, vol. 31, pp. 3552–3563, 2010.

[9] S. J. Polak, S. K. Lan Levengood, M. B. Wheeler, A. J. Maki, S. G. Clark, and A. J. Wagoner Johnson, "Analysis of the roles of microporosity and BMP-2 on multiple measures of bone regeneration and healing in calcium phosphate scaffolds," *Acta Biomaterialia*, vol. 7, no. 4, pp. 1760–1771, 2010.

[10] D. Bristow, M. Tharayil, and A. Alleyne, "A survey of Iterative Learning Control," *Control Systems Magazine*, vol. 26, no. 3, pp. 96–114, 2006.

[11] R. Horowitz, "Learning control of robot manipulators," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 115, pp. 402–411, 1993.

[12] M. Sekimoto, S. Kawamura, and T. Ishitsubo, "Basis-motion torque composition approach: Generation of motions with different velocity profiles among joints," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, October 18-22 2010, pp. 670–676.

[13] J. R. Woodard, A. J. Hilldore, S. K. Lan, C. Park, A. W. Morgan, J. A. C. Eurell, S. G. Clark, M. B. Wheeler, R. D. Jamison, and A. J. Wagoner Johnson, "The mechanical properties and osteoconductivity of hydroxyapatite bone scaffolds with multi-scale porosity," *Biomaterials*, vol. 28, no. 1, pp. 45–54, 2007.

[14] R. Chhabra and J. Richardson, *Non-Newtonian Flow and Applied Rheology: Engineering Applications*. Oxford, UK: Butterworth-Heinemann, 1999.

[15] I. Gibson, D. Rosen, and B. Stucker, *Additive Manufacturing Technologies*. Springer, 2010.

[16] J. E. Smay, J. Cesarano III, B. A. Tuttle, and J. A. Lewis, "Piezoelectric properties of 3-X periodic $Pb(Zr_xTi_{1-x})O_3$–polymer composites," *Journal of Applied Physics*, vol. 92, no. 10, pp. 6119–6127, 2002.

[17] Frost & Sullivan, "U.S. bone graft and bone graft substitutes markets, Tech. Rep. F875-01, 2007.

[18] S. J. Hollister, "Scaffold design and manufacturing: From concept to clinic," *Advanced Materials*, vol. 21, pp. 3330–3342, 2009.

[19] J. L. Drury and D. J. Mooney, "Hydrogels for tissue engineering: scaffold design variables and applications," *Biomaterials*, vol. 24, pp. 4337–4351, 2003.

[20] P. V. Giannoudis, H. Dinopoulos, and E. Tsiridis, "Bone substitutes: An update," *International Journal of the Care of the Injured*, vol. 36S, pp. S20–S27, 2005.

[21] M. M. Stevens, "Biomaterials for bone tissue engineering," *Materials Today*, vol. 11, no. 5, pp. 18–25, 2008.

[22] Centers for Medicare and Medicaid Services, Bureau of Economica Analysis and Bureau of Census, "National health expenditures, aggregate and per capita amounts, and share of gross domestic product," Tech. Rep., 2004.

[23] The Bone & Joint Decade 2000-2010, Tech. Rep., 2009.

[24] K. A. Hing, "Bioceramic bone graft substitutes: influence of porosity and chemistry," *Internation Journal of Applied Ceramic Technology*, vol. 2, no. 3, pp. 184–199, 2005.

[25] I. Drosse, E. Volkmer, R. Capanna, P. De Biase, W. Mutschler, and M. Schieker, "Tissue engineering for bone defect healing: An update on a multi-component approach," *International Journal of the Care of the Injured*, vol. 39S2, pp. S9–S20, 2008.

[26] M. K. Smith, M. C. Peters, T. P. Richardson, J. C. Garbern, and D. J. Mooney, "Locally enhanced angiogenesis promotes transplanted cell survival," *Tissue Engineering*, vol. 10, no. 1/2, pp. 63–74, 2004.

[27] K. A. Hing, B. Annaz, S. Saeed, P. A. Revell, and T. Buckland, "Microporosity enhances bioactivity of synthetic bone graft substitutes," *Journal of Materials Science: Materials in Medicine*, vol. 16, pp. 467–475, 2005.

[28] Y. H. An, S. K. Woolf, and R. J. Friedman, "Pre-clinical *in vivo* evaluation of orthopaedic bioabsorbable devices," *Biomaterials*, vol. 21, 2000.

[29] L. G. Griffith, "Emerging design principles in biomaterials and scaffolds for tissue engineering," *Annals of the New York Academcy of Sciences*, vol. 961, 2002.

[30] A. J. Wagoner Johnson and B. A. Herschler, "A review of the mechanical behavior of CaP and CaP/polymer composites for applications in bone replacement and repair," *Acta Biomaterialia*, vol. 7, pp. 16–30, 2011.

[31] S. Michna, W. Wu, and J. A. Lewis, "Concentrated hydroxyapatite inks for direct-write assembly of 3-D periodic scaffolds," *Biomaterials*, vol. 26, pp. 5632–5639, 2005.

[32] J. L. Simon, S. Michna, J. A. Lewis, E. D. Rekow, V. P. Thompson, J. E. Smay, A. Yampolsky, J. R. Parson, and J. L. Ricci, "*In vivo* bone response to 3D periodic hydroxyapatite scaffolds assembled by direct ink writing," *Journal of Biomedical Materials Research Part A*, vol. 83, no. 3, pp. 747–758, 2007.

[33] T.-M. Chu, J. Halloran, S. Hollister, and S. Feinberg, "Hydroxyapatite implants with designed internal architecture," *Journal of Materials Science: Materials in Medicine*, vol. 12, 2001.

[34] J. Franco, P. Hunger, M. Launey, A. Tomsia, and E. Saiz, "Direct write assembly of calcium phosphate scaffolds using a water-based hydrogel," *Acta Biomaterialia*, vol. 6, pp. 218–228, 2010.

[35] J. Cesarano III, R. Segalman, and P. Calvert, "Robocasting provides moldless fabrication from slurry deposition," *Ceramic Industry*, vol. 148, no. 4, pp. 94–102, 1998.

[36] D. J. Hoelzle, A. G. Alleyne, and A. J. Wagoner Johnson, "Iterative Learning Control for robotic deposition using machine vision," in *Proc. of the IEEE American Control Conference*, Seattle, WA, USA, June 11-13 2008, pp. 4541 – 4557.

[37] F. Padieu and R. Su, "An h_∞ approach to learning control systems," *International Journal of Adaptive Control and Signal Processing*, vol. 4, pp. 465–474, 1990.

[38] K. Barton and A. Alleyne, "A cross-coupled Iterative Learning Control design for precision motion control," *Control Systems Technology*, 2008.

[39] K. Barton, D. Bristow, and A. Alleyne, "A numerical method for determining monotonicity and convergence rate in Iterative Learning Control," *Iternational Journal of Control*, vol. 83, no. 2, pp. 219–226, 2010.

[40] D. Bristow and J. R. Singler, "Robust analysis of slow learning in Iterative Learning Control systems," in *Proc. of IEEE American Control Conference*, San Fransicso, CA, USA, June 29 - July 1 2011.

[41] W. Terkaj, T. Tolio, and A. Valente, *A Review of Manufacturing Flexibility*, ser. Design of Flexible Production Systems. Berlin, Germany: Springer-Verlag, 2009.

[42] W. Messner, R. Horowitz, W.-W. Kao, and M. Boals, "A new adaptive learning rule," *IEEE Transactions on Automatic Control*, vol. 36, no. 2, 1991.

[43] S. Kawamura and N. Sakagami, "Analysis of dynamics of underwater robot manipulators basing on Iterative Learning Control and time-scale transformation," in *IEEE International Conference on Robotics & Automation*, Washington, DC, USA, May 2002, pp. 1088–1094.

[44] M. Arif, T. Ishihara, and H. Inooka, "Iterative Learning Control using information database," *Journal of Intelligent and Robotic Systems*, vol. 25, pp. 27–41, 1999.

[45] M. Alsubaie, C. Freeman, Z. Cai, P. Lewin, and E. Rogers, "ILC initial input selection with experimental validation," in *Proceedings of the Symposium on Learning Control at IEEE CDC*, Shanghai, China, 2009.

[46] D. Gorinevsky, D. E. Torfs, and A. Goldenberg, "Learning approximation of feedfoward control dependence on the task parameters with application to direct-drive manipulator tracking," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 4, pp. 567–581, 1997.

[47] H. Pollack, *Computer Numerical Control*. Englewood Cliffs, NJ: Prentice-Hall, 1990.

[48] J. van de Wijdeven and O. Bosgra, "Hankel Iterative Learning Control for residual vibration suppression with MIMO flexible structure experiments," in *Proc. of IEEE American Control Conference*, 2007.

[49] J. E. Smay, J. Cesarano III, and J. A. Lewis, "Colloidal inks for directed assembly of 3-D periodic structures," *Langmuir*, vol. 18, no. 14, pp. 5429–5437, 2002.

[50] J. Russias, E. Saiz, S. Deville, K. Gryn, G. Liu, R. Nalla, and A. Tomsia, "Fabrication and *in vitro* characterization of three-dimensional organic/inorgranic scaffolds by robocasting," *Journal of Biomedical Materials Research Part A*, vol. 83, no. 2, pp. 434–445, 2007.

[51] J. E. Smay, J. Cesarano III, B. A. Tuttle, and J. A. Lewis, "Directed colloidal assembly of linear and annular lead zirconate titanate arrays," *Journal of the American Ceramic Society*, vol. 87, no. 2, pp. 293–295, 2004.

[52] J. E. Smay, S. S. Nadkarni, and J. Xu, "Direct writing of dielectric ceramics and base metal electrodes," *International Journal of Applied Ceramic Technology*, vol. 4, no. 1, pp. 47–52, 2007.

[53] D. Therriault, R. F. Shepherd, S. R. White, and J. A. Lewis, "Fugitive inks for direct-write assembly of three-dimensional microvascular networks," *Advanced Materials*, vol. 17, no. 4, pp. 395–399, 2005.

[54] G. M. Gratson, F. García-Santamaría, V. Lousse, M. Xu, S. Fan, J. A. Lewis, and P. V. Braun, "Direct-write assembly of three-dimensional photonic crystals: Conversion of polymer scaffolds to silicon hollow-woodpile structures," *Advanced Materials*, vol. 18, pp. 461–465, 2006.

[55] D. Bristow and A. Alleyne, "A high precision motion control system with application to microscale robotic deposition," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 6, pp. 1008–1020, 2006.

[56] Q. Li and J. A. Lewis, "Nanoparticle inks for directed assembly of three-dimensional periodic structures," *Advanced Materials*, vol. 15, no. 19, pp. 1639–1643, 2003.

[57] D. J. Hoelzle, "Reliability guidelines and flowrate modulation for a micro Robotic Deposition system," Ph.D. dissertation, Urbana, IL, 2007.

[58] J. M. Cordell, M. L. Vogl, and A. J. Wagoner Johnson, "The influence of micropore size on the mechanical properties of bulk hydroxyapatite and hydroxyapatite scaffolds," *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 2, pp. 560–570, 2009.

[59] J. A. Lewis, "Colloidal processing of ceramics," *Journal of the American Ceramic Society*, vol. 83, no. 10, pp. 2341–2359, 2000.

[60] Z. Cai and E.-W. Bai, "Is Wiener\Hammerstein system identification really nonlinear?" in *Proc. of IEEE Conference on Decision and Control*, Atlanta, GA, 2010, pp. 1171–1176.

[61] J. Comb, P. Leavitt, and E. Rapoport, "Velocity profiling in an extrusion apparatus," Patent US 6 054 077, April, 2000.

[62] S. Khalil, J. Nam, and W. Sun, "Multi-nozzle deposition for construction of 3D biopolymer tissue scaffolds," *Rapid Prototyping Journal*, vol. 11, no. 1, 2005.

[63] H. Elci, R. Longman, M. Phan, J.-N. Juang, and R. Ugoletti, "Discrete frequency based learning control for precision motion control," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, 1994, pp. 2767–2773.

[64] J. A. Lewis, "Direct writing of 3D functional materials," *Advanced Functional Materials*, vol. 16, pp. 2193–2204, 2006.

[65] T.-M. Chu, D. Orton, S. Hollister, S. Feinberg, and J. Halloran, "Mechanical and *in vivo* performance of hydroxyapatite implants with controlled architectures," *Biomaterials*, vol. 23, 2002.

[66] E. Ritman, "Micro-Computed Tomography - current status and developments," *Annual Review of Biomedical Engineerig*, vol. 6, pp. 185–208, 2004.

[67] J. Cesarano III, "A review of Robocasting technology," in *Materials Research Society Symposium Proceedings*, Boston, MA, 1999, pp. 133–139.

[68] T. R. Elvitigala, A. D. Polpitiya, W. Wang, Stöckel, A. Khandelwal, R. S. Quatrano, H. Pakrasi, and B. K. Ghosh, "High-throughput biological data analysis: A step towards understanding cellular regulation," *IEEE Control Systems Magazine*, pp. 81–100, December 2010.

[69] Q. Fu, E. Saiz, and A. P. Tomsia, "Direct ink writing of highly porous and strong glass scaffolds for load-bearing bone defects repair and regeneration," *Acta Biomaterialia*, vol. need this, 2011.

[70] J. A. Lewis, J. E. Smay, J. Stuecker, and J. Cesarano III, "Direct ink writing of three-dimensional ceramic structures," *Journal of the American Ceramics Society*, vol. 89, no. 12, pp. 3599–3609, 2006.

[71] L. Liu, Z. Xiong, Y. Yan, R. Zhang, X. Wang, and L. Jin, "Multinozzle low-temperature deposition system for construction of gradient tissue engineering scaffolds," *Journal of Biomedical Materials Research Part B: Applied Biomaterials*, vol. 88B, no. 1, pp. 254–263, 2008.

[72] M. H. Ross and W. Pawlina, *Histology: A Text and Atlas.* Lippincott Williams & Wilkins, 1995.

[73] H.-S. Ahn, Y. Chen, and K. L. Moore, "Iterative Learning Control: Brief survey and catagorization," *IEEE Transactions on Systems, Man, and Cyberkinetics - Part C: Applications and Reviews*, vol. 37, no. 6, pp. 1099–1121, 2007.

[74] S. Mishra, K. Barton, A. Alleyne, P. Ferreira, and J. Rogers, "High speed drop-on-demand printing with a pulsed electrohydrodynamic jet," *Journal of Micromechanics Microengineering*, vol. 20, no. 9, pp. 1–8, 2010.

[75] Y. Wu and Q. Zou, "An iterative based feedforward-feedback control approach to high-speed afm imaging," in *Proc. of the IEEE American Control Conference*, 2009, pp. 1658–1663.

[76] C. T. Freeman, Z. Cai, P. L. Lewin, and E. Rogers, "Objective-driven ILC for point-to-point movement tasks," in *Proc. of the IEEE American Control Conference*, St. Louis, MO, USA, June 2009, pp. 252–257.

[77] S. Graebe and A. L. Ahlén, "Dynamic transfer among alternative controllers and its relation to antiwindup controller design," *IEEE Transactions on Control Systems Technology*, vol. 4, no. 1, pp. 92–99, 1996.

[78] K. Barton, D. Hoelzle, A. Alleyne, and A. Wagoner Johnson, "Cross Coupled Iterative Learning Control of systems with dissimilar dynamics: Design and implementation," *International Journal of Control*, vol. 84, no. 7, pp. 1223–1233, 2011.

[79] Y. Koren, "Cross-coupled biaxial computer control for manufacturing systems," *ASME Journal of Dynamic Systems, Measurement, and Control*, 1980.

[80] G. Chiu and M. Tomizuka, "Contouring control of machine tool feed drive system: A task coordinate frame approach," *Transactions on Control Systems Technology*, 2001.

[81] K. Barton, J. van de Wijdeven, A. Alleyne, M. Steinbuch, and O. Bosgra, "Norm optimal cross-coupled Iterative Learning Control," in *Proc. of IEEE Conference on Decision and Control*, 2008.

[82] A. Poo, J. Bollinger, and G. Younkin, "Dynamic errors in type 1 contouring systems," *IEEE Transactions on Industry Applications*, vol. 1A-8, pp. 477–484, 1972.

[83] K. Barton, D. Hoelzle, and A. Alleyne, "Cross Coupled Iterative Learning Control of dissimilar systems," in *Proc. of ASME Dynamic Systems and Control Conference*, 2009.

[84] T. Sugar and V. Kumar, *Lecture Notes in Control and Information Sciences: Control and coordination of multiple mobile robots in manipulation and material handling tasks*. Springer Berlin/Heidelberg, 1999.

[85] E. Haseltine and J. Rawlings, "Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics," *Journal of Chemical Physics*, vol. 117, pp. 6959–6969, 2002.

[86] A. Julius and G. Pappas, "Probabilistic testing for stochastic hybrid systems," in *Proc. of 47th IEEE Conference on Decision and Control*, Cancun, Mexico, Dec 9-11 2008, pp. 4030–4035.

[87] R. Shat, B. Rasmussen, and A. Alleyne, "Application of a multivariable adaptive control strategy to automotive air conditioning systems," *International Journal of Adaptive Control and Signal Processing*, vol. 18, pp. 199–221, 2004.

[88] M. Phan and R. Longman, "A mathematical theory of learning control for linear discrete multivariable systems," in *Prod. of the AIAA/AAS Astrodynamics Specialist Conference*, 1988.

[89] J. van de Wijdeven and O. Bosgra, "Residual vibration suppression using hankel Iterative Learning Control," *Int. J. of Robust and Nonlinear Control*, 2008.

[90] D. Bristow and B. Hencey, "A q, l factorization of norm-optimal Iterative Learning Control," in *Proc. of IEEE Conference on Decision and Control*, 2008.

[91] K. Barton and A. Alleyne, "Norm optimal ILC with time-varying weighting matrices," in *Proc. of IEEE American Control Conference*, 2009, pp. 264–270.

[92] Y. Koren and C. Lo, "Variable-gain cross-coupling controller for contouring," *CIRP Annals*, 1991.

[93] K. Srinivasan and P. Kulkarni, "Cross-coupled control of biaxial feed drive servomechanisms," *ASME Journal of Dynamic Systems, Measurement, and Control*, 1990.

[94] N. Amann, D. Owens, and E. Rogers, "Iterative Learning Control for discrete-time systems with exponential rate of convergence," *IEE Proceedings: Control Theory and Applications*, 1996.

[95] M. Norrlöf and S. Gunnarsson, "On the design of ILC algorithms using optimization," *Automatica*, 2001.

[96] J. Lee, K. Lee, and W. Kim, "Model-based Iterative Learning Control with a quadratic criterion for time-varying linear systems," *Automatica*, 2000.

[97] D. Hoelzle, A. Alleyne, and A. Wagoner Johnson, "Iterative Learning Control using a basis signal library," in *Proc. of IEEE American Control Conference*, St. Louis, MO, USA, June 10-12 2009, pp. 925–930.

[98] D. J. Hoelzle, A. G. Alleyne, and A. J. Wagoner Johnson, "Basis task approach to Iterative Learning Control with applications to micro-Robotic Deposition," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 5, pp. 1138–1148, 2011.