Fall 2022

Digital Twins in Advanced Manufacturing to Enhance Manufacturing Efficiency

Max Kirkpatrick

DIGITAL TWINS IN ADVANCED MANUFACTURING TO ENHANCE MANUFACTURING EFFICIENCY

by

Max Kirkpatrick

Bachelor of Science in Engineering
University of South Carolina, 2018

Submitted in Partial Fulfillment of the Requirements

For the Degree of Master of Science in

Aerospace Engineering

College of Engineering and Computing

University of South Carolina

2022

Accepted by:

Ramy Harik, Director of Thesis

David Rocheleau, Reader

David Wilkins, Reader

Cheryl L. Addy, Interim Vice Provost and Dean of the Graduate School

ACKNOWLEDGEMENTS

I would like to thank Ramy Harik for the opportunities afforded to me and his advising over the last 7 years. I would also like to thank the entire neXt research team and the McNair team as a whole, without whom this research would not have been possible. Additionally, I would like to thank Dave Wilkins and Siemens Digital Industries Software for the opportunity to complete this research while working in industry for the last 4 years. I have learned so much, both in industry and in research, none of which would have been possible without Siemens' collaboration. I would like to deliver a final thank you to everyone else who has supported me over the course of this program; you are too many to name, but your contributions are known and appreciated.

ABSTRACT

Digital Twins of manufacturing systems are an emerging tool for improvement, optimization, and monitoring of physical manufacturing systems. The idea of representing physical systems with digital depictions has been prevalent since the early days of Computer Aided Design (CAD), but recently the concept of a Digital Twin has been expanded to encapsulate more than just a digital model of a design. Instead, the Digital Twin offers a unique, comprehensive, and real-time toolset for analysis and improvement of physical systems. These tools can be used to improve system efficiency, perform "what-if" analyses, virtually commission manufacturing equipment, analyze system performance in real time, among many other capabilities. To validate these capabilities and assess their efficacy in a manufacturing setting, the creation, use, and analysis of Digital Twin (DT) systems was recorded and assessed across two manufacturing use cases. These use cases were both physical systems where the DT models could be tested and validated against real manufacturing equipment performance. DT models were created for both the Future Factories robotics lab and the Automated Fiber Placement Lab at the University of South Carolina's McNair Aerospace Research Center. The accuracy and utility of each DT model was assessed through physical experiments in each lab space. Advanced technologies such as Edge Computing, visual motion capture, virtual commissioning, and offline programming were utilized in the context of the DT models to enhance the DT environment and improve system accuracy. The assessment of these various DT techniques in the two labs demonstrated the utility and accuracy of a DT of manufacturing across both use cases.

Significant improvements were observed across the manufacturing design lifecycle, including design, programming, commissioning, and production. Widespread use of DT systems in manufacturing can drastically improve manufacturing efficiency, safety, and time to production, and it is expected that these systems will see continued development and use in the future.

TABLE OF CONTENTS

Acknowledgements iii

Abstract iv

Table of Contents vi

List of Tables viii

List of Figures ix

List of Abbreviations xiii

Chapter 1: Introduction 1

Chapter 2: Digital Twins in Manufacturing: A Review 5

2.1: Introduction to Digital Twins 5

2.2: Definitions and Categorization 9

2.3: Discussion 25

2.4: Conclusions 27

Chapter 3: Digital Twin Modeling for Future Factories and Automated Fiber Placement 29

3.1: Introduction and Application Descriptions 29

3.2: Requisite Technologies and Techniques 32

3.3: Digital Twin Modeling for Future Factories 36

3.4: Digital Twin Modeling for Automated Fiber Placement 46

3.5: Conclusions on Digital Twin Modeling 51

Chapter 4: Implementation of Digital Twin Systems for Future Factories and Automated Fiber Placement 52

4.1: Digital Twin Implementation for Future Factories.....52

4.2: Digital Twin Implementation for Automated Fiber Placement.....65

Chapter 5: Experimentation, Results, and Conclusions on Use of Digital Twin Systems for Future Factories and Automated Fiber Placement.....76

5.1: Digital Twin Experiments and Results for Future Factories.....76

5.2: Digital Twin Experiments and Results for Automated Fiber Placement.....110

5.3: Conclusions on the Use of Digital Twin Systems for Manufacturing.....131

5.4: Situation of Research.....134

References.....135

Appendix A: AFP Motion Reconstruction Comparison Figures and Vector Fields.....141

LIST OF TABLES

Table 2.1 Digital Twin of Equipment Publications 10

Table 2.2 Digital Twin of Process Publications.....14

Table 2.3 Digital Twin of Product Publications 16

Table 5.1 DT and Hand Measurement of Dimensions from the FF Cell.....80

Table 5.2 DT and 3D Scan Measurements of Dimensions from the FF Cell 84

Table 5.3 Critical Dimensions and Deltas for both Hand Measurement and 3D Scan Measurement.....87

Table 5.4 Combined Motion Capture Data for R02 and R03 Circle Fitting.....91

Table 5.5 Coordinate Data of Resultant Locations Relative to Cell Origin 92

Table 5.6 A Comparison of Measurement Techniques for Dimensions Locating R02 and R03.....93

Table 5.7 Z-Deviation Data for All 6 Tested Robot Orientations 103

Table 5.8 Average Point Errors for Motion Reconstruction of All Plies 123

LIST OF FIGURES

Figure 2.1 An Example of a DT of Equipment Model 13

Figure 2.2 Digital Twin of Product for a Cowl Component 18

Figure 2.3 Distribution of Publications Among DT Categories 26

Figure 2.4 Capabilities of Published DT Systems 27

Figure 3.1 The Future Factories Lab at the University of South Carolina 30

Figure 3.2 The Automated Fiber Placement Lab at the McNair Center 31

Figure 3.3 Various Layout Alternative Considered for the Future Factories Lab 37

Figure 3.4 The Final Design of the Future Factories Layout 38

Figure 3.5 The Kinematic Tree of a Yaskawa HC10 Robot in Process Simulate 39

Figure 3.6 The Joint Definition for J1 of the Yaskawa HC10 Robot 40

Figure 3.7 TOOLFRAME Definition for the HC10 Robot Model 41

Figure 3.8 The System Architecture of the VC Model 43

Figure 3.9 The Smart Object Overview for a Robot in the FF Lab 44

Figure 3.10 The AFP Lab Modeled in Process Simulate 48

Figure 3.11 The Kinematic Model of the AFP Machine Inside of Process Simulate 49

Figure 3.12 The Finalized Kinematic Model for the AFP System and Mandrel 50

Figure 4.1 The Point Cloud Produced by the 3D Scan of the FF Lab 55

Figure 4.2 The PLC Programming Structure Used in the FF Lab 56

Figure 4.3 Pick and Place Operations for R01 in the FF DT Model 59

Figure 4.4 Location Creation Inside the FF DT Model 60

Figure 4.5 Different Robot Configurations Inside the FF DT Model.....61

Figure 4.6 Two Motion Capture Target Configuration for End Effectors in the FF Lab..63

Figure 4.7 The IoT Architecture Used in the FF Lab 65

Figure 4.8 The 3D Scan of the AFP Lab 67

Figure 4.9 Image of the Imported Course STP File for AFP Programming.....69

Figure 4.10 A Collision Analysis Performed Inside PS 70

Figure 4.11 The Dataflow Between Applications for AFP and ACSIS Programming 71

Figure 4.12 Programming of ACSIS in PS.....73

Figure 4.13 The Layup Process for the Curved Tool.....75

Figure 5.1. Fixture Locating Dimensions from the DT Model.....78

Figure 5.2 Conveyor Locating Dimensions from the DT Model.....78

Figure 5.3 Table Locating Dimensions from the DT Model 79

Figure 5.4 Robot Locating Dimensions from the DT Model.....79

Figure 5.5 Obstacle Locating Dimensions from the DT Model 80

Figure 5.6 The Resultant Trimmed 3D Scan of the FF Lab 82

Figure 5.7 False Points Resulting from Reflections in the Glass Windows of the Doors in the FF Lab.....83

Figure 5.8 R02 As Viewed in the FF Point Cloud Data from the Left and Right 84

Figure 5.9 The Semi-Circular Paths Used for Motion Capture Calibration of R02 and R03 89

Figure 5.10 R02 Circular Path Curve Fitting Results.....90

Figure 5.11 R03 Circular Path Curve Fitting Results.....91

Figure 5.12 One of the Linear Test Paths Used on R05 97

Figure 5.13 Raw Motion Capture Results from the First Linear Test on R05.....98

Figure 5.14. The X-Z Projection of R05's Robot Motion During the First Linear Test ...98

Figure 5.15 Discretized Motion Capture Results from the First Linear Test on R05.....99

Figure 5.16 X-Z Projection of the Discretized Results from the First Linear Test on R05 99

Figure 5.17 The 6 Different Orientations Tested in Linear Motion Capture Paths 100

Figure 5.18 Up Orientation Linear Test Discretized Results.....101

Figure 5.19 Right Orientation Linear Test Discretized Results.....101

Figure 5.20 Down Orientation Linear Test Discretized Results.....101

Figure 5.21 Left Orientation Linear Test Discretized Results.....102

Figure 5.22 Forward Orientation Linear Test Discretized Results.....102

Figure 5.23 Backward Orientation Linear Test Discretized Results 102

Figure 5.24 Uncorrected Linear Test Discretized Results.....104

Figure 5.25 Corrected Linear Test Discretized Results.....104

Figure 5.26 The Planar Test Path for R05 105

Figure 5.27 Results from the Planar Test in the Up Orientation 105

Figure 5.28 Uncorrected Planar Path Discretized Results.....107

Figure 5.29 Corrected Planar Path Discretized Results.....107

Figure 5.30 The Resultant Trimmed 3D Scan of the AFP Lab 111

Figure 5.31 "Phantom" Images of the Lynx Z Carriage in the 3D Scan 112

Figure 5.32 The 3D Scan Overlayed On Top of the DT Model of the AFP Lab 113

Figure 5.33 Discrepancies Between the DT and the 3D Scan for the Mandrel Location and AFP Head Location.....114

Figure 5.34 The Delta Observed Between the DT Model and the 3D Scan for the X Axis of the Lynx Machine.....115

Figure 5.35 The Reconstruction of Ply 0101 Using Forward Kinematics in the Uncorrected DT Model 117

Figure 5.36 The Reconstruction of Ply 0101 Using Forward Kinematics in the Corrected DT Model Based on 3D Scanning Results 118

Figure 5.37 An Image of the Lynx Machine Performing Layup on the Curved Tool Surface 121

Figure 5.38 Machine Motion Reconstruction for Ply 0105 122

Figure 5.39 Planned Machine Motion for Ply 0105 122

Figure 5.40 Error Vector Field for the Motion Reconstruction of Ply 0105 123

Figure 5.41 The Motion Capture Recording for Ply 0101 125

Figure 5.42 The Motion Capture Recording for Ply 0102 125

Figure 5.43 The Error Vector Field for the Motion Capture from Ply 0101 126

Figure 5.44 The Error Vector Field for the Motion Capture from Ply 0102 126

Figure A.1 Motion Reconstruction for Ply 0101 141

Figure A.2 Motion Reconstruction for Ply 0102 141

Figure A.3 Motion Reconstruction for Ply 0105 142

Figure A.4 Motion Reconstruction for Ply 0112 142

Figure A.5 Motion Reconstruction for Ply 0117 142

Figure A.6 Motion Comparison Error Vector Field for Ply 0101 143

Figure A.7 Motion Comparison Error Vector Field for Ply 0102 143

Figure A.8 Motion Comparison Error Vector Field for Ply 0105 144

Figure A.9 Motion Comparison Error Vector Field for Ply 0112 144

Figure A.10 Motion Comparison Error Vector Field for Ply 0117 145

LIST OF ABBREVIATIONS

ACSIS ..... Automated Composites Structure Inspection System

AFP ..... Automated Fiber Placement

API ..... Application Programming Interface

BOE ..... Bill of Equipment

BOM ..... Bill of Materials

BOP ..... Bill of Process

CAD ..... Computer Aided Design

CAM ..... Computer Aided Manufacturing

CAPP ..... Computer Aided Process Planning

DB ..... Data Block

DT ..... Digital Twin

FB ..... Function Block

FBD ..... Functional Block Diagram

FC ..... Function Call

FEA ..... Finite Element Analysis

FF ..... Future Factories

HiL ..... Hardware in the Loop

HMI ..... Human Machine Interface

IO ..... Input Output

IoT ..... Internet of Things

IIoT ..... Industrial Internet of Things

IMT ..... Ingersoll Machine Tools

KRL..... KUKA Robot Language

LAD .....Ladder (Programming Language)

ML..... Machine Learning

MQTT .....Message Queuing Telemetry Transport

OB..... Organizational Block

OLP..... Off-Line Programming

OPC.....Open Platform Communications

OPCUA..... Open Platform Communications Unified Architecture

PF .....Ply File

PLC ..... Programmable Logic Controller

PLM ..... Product Lifecycle Management

PPF..... Posted Ply File

PS .....Process Simulate

SCL .....Structured Control Language

SIL..... Safety Integrity Level

SiL.....Software in the Loop

STL .....Statement List (Programming Language)

TCP .....Tool Center Point

TCPF .....Tool Center Point Frame

TIA.....Totally Integrated Automation

VC.....Virtual Commissioning

VCP.....Vericut Composites Programming

CHAPTER 1

INTRODUCTION

Digital Twins of manufacturing systems are an emerging tool for improvement, optimization, and monitoring of physical manufacturing systems. The idea of representing physical systems with digital depictions has been prevalent since the early days of Computer Aided Design (CAD), but recently the concept of a Digital Twin has been expanded to encapsulate more than just a digital model of a design. Instead, the Digital Twin offers a unique, comprehensive, and real-time toolset for analysis and improvement of physical systems. These tools can be used to improve system efficiency, perform "what-if" analyses, virtually commission manufacturing equipment, analyze system performance in real time, among many other capabilities. This thesis studies the application of these Digital Twin models to the manufacturing space.

Two use cases were used to analyze the creation and use of Digital Twins of manufacturing. These use cases are both physical systems where the Digital Twin models could be tested and validated against real manufacturing equipment performance. The chapters of this document will first introduce the concept of a Digital Twin of a manufacturing system and describe the theory behind these systems. Then we will walk through the development of Digital Twin systems for two labs at the University of South Carolina McNair Aerospace Research Center, and finally discuss results from both Digital Twin systems. This thesis will also document the usefulness of these systems and describe various advantages to using DT models for manufacturing systems.

The creation of Digital Twin (DT) models for the two labs at McNair was performed using industry standard tools as the backbone for the DT. Additionally, industry standard automation equipment such as 6 axis industrial arms and PLCs were used to demonstrate immediate applicability to industry problems. Some additional customizations were performed to these systems, and in these cases the customizations were created in environments where they can be natively integrated into the industrial system. The use of industry ready tools, software, and hardware leads to immediate applicability of the results presented in this thesis to industry. Additionally, the use of these tools removes development effort from the research team, allowing us to focus solely on the unsolved problems with the DT.

The two labs selected for this research were the Future Factories lab and the Automated Fiber Placement lab in the McNair center at the University of South Carolina. The Future Factories (FF) lab was constructed as a practical demonstration of many new and emerging manufacturing and programming techniques. The FF lab researches various techniques such as machine learning and advanced robotics. The lab was designed using DT technologies, and as such provides a unique opportunity to test various DT techniques for both planning and production in a well-controlled environment.

The Automated Fiber Placement (AFP) lab was selected as a candidate for DT creation because it was an already functional industrial manufacturing system. This system provided an opportunity to deploy a DT model to a running industrial system and analyze its efficacy and utility. The goal in creating a DT model of the AFP lab was to improve the manufacturing process of the system from initial programming to actual production of a

composite part. The results of the creation and use of both DT systems are described in chapter 5 of this thesis.

These models of the FF and AFP labs provide various elements of a DT of manufacturing in one unified environment. The FF and AFP DT models provide a positional layout of elements in the manufacturing workspace, kinematic modeling of moving components, modeling of logic driven components, programming and virtual commissioning capabilities, and an environment for analysis of real results from the manufacturing system. This document will demonstrate the creation and use of these models for both systems across all the above-described areas. Additionally, this thesis will show that DTs of manufacturing are powerful systems that can improve the output, usability, safety, and commissioning time of manufacturing systems.

The development of the DT model of the FF lab, as described in chapter 4 of this thesis, and the testing of the DT model of the FF lab against the physical system, as described in chapter 5 of this thesis, will show the usefulness of the DT system, as well as some potential drawbacks and pitfalls to avoid. Chapter 5 will demonstrate positional accuracy of the DT model, accurate kinematic and logical models, and the use of virtual commissioning to improve the equipment commissioning process. Various technologies like 3D scanning and real-time motion capture techniques will also be explored, and their applicability and utility to the DT model will be analyzed.

The development of the DT model of the AFP lab, described in chapter 4 of this thesis, and the results from this lab, described in chapter 5 of this thesis, will demonstrate the utility of the DT model to an already existent industrial manufacturing system. This document will demonstrate that the DT model provided meaningful benefits during the

programming and execution stages of manufacturing in the AFP lab. This document will also show how using the DT to program equipment allowed for all machines in the lab to be programmed in one integrated environment, incorporating both the AFP itself and supporting systems. Additionally, we will show how the DT model of the AFP lab allows for analysis of results from the AFP experiments to be performed in one integrated environment, and for these results to be contextualized against the original planned machine paths. Various techniques and technologies such as motion capture and kinematic path reconstruction will allow researchers to visualize and contextualize "as manufactured" data against the original product design. This contextualization in the DT could eventually allow for better design for manufacturability in the AFP space.

The combination of various DT techniques across two labs will demonstrate the utility of a DT of manufacturing across two disparate use cases. This document will describe the creation and use of a DT across the two use cases, demonstrate how various DT techniques can be used to improve design, programming, commissioning, and system analysis in a manufacturing setting. The DT of manufacturing is a powerful emerging tool in the manufacturing space, and one that we expect will see increased utilization in the future.

CHAPTER 2

DIGITAL TWINS IN MANUFACTURING: A REVIEW

In the last 20 years, the term “Digital Twin” has become increasingly prevalent in both industry and academia. The concept of transforming physical systems into digital representations has been around since the early days of Computer Aided Design (CAD), but recently the concept of a digital twin has been expanded to encapsulate more than just a digital model of a design. The term “Digital Twin” has developed several different meanings over the years. As used in the literature, a digital twin may be anything from a simple CAD model of a machine to a sophisticated, real-time, predictive system with bi-directional communication that informs or even automatically implements changes to the physical system. This paper seeks to present and categorize the current state of the art for digital twin systems in manufacturing, as well as clarify the use of the term “Digital Twin” through addition of informative qualifiers. This review has found that much of the existing work towards digital twins has been limited in scope, as most research only implements a subset of the capabilities of a digital twin for a manufacturing system. Manufacturing systems are quite complex, and as such, the creation of a “true digital twin” that encompasses the equipment, product, and process components of such a system has yet to be accomplished.

2.1 INTRODUCTION TO DIGITAL TWINS

The concept of a Digital Twin (DT) of a system is not new [1] [2]. The use of digital tools to represent physical systems has existed since at least 1964, with the development

of Ivan Sutherland's *Sketchpad* software [3]. *Sketchpad* is commonly regarded as the first Computer Aided Design (CAD) system, and since its development, CAD systems have become ubiquitous in engineering design. These CAD systems created the first DT models of products, being a one-to-one digital representation of the product *as planned*. As technology developed and computer aided systems became more capable, the use of computer systems for simulation of product performance using technologies such as finite element analysis (FEA) was developed, and the beginnings of the modern DT began to take shape. While the concept of a "virtual, digital equivalent to a physical product" was introduced as early as 2003 at the University of Michigan [4], the term "Digital Twin" didn't appear until at least 2011 as conceived by Michael Grieves and John Vickers [4] [2]. NASA initially defined the term digital twin in 2012 as follows: "A Digital Twin is an integrated multi-physics, multiscale, probabilistic simulation of an as-built vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its corresponding flying twin" [5]. As introduced, the term primarily pertained to a digital twin of an aircraft, however, in the years that followed, DT systems were applied to various use cases, including DT systems of manufacturing, maintenance, process, and product use [6].

Over the last 15 years, digitization, and the creation of sophisticated DTs to describe the behavior of complex physical systems, has become a driving force in the manufacturing market [7]. Based on a review conducted by Kendrik et al., the expectation is that the market for digital twin technologies will continue to grow by up to $27.06 billion by 2025 [8] [2]. As such, many companies, including Siemens, PTC, Dassault Systems, among many others, have developed tools and solutions for enabling the creation and use of a DT

[1]. Adoption of DT systems in industry is expected to continue to increase for the foreseeable future, and will continue to improve productivity and product quality [7] [9].

Until this point, we have been discussing DTs as originally defined by Grieves and NASA. While this definition gives a useful overview DT technology, it does not specifically pertain to DTs of manufacturing systems. In fact, the original definition provided by NASA in 2012 specifically refers to flying vehicles and the simulation of the as built vehicle using the DT [5]. This definition is limiting when discussing manufacturing DT systems, as it restricts us to just a digital twin of the produced product. Additionally, this definition heavily pertains to physics-based simulation of the product, but as we will discuss, DT systems can pertain to far more than just simulation. In particular, some cyber-physical systems can dynamically inform and provide live feedback to their physical components [10].

As the NASA interpretation of the digital twin is somewhat limiting when considering manufacturing systems, the following definition provided by Negri et al. will serve as the basis for discussion for this review: "The DT consists of a virtual representation of a production system that is able to run on different simulation disciplines that is characterized by the synchronization between the virtual and real system, thanks to sensed data and connected smart devices, mathematical models and real time data elaboration" [11]. Negri's definition allows for the application of the DT to a production system, and implies the possibility for real time connection, data interpretation, and corrective action. In much of modern development in the DT space, these are key areas of research interest [7] [12]. In particular, the capability to take real time corrective action has been identified by multiple literature reviews as a critical component of the DT [13] [12] [10].

Because of the complexity of the topic, many previous reviews have provided definitions and classifications for DT systems. In 2018, Kritzinger et al. provided three main classifications for DT systems using their level of maturity as a metric. These classifications included "Digital Model," "Digital Shadow," and "Digital Twin," depending on the level of integration between the digital and physical object [6]. In a review of cyber physical systems for Industry 4.0, Lee et. al proposed a "5C" architecture for cyber physical systems, including the connection level, the data conversion level, the digital simulation level, the cognition and feedback level, and the automatic self-configuration level [14]. Phanden et al. proposed distinctions between DTs of product and process, while noting that in the aerospace industry DTs of product are far more common. Additionally, Phanden mentions the topic of virtual commissioning for robotics DTs, an especially important topic for DTs of manufacturing systems [15] [16].

This review seeks to build upon these previous works by proposing an additional categorical classification scheme for DTs in published research that can work in conjunction with the previously proposed schemes. This review draws inspiration from Kritzinger's and Lee's classification schemes to create a modified classification scheme for DT systems. Additionally, this classification scheme is based on industry applications of DT technology, across the product lifecycle [17] [7]. Through this classification, this paper seeks to understand where the existing work is the strongest, and where the most additional research effort is required to continue the development of the topic.

2.2 DEFINITIONS AND CATEGORIZATION

2.2.1 Three Types of Digital Twins

Building on the definitions introduced above, and with some basis from industry PLM tools, we present three high level classification for DT models. These three DT types are as follows: *Digital Twin of Equipment, Digital Twin of Process,* and *Digital Twin of Product*. A DT of equipment is a DT model of the tools and resources used to manufacture a product. A DT of process is a DT model of the production process used to manufacture the product, including the production plan, scheduling, etc. A DT of product is a DT in the traditional NASA defined sense; it is a DT model of the product itself and its use. These three classifications are based on key components of product lifecycle management, particularly in the manufacturing space. During manufacturing planning, three main structures are considered in most PLM tools [18]. These structures are the *Bill of Materials* (BOM), the *Bill of Equipment* (BOE), and the *Bill of Processes* (BOP) [18] [19] [20] [21], which correspond with the *DT of Product, DT of Equipment,* and *DT of Process*, respectively. These three structures are linked and managed using common PLM tools like Siemens' Teamcenter. As this is an already existing separation from industry, we have used them here as the basis for our classification.

2.2.1.1 Digital Twin of Equipment

The first of the three main types of DTs we will investigate is arguably the most important and prevalent in the manufacturing space: the *DT of Equipment*. A DT of equipment is any model that meets any of the criteria of a DT provided above and is primarily used to model manufacturing equipment used to produce a product. DTs of equipment can vary in specificity from incredibly detailed models and simulations of

individual actuators all the way up to high level simulations of entire manufacturing facilities. Some examples of DTs of equipment include kinematic simulations of robotic arms [22], models of complete manufacturing cells [23], complex event driven simulations of manufacturing facilities [24], and multi-physics simulations of individual components of any piece of equipment [25], among many other types of models.

Various examples of DTs of equipment exist and have been proposed in the literature. As mentioned above, this is one of the most common types of DT in the manufacturing space. This review identified 17 publications primarily focused on demonstrating or proposing various forms of a DT of equipment. These sources, along with a brief description, are listed in Table 2.1.

Table 2.1. Digital Twin of Equipment Publications

Authors	Title	Year	Description
Talkhestani et al.	Consistency check to synchronize the Digital Twin of manufacturing automation based on anchor points	2018	Presents a methodology for synchronization of a digital twin model with changes in the real system using a rules-based approach.
Zhang et al.	A Reconfigurable Modeling Approach for Digital Twin-based Manufacturing System	2019	Presents modeling approach for reconfigurable manufacturing systems. Allows use of digital twin to explore possible options and issue corrective actions.
Liu et al.	A Cyber-Physical Machine Tools Platform using OPC UA and MTConnect	2019	OPCUA based approach to integration of machine tools into a cyber physical data collection system
Guerrero et al.	Virtual Commissioning with Process Simulation (Tecnomatix)	2014	Documents a virtual commissioning project using Process Simulate and a S7-300 PLC
Samadikhoshkho et al.	Modeling and Control of Aerial Continuum Manipulation Systems: A Flying Continuum Robot Paradigm	2020	Drone based flying continuum robotic manipulator is proposed. Basic control architecture is created and tested in simulation.
Bayram & Bozma	Coalition Formation Games for Dynamic Multirobot Tasks	2015	Mathematical modeling for collaborative multirobot tasks.

			Dynamic assignment of robots to tasks based on needs of that task.
Tao et al.	IoT-Based Intelligent Perception and Access of Manufacturing Resource Toward Cloud Manufacturing	2014	Application of IoT devices in Cloud Manufacturing. Manufacturing as a service requires real time knowledge of states of manufacturing equipment. Presents layered architecture for organization of manufacturing and compute resources
Uhlemann et al.	The Digital Twin: Realizing the Cyber-Physical Production System for Industry 4.0	2017	Proposes guidelines for digital twin implementation in production systems for SMEs. Primarily about data collection from a physical system. Some focus on machine vision
Yildiz et al.	Virtual Factory: Digital Twin Based Integrated Factory Simulations	2020	Virtual factory approach for manufacturing systems to support design changes to product and production. Primarily about visualization and training. VR approaches for training and design changes.
Guo et al.	Modular based flexible digital twin for factory design	2018	Factory level digital twin approach. Factory model built and optimized using Plant Sim. Evaluation of multiple different designs using digital factory and down selection
Park et al.	Design and implementation of a digital twin application for a connected micro smart factory	2019	Creation of a digital twin of a “micro smart factory” based on a number of small “cells” such as 3D printers, robots, CNCs etc. Data collection from all devices mapped back to 3D models. Decision making based on current machine status as represented in digital twin.
Zuehlke	SmartFactory—Towards a factory-of-things	2010	Researchers developed a smart factory model and tested against a physical demonstration platform. Provides a framework for control from sensor /actuator level to MES and ERP level.
Kuts et al.	Exploiting Factory Telemetry to Support Virtual Reality Simulation in Robotics Cell	2017	VR based digital twin of manufacturing cell. Relay data from cell to VR environment (unity) to allow users to visualize cell in real time. DT used to evaluate alternatives, learn about cell, training, etc. or use in

			an active mode where the user can influence the cell.
Long et al.	Attitude data-based deep hybrid learning architecture for intelligent fault diagnosis of multi-joint industrial robots	2020	Robot health monitoring and fault detection based on attitude data from last joint in kinematic chain
Seo et al.	Drone-enabled bridge inspection methodology and application	2018	Demonstrated capability to use drones for visual inspection of bridge systems. Compared to traditional methods, yielded same or better results.
Rakha & Gorodetsky	Review of Unmanned Aerial System (UAS) applications in the built environment: Towards automated building inspection procedures using drones	2018	Use of drones for visual inspection of buildings, primarily for energy consumption audits. Analysis of thermal imaging to identify areas of energy loss.
Kusiak & Verma	Analyzing bearing faults in wind turbines: A data-mining approach	2012	Machine learning approach to fault prediction in wind turbines. Predicted over-temp faults up to 1.5 hr. before fault. Models based on real data from wind turbines, using input parameters such as voltage phase, current phase, torque, and temperatures.

One particularly good example of a DT of equipment is shown by Park et al. in *Design and implementation of a digital twin application for a connected micro smart factory*. The authors demonstrate the creation and use of a DT system for all of the manufacturing cells, and their connections, used in the production of a product [26]. They call this system a "micro smart factory," a," have a complete representation of this system in the digital world that they use for planning, monitoring, etc. In *Exploiting Factory Telemetry to Support Virtual Reality Simulation in Robotics Cell*, Kuts et al. present a DT of equipment model based on a virtual reality platform that monitors and reports the state of a robotic cell in real time [27]. This system allows for simulation of various manufacturing alternatives, as well as a real time connection to allow the VR user to

monitor and effect changes to the equipment in the real cell. DTs of equipment do not always have to be associated with manufacturing. One final example from Kusiak and Verma demonstrates a successful DT of equipment used to monitor and predict bearing faults in an operational wind turbine [28].

Figure 2.1. An Example of a DT of Equipment Model as Shown by Park et al. in *Design and Implementation of a Digital Twin Application for a Connected Micro Smart Factory* [26]

2.2.1.2 Digital Twin of Process

The second type of DT model for manufacturing we will introduce is the *Digital Twin of Process*. The DT of process focuses heavily on *how* a product is manufactured, noting individual steps or sequences of operations required to produce the product. A DT of process may have substantial overlap with a DT of equipment, as a process description often requires equipment on which to perform said process. A DT of process, however, is distinct in its focus on twinning the specific process used, and providing monitoring, feedback, etc. for that process. Many DTs of process pertain to evaluating algorithms for scheduling or path planning using a DT system. For example, Phillips and Likhachev present a model for dynamic path planning of Automated Guided Vehicle (AGV) motions

using a DT of process model [29]. More recently, Xia et al. present the use of a DT of process for training reinforcement learning algorithms used for dynamic scheduling and machine vision in a production environment [30]. This review has identified 13 publications focused on demonstrating or proposing DT of process systems. These papers are listed in Table 2.2.

Table 2.2. Digital Twin of Process Publications

Authors	Title	Year	Description
Leng et al.	Digital twin-driven manufacturing cyber-physical system for parallel controlling of smart workshop	2018	Presents a model for use of a digital twin of manufacturing for highly customizable products by allocating manufacturing resources dynamically to create personalized products
Xia et al.	A digital twin to train deep reinforcement learning agent for smart manufacturing plants: Environment, interfaces, and intelligence	2020	Initial results towards using deep reinforcement learning applied to manufacturing systems – dynamic scheduling, vision, etc.
Liu et al.	Intelligent scheduling of a feature-process-machine tool supernetwork based on digital twin workshop	2020	Digital twin-based scheduling for a machine shop. Demonstrated effectiveness of a machine learning based scheduling algorithm using digital twin workshop.
Oyekan et al.	Applying a 6 DoF Robotic Arm and Digital Twin to Automate Fan-Blade Reconditioning for Aerospace Maintenance, Repair, and Overhaul	2020	Closed loop robotic grinding using a vision-based approach. Utilized a DT model for testing before deployment on robotic system.
Chen et al.	Discrete event-driven model predictive control for real-time work-in-process optimization in serial production systems	2020	Real time event driven model for production system optimization. Applications include resource use optimization, timing, and scheduling.
Phillips & Likhavchev	SIPP: Safe interval path planning for dynamic environments	2011	Presents an algorithm for path planning of AVG motion in dynamic environments. Tested using DT system.
Yoshida et al.	Planning 3-D Collision-Free Dynamic Robotic Motion Through Iterative Reshaping	2008	2 stage (static then dynamic) motion planning algorithm based on collision detection. Tested against simulated data.
Lerman et al.	Analysis of Dynamic Task Allocation in Multi-Robot Systems	2006	Presents a mathematical model of multi-robot task allocation in a shared workspace for large numbers of robots.

			Simulated algorithm in virtual environment, test results match mathematical models.
Pellegrielli et al.	Motion planning and scheduling for human and industrial-robot collaboration	2017	Dynamic path planning for human robot interaction. Optimization based on cycle times and equipment availability.
Chettibi et al.	Minimum cost trajectory planning for industrial robots	2004	Cost minimizing motion planning algorithm for industrial robotic systems. Allows for collision avoidance, velocity, jerk, accel, torque considerations.
Hu et al.	Petri-net-based dynamic scheduling of flexible manufacturing system via deep reinforcement learning with graph convolutional network	2020	Deep reinforcement learning approach to manufacturing operation scheduling. Multiple learning network approaches used against digital model.
Zhang et al.	A data- and knowledge-driven framework for digital twin manufacturing Cell	2019	Provides a framework for a real time digital twin of a manufacturing cell, outlining types of data that can be exchanged for different manufacturing systems (i.e., robots). Provides a model for giving real-time feedback from the digital twin to robots.
Xu et al.	Digital twin-based industrial cloud robotics: Framework, control approach and implementation	2020	Provides a framework for a cloud based digital twin of industrial robot systems. Distributed control architecture where the programmer works remotely, assigning manufacturing goals. Paths are determined locally using simulation software.

For further illustration, a few use cases of the DT of process are described below. Leng et al. present the concept of using a DT of process to plan and execute production of highly customizable products using a dynamic manufacturing planning technique in *Digital twin-driven manufacturing cyber-physical system for parallel controlling of smart workshop* [21]. In *Petri-net-based dynamic scheduling of flexible manufacturing system via deep reinforcement learning with graph convolutional network*, Hu et al. present a deep reinforcement learning approach to manufacturing operation scheduling that uses a DT model as a testing system [31]. As a final example, Oyekan et al. demonstrate a DT of

process that models a grinding operation for turbine blade repair. This physics-based model predicts material removal for given process parameters and is used in conjunction with robot simulation software to plan robotic grinding operations [32].

2.2.1.3 Digital Twin of Product

The third and final type of DT we present in this review is the *Digital Twin of Product*. A DT of product is far more closely related to the original NASA definition of the DT provided above. A DT of product is a DT model of a physical part or product that is the result of some manufacturing process. The DT of product can encompass details from CAD modeling, design simulations, through to product use and maintenance, as well as any other stage of the product lifecycle. Historical examples of DTs of product have included many aircraft and aircraft components, as the aerospace industry has been one of the most focused on the DT of product [4] [5]. DT of product models often rely on sophisticated physics-based modeling of products, using techniques like FEM. This review has identified 8 examples of publications pertaining to DTs of product in the manufacturing space. Far more examples of DTs of product exist outside the sphere of manufacturing. These publications are listed in Table 2.3, and a few examples of DTs of product are described below.

Table 2.3. Digital Twin of Product Publications

Author	Title	Year	Description
Seon et al.	Towards a digital twin for mitigating void formation during debulking of autoclave composite parts	2020	Introduces using a DT for mitigating void formation during an autoclave curing process. Details the creation of a finite element DT based on in situ XRAY scanning of cured parts.
Sacco et al.	Machine Learning Based AFP Inspection: A Tool for Characterization and Integration	2019	Presents methodology for automatic recognition of AFP defects using ML techniques.
Knapp et al.	Building blocks for a digital twin of additive manufacturing	2017	Presents thermal and physics-based modeling for additive manufacturing processes.

			Provides comparison of physics-based models to real behavior from additively manufactured parts.
Hurkamp et al.	Combining Simulation and Machine Learning as Digital Twin for the Manufacturing of Overmolded Thermoplastic Composites	2020	Finite element based DT for composites overmolding processes. Real machine data is input to finite element model to predict bond strength.
Zambal et al.	A digital twin for composite parts manufacturing	2018	Profilometry based defect recognition on dry fiber AFP parts. Assignment and classification of defects – analysis of impact of defects on part mechanical properties. Predictive structural analysis to determine if rework is required.
Stieber et al.	Towards Real-time Process Monitoring and Machine Learning for Manufacturing Composite Structures	2020	Process modeling for resin transfer in thermoplastic composites. Sensor equipped tooling monitors transfer of matrix to fiber. A machine learning model is trained based on sensor data to predict transfer effectiveness.
Hughes et al.	DIGITAL TWIN METHODOLOGY FOR COMPRESSION MOULDED THERMOPLASTIC COMPOSITE OPTIMISATION	2018	Digital twin of thermoplastic composite product. Fibersim used to create multiphysics model of layup to optimize design.
Haag & Anderl	Digital twin – Proof of concept	2018	Demonstrated a DT proof of concept model for a beam bending system. Finite element analysis used to predict how beam will behave, compared with physical system. Monitoring of physical system linked back to DT.

In *Digital Twin Methodology for Compression Moulded Thermoplastic Composite Optimization*, Hughes et al. present a DT of product model for determining an optimal composite layup strategy. They employ the use of physics-based modeling techniques to optimize the product design before manufacturing [33]. Knapp et al. present a DT of product model for laser based additive manufacturing, considering thermal and physics effects in *Building blocks for a digital twin of additive manufacturing* [34]. In *Machine Learning Based AFP Inspection: A Tool for Characterization and Integration*, Sacco et al. demonstrate a DT of product system for composite part inspection, allowing for the detection of defects on a completed part and mapping those defects back to the original

part model [35]. These are only a few of many examples of DTs of product available in the literature.

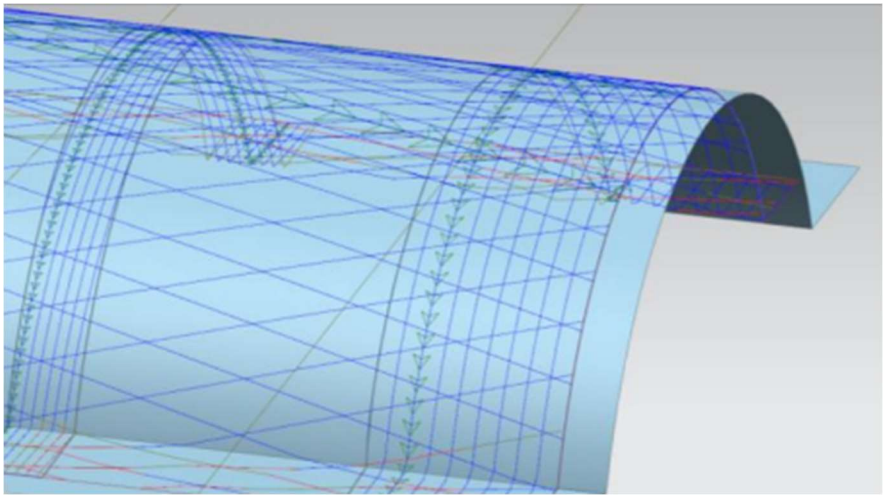

Figure 2.2. Digital Twin of Product for a Cowl Component as Shown by Hugues et al. in *Digital Twin Methodology for Compression Moulded Thermoplastic Composite Optimization* [33]

2.2.2 Classification of Capabilities of Digital Twins

In addition to the existing three categories of DTs provided above, we also present sub-classifications of DTs to better differentiate their capabilities in addition to their application (equipment, process, or product). These additional classifications provide a means to assess the technical readiness of a DT system, as well as allow a view of what capabilities of DTs are commonplace, and what capabilities require more development. When assessing a particular publication, the primary type of the DT (equipment, process, or product) was first assessed, and then the additional capabilities of the DT system were identified. The capabilities we identified for this assessment are as follows: *Simulation, Real-Time Communication, Data Flow Direction, Data Display and Feedback, Predictive Models, Adaptive / Learning Systems,* and *Independent Corrective Actions*. These classifiers are described in more detail in the sections below.

2.2.2.1 Simulation

The first, and most common, qualifier we assess here is *Simulation*. A DT that utilizes a detailed simulation model (at the level of detail required for the particular use case) to provide additional information about system behavior to an operator or another part of the DT is described as a DT with simulation capabilities. The simulation model used may be probabilistic, kinematic, physics based, or machine learning based, among many other simulation technologies. While simulation is not strictly required for a system to be regarded as a DT, nearly all DT systems have some simulation component, and simulation is somewhat of a hallmark for DT systems.

One interesting type of simulation DT with relevance to manufacturing is DTs that utilize *Virtual Commissioning*. As presented by Guerrero et al., Virtual Commissioning is a simulation of a manufacturing process that includes both the physical and logical (control) components of that process [23] [36]. This means simulating both the motion of a cell, as well as the control logic driving that cell. In practice, this results in a simulation encompassing all the physical components of the cell, as well as the PLC (Programmable Logic Controllers) controlling that cell. The PLC is included in the simulation either as a software component, or by connecting a physical PLC to the simulation over a protocol such as OPCUA.

Of the papers analyzed for this review, nearly all included some simulation component across all three main types of DT. Some notable examples are Yildiz et al.'s depiction of a virtual reality based model for virtual factories [37], Zuehlke's "SmartFactory" which created a factory level simulation of a physical system to plan and test order execution systems [38], Lerman et al.'s dynamic task allocation for robotic

systems modeling [39], and Steiber et al's process modeling for resin transfer in thermoplastic composites, which provides a more traditional DT of product with a robust simulation component [40]. These models, and many more, utilize simulation as a key component of the DT, and rely on the information from these simulation models to inform the behavior of the DT.

2.2.2.2 Real-Time Communication

Many DT models utilize communication between the virtual system and the physical system that they represent. Such systems may have real-time, or near real-time communication between these two elements. Real-time communication is a key differentiator, as real-time systems are often able to inform process decisions "in the moment" or update the DT immediately upon a change. Real-time data may be used for any number of different tasks in the DT or the physical system, but the capability to transmit and receive real time data in a DT model is a key capability. We provide the following definition for a real-time DT. A "real-time" DT is a DT that utilizes real-time (or near real-time) communication between the virtual and physical systems (either one way or two way, in either direction). Such a system may allow an operator to see and inform decisions based on real-time status information of a manufacturing system. This real time data is then used in another part of the DT.

DT models that include a real-time communication component are less common than other types of DT. Real-time communication is not required for a DT to perform many functions, and as such, we see far fewer examples of real-time DTs in the literature. Real-time DTs tend to be DTs of equipment or process; real-time DTs of product were not observed in this review. Some notable examples of real-time DTs from our review are as

follows: Liu et al.'s DT of equipment presentation of an OPCUA based approach for integration of machine tools into a cyber physical data collection system [41], Tao et al.'s IoT based approach for manufacturing resource monitoring in cloud based manufacturing [42], and Zhang et al.'s data driven approach to at DT of manufacturing, allowing data from the physical system to better inform simulation models [43].

2.2.2.3 Data Flow – Virtual to Physical and Physical to Virtual

Data flow, and the direction of data flow, is a key component of a DT model. As mentioned above, many DT systems use real-time communication to send data between the physical and virtual models, in either direction. Data flow in DT systems is important, as DT models need to be updated to reflect the physical system, and physical systems need to be informed using data from the DT model. We use the presence of this data flow, and its direction, to better classify DT systems.

Data can flow from the virtual model to the physical system, or from the physical system to the virtual model, or both. A DT that can push data from a virtual environment to the physical one, either independently or with operator intervention, is referred to as having virtual to physical communication. This communication does not have to be real-time. Examples of this data may be manufacturing strategies, robot trajectories, scheduling approaches, etc. A DT that can pull data from the physical environment to the virtual one, either independently or with operator intervention, is referred to as having physical to virtual communication. Likewise, this communication does not have to be real-time. Examples of this data may be machine coordinates, operational variables (such as temperatures, pressures, etc.), equipment availability, cycle times, etc. Some systems can

perform both types of communication. Such systems are said to have bi-directional communication.

Models that use virtual to physical communication are often used for process planning and execution. Some examples are Bayram and Bozma's mathematical modeling approach for planning multi-robot tasks [44], and Pellegrinelli et al.'s DT of equipment for motion planning and scheduling of robot operations in a human-machine collaborative environment [45]. Physical to virtual communication is often used to inform and update DT models to better match behavior of physical systems. Some examples are Liu et al.'s modeling of a DT workshop that utilized real data from the physical world to better inform their scheduling algorithms [46], and Sacco et al.'s machine learning based AFP inspection system where data from the inspection system was used to better inform the DT of product [35].

2.2.2.4 Data Display and Operator Feedback

The use of the data collected from a DT system, either from the virtual model or from the physical system, is critical to the operation of the DT. One simple application of this collected data is to display the data for an operator to better inform their decisions and provide feedback to the system. We say that such a DT system has *Data Display and Operator Feedback*. Such a DT provides easy display of information from a manufacturing system to an operator, often in the form of an interactable dashboard. This tool can be cloud-based but does not have to be. Such systems are slowly becoming more common place in industry as well, with the onset of cloud-based manufacturing monitoring tools.

Data display systems are relatively common for DT models. Adding the capability to view data from the cell and make informed decisions based on this data is a relatively

easy endeavor and can be added to most DT systems as an afterthought. With that said, we have identified a few examples of DT systems that execute data display well. Xu et al. present a DT model for industrial cloud robotics that incorporates data display to an operator through a cloud-based portal [47], while Kuts et al. use a virtual reality system to display data back to an operator in an immersive environment [27]. Another useful example is Zambal et al.'s use of a DT for composites part manufacturing which included data from inspection scans displayed in a useful rendering of defect locations on composites parts [48].

2.2.2.5 Predictive Models

One of the hallmark benefits of a DT model is the ability to create meaningful predictions about future behavior of a physical system based on the DT model. These predictions can range from simple planning of behavior using a simulation model to far more sophisticated predictions based on historical data and learning systems. We define a predictive model DT as a DT that provides useful predictions about the physical system's behavior utilizing either historical data or simulation models. These predictions can be used to automatically issue corrective actions, refine simulations, or inform operators, but do not have to be used in any of these manners.

Some example of predictive model DTs are Haag and Anderl's DT proof of concept showing a predictive FEA based model used to analyze beam bending in a DT system [49], Chettibi et al.'s approach for trajectory planning for industrial robots using a predictive model to predict and plan robot motion [50], and Zhang et al.'s modeling approach for DT based manufacturing systems which used DT models to predict behavior

of manufacturing systems and explore possible options and implementations before choosing a course of action [51].

2.2.2.6 Adaptive / Learning Systems

The ability of a DT to inform and update its virtual component to better match the physical system is very useful capability. A DT that utilizes an adaptive or learning system to update simulations and predictions based on incoming data from the real system is referred to as adaptive or learning. Such a system is preferred to be automatic but does not have to be. Many of these "learning" systems are based on machine learning or heuristic approaches. Adaptive or learning systems offer a key capability to DT systems, increasing the utility of the system and allowing a given DT to apply to multiple different environments by increasing adaptability. Learning system are also far less common than other types of DTs, as creating a useful adaptive and learning system is far more complex than a simple static model.

A few examples of adaptive or learning DT systems are Kusiak and Verma's data mining approach to wind turbine health monitoring, which used data collected from the field to inform and change models to better predict health degradation of wind turbines [28], Long et al.'s deep learning architecture for intelligent fault monitoring of industrial robots, which used attitude data from sensors mounted on a robot arm to inform a machine learning model that was trained to detect faults in the robot's motion [22], and Liu et al.'s DT approach to intelligent scheduling of manufacturing operations in a workshop, as previously discussed [46]. These systems represent serious advancements in DT systems that are capable of learning and adapting to the data from their physical component.

2.2.2.7 Independent Corrective Actions

The final capability of DT systems that we will discuss here is the ability for the system to issue independent corrective actions to the physical system. Such DTs are capable of independently effecting change to the physical system it is mirroring, without outside human or other intervention. These changes are used to avoid predicted faults or to better optimize machine operation based on simulation results. This more integrated capability for DT systems is often an end goal for DT development and displays a key ability to apply the knowledge gained from the DT system to the physical system. Such a capability requires a large amount of trust in the accuracy of the DT model, as the corrective actions taken are taken completely independently based on the data from the DT. As such, this is the least common capability of DT systems analyzed.

Examples of this capability were shown by Talkhestani et al.'s methodology for synchronization of a DT model against a physical model with induced changes [52], and Phillips and Likhachev's safe interval path planning for AGV motion in dynamic environments [29].

2.3 DISCUSSION

Many models for DTs in manufacturing systems have been proposed. These DTs range from more traditional DTs of Product to DTs of Equipment and Process, as described in section two of this review. Among the publications analyzed, DTs of Equipment were the most prominent systems for DTs related to manufacturing. The least common type of DT was DTs of Product. Figure 2.3 displays the distribution of the analyzed DT systems among the three main categories of DTs. DTs of Product are likely less common in the analyzed literature due to the selected search terms and the focus on manufacturing.

Creating DTs of Product is less common in the manufacturing space because of the focus on the process and equipment required to produce the product. DTs of Product are however more common among more traditional DT papers. The exact distribution of papers across the three categories is heavily impacted by the search terms used and active research areas, and as such is not necessarily representative of the research community as a whole, but rather a view into DT of manufacturing specifically. That said, integration of more DT of Product systems into manufacturing DT systems is likely an area that could use more investigation.

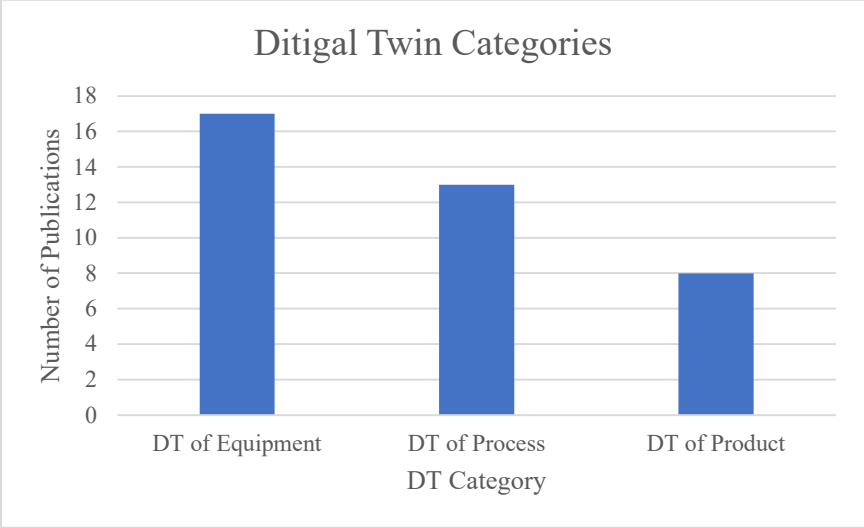

Figure 2.3. Distribution of Publications Among DT Categories

The distribution of capabilities of DT of manufacturing systems is also quite interesting. Across the seven capabilities assessed, DTs that included simulation and predictive capabilities were the most common. These capabilities are central pillars to DT systems, and as such their prevalence makes sense. The lease common types of DT systems were systems capable of issuing independent corrective action, learning systems, and systems capable of real time data processing. Independent and learning systems introduce substantial complexity to the implementation of a DT, and as such their lower prevalence

in published research is expected. Additionally, machine learning applications in manufacturing is still a relatively new research field and will likely grow in the future. The full distribution of DT capabilities is shown in Figure 2.4.

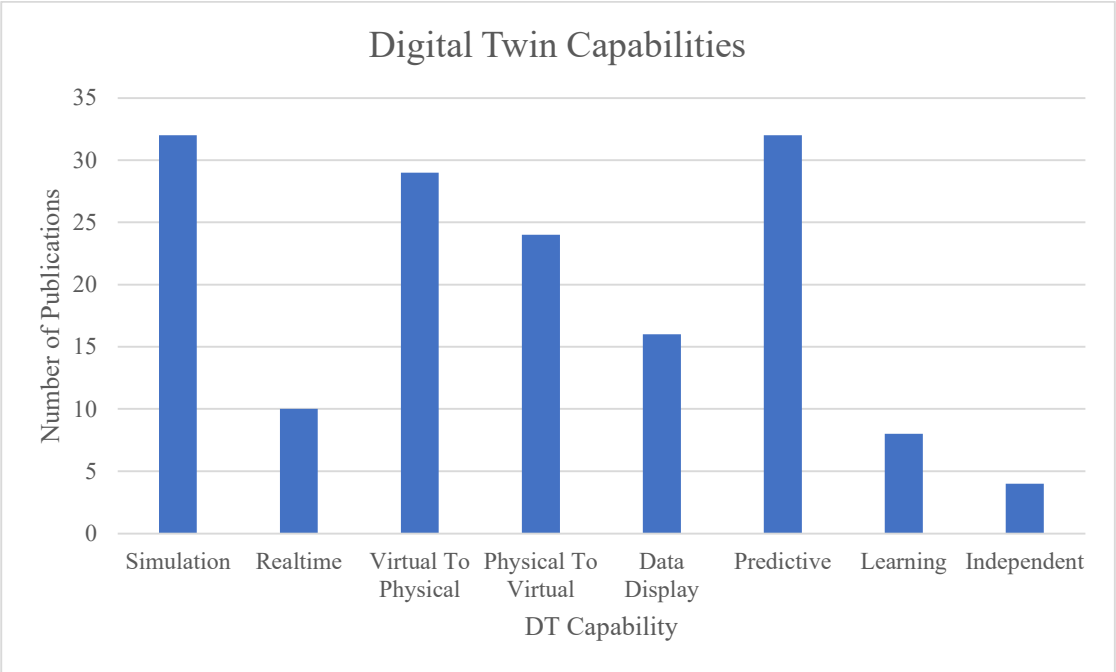

Figure 2.4. Capabilities of Published DT Systems

2.4 CONCLUSIONS

This review analyzed 38 recent publications pertaining to DTs of manufacturing. The DT systems from these papers were categorized into three main categories based on the portion of the PLM they addressed. The categories selected were DTs of Equipment, DTs of Process, and DTs of Product. DTs of Equipment were the most common type of DT system. Additionally, these publications were reviewed for capabilities of the DT systems. The most common capabilities were Simulation and Prediction, and the least common capabilities were Independence, Learning systems, and Real Time feedback. From this analysis, we argue that future development in DT systems incorporating machine learning models and DT systems capable of independently issuing corrective action to a

manufacturing system is required. This development would increase the capabilities of DT systems in manufacturing to advance our understanding of manufacturing processes, improve manufacturing performance, and unlock new capabilities for applications of DT systems in manufacturing.

CHAPTER 3

DIGITAL TWIN MODELING FOR FUTURE FACTORIES AND AUTOMATED FIBER PLACEMENT

3.1 INTRODUCTION AND APPLICATION DESCRIPTIONS

As described in chapter two of this thesis, the creation of a DT of manufacturing offers many benefits to the efficiency and knowledge of a production system. To test these DT systems and attempt to realize many of the benefits of these DT systems, we have constructed DT models for two manufacturing systems. The two demonstration platforms for DTs of manufacturing presented in this thesis are the University of South Carolina's Future Factories Lab and Automated Fiber Placement system. These two environments provide numerous opportunities to demonstrate applications of DTs of manufacturing across varied processes and manufacturing techniques. For the purposes of this research, the Future Factories lab often serves as a technology testbed for various DT techniques before their deployment to the AFP system. The Future Factories and Automated Fiber Placement test environments are described below.

3.1.1 Future Factories

The Future Factories (FF) lab is a test environment for various advanced manufacturing technologies constructed at the University of South Carolina's McNair Aerospace Research Center. The FF lab was built with the explicit purpose of being an open platform for testing advanced manufacturing techniques such as robotics, virtual commissioning, computer vision, machine learning, Industrial Internet of Things (IIoT),

data analytics, motion capture, and 3D scanning. The platform consists of five industrial robot arms and four conveyor systems, arranged in such a manner that the robots all have a high degree of flexibility in their assigned tasks, and can work cooperatively with other components of the cell. An image of the FF lab is shown below in Figure 3.1.

Figure 3.1. The Future Factories Lab at the University of South Carolina McNair Center

The automated process currently being demonstrated in the FF lab is a simple process assembling a 3D printed model rocket. The exact process being performed is of little consequence to this research, but the manner in which the rocket is assembled, and the technologies used to develop this process are relevant to this thesis. The rocket is assembled using a collaborative operation between multiple components of the cell, including three robotic arms, their respective grippers, and a conveyor system. The modeling of this process is described in more detail in section 3.3 of this chapter.

3.1.2 Automated Fiber Placement

The Automated Fiber Placement (AFP) lab at the University of South Carolina's McNair Aerospace Research Center is a lab housing a fully capable Lynx automated fiber placement machine developed by Ingersoll Machine Tools. This Lynx AFP machine is a 7-axis gantry style machine with three linear axes, three rotational wrist axes, and one external positioning axis. The Lynx system utilizes this combination of linear and rotational axes to create compound kinematic motion to accurately place carbon fiber tows onto the surface of a tool mounted on the external position axis. The AFP machine at McNair can produce full scale composite parts and offers the university a unique opportunity to perform experiments on production hardware. An image of the AFP lab is shown below in Figure 3.2.

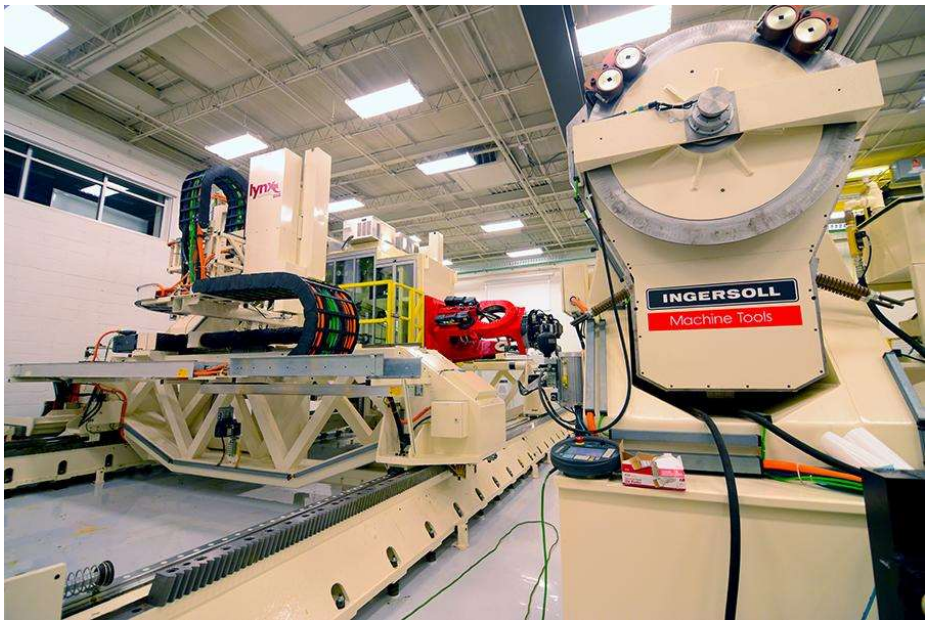

Figure 3.2. The Automated Fiber Placement Lab at the McNair Center

Additionally, the AFP lab contains an Automated Composite Structure Inspection System (ACSIS) for composite parts after their layup by the AFP machine. ACSIS consists of a rail mounted industrial robot arm and four profilometry scanners mounted on the end

of this arm. The combined motion of the robot and the rail allows the scanners to the entire surface of the composite part and precisely image the surface of the part to identify defects produced during the layup. The modeling and programming of ACSIS and the AFP system is discussed in more detail in section 3.4 of this chapter.

3.2 REQUISITE TECHNOLOGIES AND TECHNIQUES

To achieve the desired capabilities and level of accuracy of the DT systems for both FF and AFP, certain technologies and techniques were utilized. A relevant selection of these technologies and modeling techniques are described below.

3.2.1 Kinematic Modeling

Physical modeling of the manufacturing system in question is often the first step in the creation of a DT of manufacturing. After CAD models are generated for the DT, some physics-based modeling approach is often taken to describe the relationships between these CAD components and how the various pieces of the physical system interact. Many DT systems use a multi-physics or dynamics-based approach to modeling the physical interactions between components in the cell, however in this case, a kinematic modeling approach was taken. For this research, the simulation tool Process Simulate developed by Siemens Digital Industries Software was used to define the kinematic model for both FF and AFP. Kinematic modeling explicitly defines the movable components of the system and the relationship between these components by defining links and joints. Links define the movable rigid bodies of a component, and joints define the allowed relative motion between adjacent links in a kinematic tree. This explicitly constrained motion modeling technique allows for very fast calculation of motion in systems that may contain many moving components. Kinematic modeling was selected for this research due to the

simulation performance benefits over dynamics-based approaches, without losing much simulation accuracy.

3.2.2 Off-Line Programming

Off-Line Programming is an equally ubiquitous technology in the use of many DT of manufacturing systems. The DT is often used to plan and evaluate robotic motion paths in the DT before these paths are deployed to a real manufacturing system. In many cases the robot paths, or motion paths of any other kinematic device, are critical to the success of the manufacturing system, and their accuracy is reliant on the accuracy of the DT. The act of using a DT system to program a robot (or other motion system) based only on the virtual models available in the DT is referred to as Off-Line Programming (OLP). OLP is heavily used on industrial robotic systems containing many robots performing complex tasks, as programming these systems by hand can be exceedingly difficult and time consuming. The OLP tool used in this research was Process Simulate.

3.2.3 Virtual Commissioning

Virtual Commissioning (VC) is a logical extension of OLP applied to systems containing logic driven components, such as Programmable Logic Controllers (PLCs). As described in Guerrero et al's Virtual Commissioning with Process Simulation (Tecnomatix),VC allows users to couple the kinematic model portion of the DT with a simulation of the logic that controls the system [23]. In a physical system, this logical component is often provided by a PLC. In the DT, the control logic driving the cell can be provided by a simulation or emulation of a PLC, or a physical PLC can be coupled with the DT to provide control input. These two types of VC are called Software in the Loop (SiL) and Hardware in the Loop (HiL), respectively. VC allows controls and robotics

engineers to test their control logic and robot programs in one integrated DT environment before deploying any software to the physical system.

3.2.4 Motion Capture

Motion capture systems are traditionally used in filmmaking studios to record the motions of actors in 3D space for recreation in software later. We have proposed using these same systems to track the motion of physical components in a cell for reconstruction of this motion inside a DT of manufacturing. The motion capture tool used in this research was an Optitrack motion capture system. This system uses multiple IR cameras with emitters to record the location of retroreflective tracking balls attached to rigid bodies in the physical system. The motion capture software, Motive, is then used to record and export the exact location of tracked bodies inside the physical system during manufacturing operations. This information can then be used to reconstruct the paths of robots, conveyors, or other kinematic devices. The reconstructed paths can then be compared against the planned paths from the DT to evaluate the accuracy of the DT and the OLP solution. This is particularly relevant on paths where locational accuracy is paramount to production quality.

3.2.5 3D Scanning

3D scanning is a similar technology to motion capture, relying on optical systems to determine locational data for points on an object [53]. A 3D scanner uses an optical system, usually based on IR lasers or blue light cameras to capture the location of point in a space and exports those points as a "point cloud" for later use. 3D scanners are often used for geometry reconstruction of parts or product data but have also been used at room or factory scale for reconstruction of a physical layout. When used in this manner, 3D

scanners can generate relatively accurate models of entire workspaces for later use in layout or path planning applications.

The 3D scanning system used in this research was a Faro scanning system. This Faro scanner was capable of producing scans at room scale with a mean point error of ~1mm across the entire FF and AFP workspaces. This point cloud data was used to assess the accuracy of DT layout for both labs. In the AFP lab, this point cloud data was also used to assess the accuracy of the machine model used in the DT. The results of these comparisons are discussed in chapter 5 of this thesis.

3.2.6 IoT and Sensing

Internet of Things (IoT) and Industrial Internet of Things (IIoT) technologies are umbrella terms describing data collection and analytics for many distributed components in a system [54]. IoT often refers to the more general consumer use case for data collection, where IoT devices may be smart home devices, appliances, vehicles, consumer goods, etc. IIoT systems apply the same concepts to industrial systems, where assets may be manufacturing systems, robot arms, machines, sensors measuring process parameters, etc. Data from these sources are then aggregated and contextualized into large data pools, often on cloud storage platforms. Data can then be processed and analyzed to develop trends and implications of that data to derive manufacturing knowledge and potential process improvements. In this research, IIoT techniques were utilized in both the FF and AFP labs to record and characterize processes and provide data and analysis back to the DT. The DT is then capable of utilizing this data to reconstruct actions taken in the physical system.

3.2.7 Edge Computing

Related to IIoT, Edge Computing is a tool for data processing that allows computationally intensive operations to be performed in real time as data is collected, without having to send that raw data to a cloud environment [54]. Edge computing refers to performing preliminary data processing operations "on the edge" using hardware located in or next to manufacturing systems, often directly interfacing with sensors or data collection points. Locating the processing capabilities directly next to the data sources allows for data to be processed in real time as the data is collected, and relay only the necessary data back to the cloud infrastructure. This real time processing of data also allows for edge devices to inform process changes in the cell, based on data incoming data. One application where edge computing has seen use is in computer vision, where the results of the vision analysis need to be relayed to the system in near real time. In this research the use of edge computing is limited, however edge computing capabilities are being investigated in the FF lab to allow for faster data processing and more efficient data use.

3.3 DIGITAL TWIN MODELING FOR FUTURE FACTORIES

The DT modeling tool Siemens Process Simulate was used to create a dimensionally and logically accurate DT system of the FF lab. The modeling of the system was divided into three steps: physical layout, kinematic modeling, and logical modeling. This breakdown allowed the researchers to mature the DT model at each step, building on the developments of previous steps. These steps are described in detail below.

3.3.1 Physical Layout

The design of the Future Factories system was planned using DT tools. As a first step, 3D CAD models were retrieved for all standard components of the FF cell, such as

robots, tables, actuators, conveyors, etc. Additionally, models were constructed for the walls and fencing that would define the boundary of the cell. These CAD models were loaded in and located inside the Process Simulate model to develop the outer extents of the cell. The additional off the shelf components were then loaded into the simulation, and multiple layout alternatives were developed using the layout tools inside Process Simulate. These layout alternatives are shown in Figure 3.3 below.

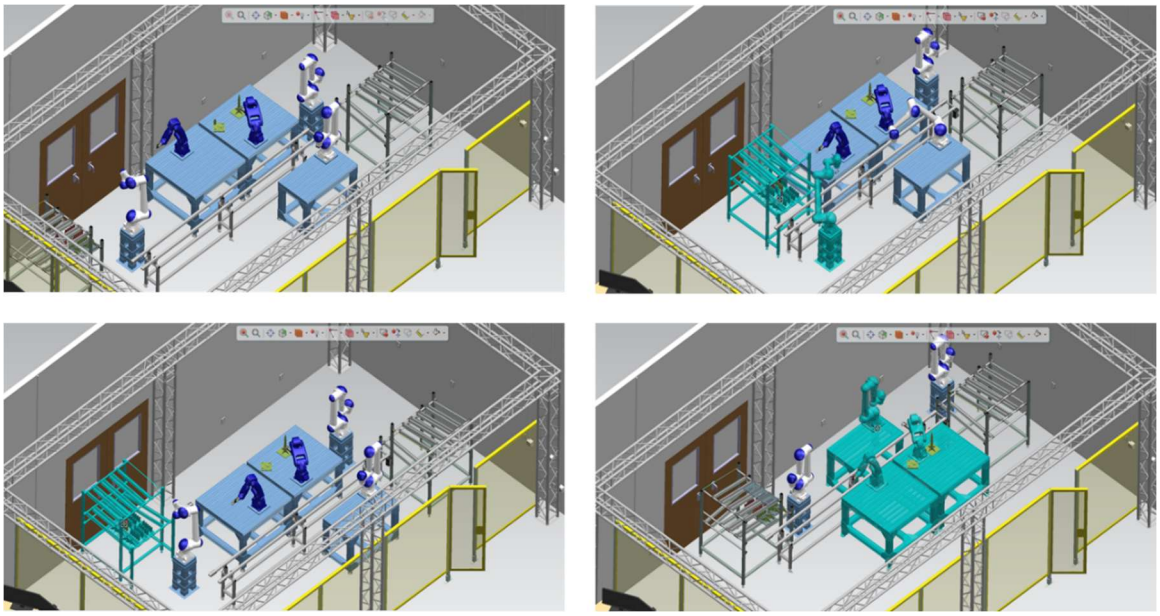

Figure 3.3. Various Layout Alternative Considered for the Future Factories Lab.

Simple simulations were then created using the DT tool for each layout, and key indicators like robot reach and collision analysis were used to narrow down the design space to a final layout design (section 3.2 of this chapter describes the process required to define the kinematics for these tests to be performed). The final selection for the FF layout consists of 2 GP8 industrial robots from Yaskawa for fast assembly tasks, 3 HC10 human collaborative robots from Yaskawa for long reach material handling tasks, and 4 conveyor belts that create a loop to allow for components to be passed back to the beginning of the process for rework and reuse of conveyor trays. Additionally, appropriate safety fencing

and mounting tables and pedestals were used to ensure the safe operation of the cell and solid mounting of robots in accurate positions. A steel gantry was also included in the design of the cell to provide a mounting location for motion capture cameras and other imaging devices. The final planned layout of the FF lab is shown in Figure 3.4.

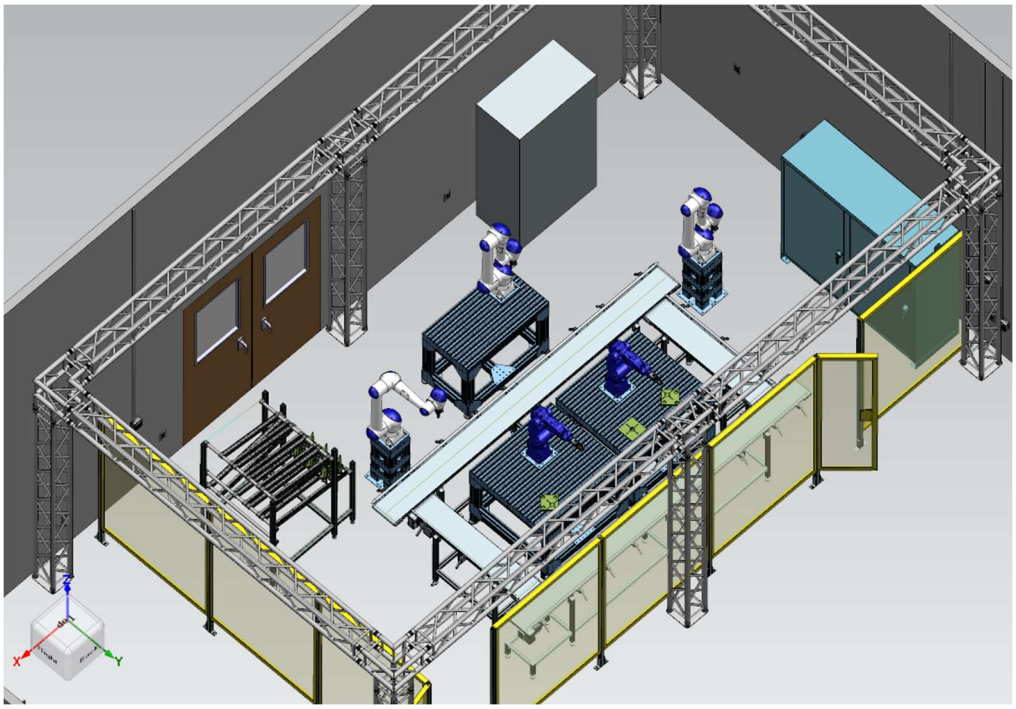

Figure 3.4. The Final Design of the Future Factories Layout

As previously discussed, the FF lab was intentionally designed as a flexible manufacturing system. The selected layout shown above has several benefits that allow for flexible manufacturing. The robots are positioned such that the two GP8s can interact with each other and their neighboring HC10s, and the HC10s are positioned to be able to reach various key locations to move materials where they need to be based on the cell status. Fixtures can be mounted to the two main worktables, and an alternative worktable is provided for the middle HC10 to position parts for assembly or inspection if required. This

layout results in a highly flexible manufacturing system with many possible manufacturing paths and processes to yield a finished part.

3.3.2 Kinematic Definitions

The final layout of the FF lab is essentially an assembly of CAD parts in a specific location. To transition from this static assembly to a dynamic simulation of a process, the kinematics of all of the moving components in the cell had to be defined. This includes kinematic definitions for robots, grippers, other actuators, and conveyors. In Process Simulate, kinematic components are defined by specifying solid bodies as kinematic links and defining joints that act along specific axes to relate the relative positions of these links to each other. This network of links and joints creates a kinematic tree for the component, which describes the dependencies of joints and links in the structure. An example kinematic tree and associated robot arm is shown in Figure 3.5.

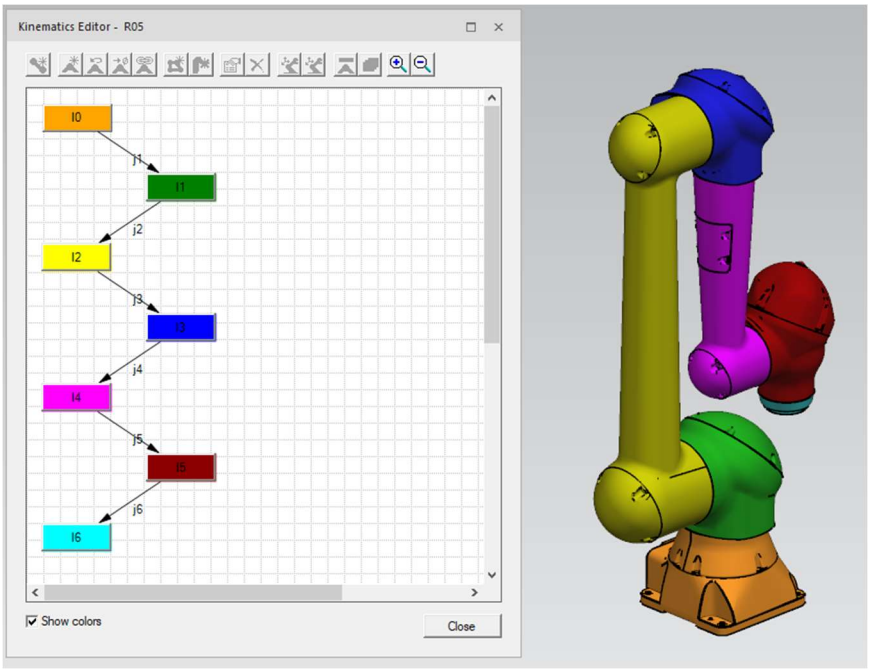

Figure 3.5. The Kinematic Tree of a Yaskawa HC10 Robot in Process Simulate

In the kinematic tree shown above, the links are represented by the colored bodies and associated colored nodes in the tree, and the joints are the labeled arrows between those nodes. Joints are defined by a parent and child link, and a prismatic or revolute axis on which they act. In the tree above, J1 is defined as a revolute joint between the first two links of the robot and acts along the center of rotation of the base of the robot. If the value for J1 is changed, all children of J1 will be transformed by the value described by J1, that is, rotating the robot around its base. For further illustration, the joint definition for J1 is shown in Figure 3.6 below. Similar joint definitions were performed for all subsequent joints in the HC10 robot to build a functional kinematic model. This procedure was then repeated for all other kinematic devices in the cell, including robots, conveyors, end of arm tooling, and other actuators.

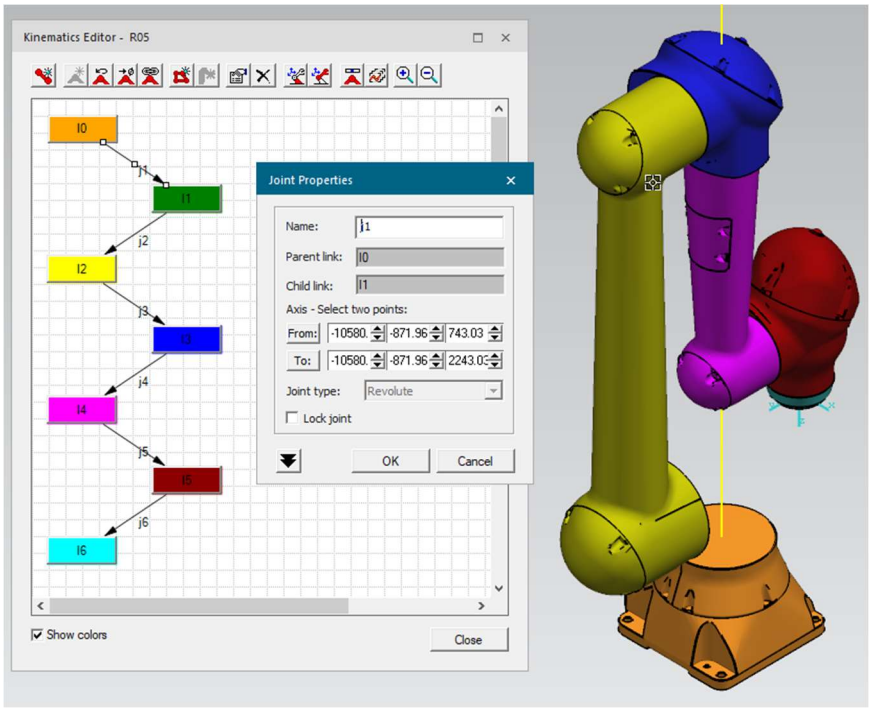

Figure 3.6. The Joint Definition for J1 of the Yaskawa HC10 Robot

After defining an accurate kinematic model of the robots, an inverse kinematics model can be applied to solve for the joint values for any given target position. This inverse kinematics algorithm is provided by the Process Simulate software and is required to treat the kinematic models as robots for simulation. To apply inverse kinematics to a robot model, it must first be defined as a robot inside the model and have a BASEFRAME and a TOOLFRAME. The BASEFRAME specifies the base of the robot, the location to which all targets are referenced. The TOOLFRAME specifies the location at the end of the arm to solve for. These two properties are specified easily inside Process Simulate using the kinematics editor, as shown in Figure 3.7. At this stage, special care was taken to ensure that the TOOLFRAME on the end of the arm matches the definition provided by Yaskawa for programming their robots. The inverse kinematics capability was then verified using the Robot Jog tool inside Process Simulate. A similar procedure was then followed for the GP8 robot model.

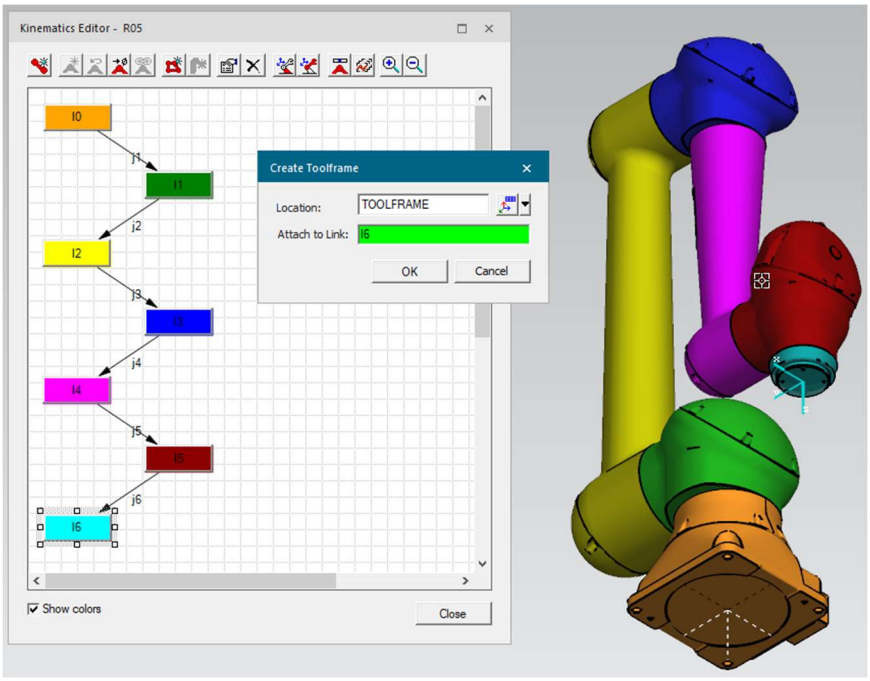

Figure 3.7. TOOLFRAME Definition for the HC10 Robot Model

3.3.3 Logical Component Definitions

Simulation of the FF lab requires more than just simulation of kinematic movements inside the cell. The FF lab is a sophisticated electromechanical system with many logical components that require simulation and validation as well. These components range in complexity from a simple digital solenoid to a complex PLC that controls the behavior of most of the components in the cell. Simulation of these components is handled by two systems. A PLC simulation software called PLCSim Advanced is used to simulate the behavior of the S7-1516F PLC used for main cell control, and "Smart Objects" inside Process Simulate are used to simulate the behavior of the other less complex logical devices in the cell. The PLCSim Advanced instance is then connected to the Process Simulate model using the PLCSim Advanced API to couple the two simulations together and to provide control logic to the Process Simulate model. This creates a Software in the Loop Virtual Commissioning model. Figure 3.8 describes the architecture of this virtual commissioning system. The configuration and testing of this virtual commissioning model is described in more detail in chapter 4 of this thesis.

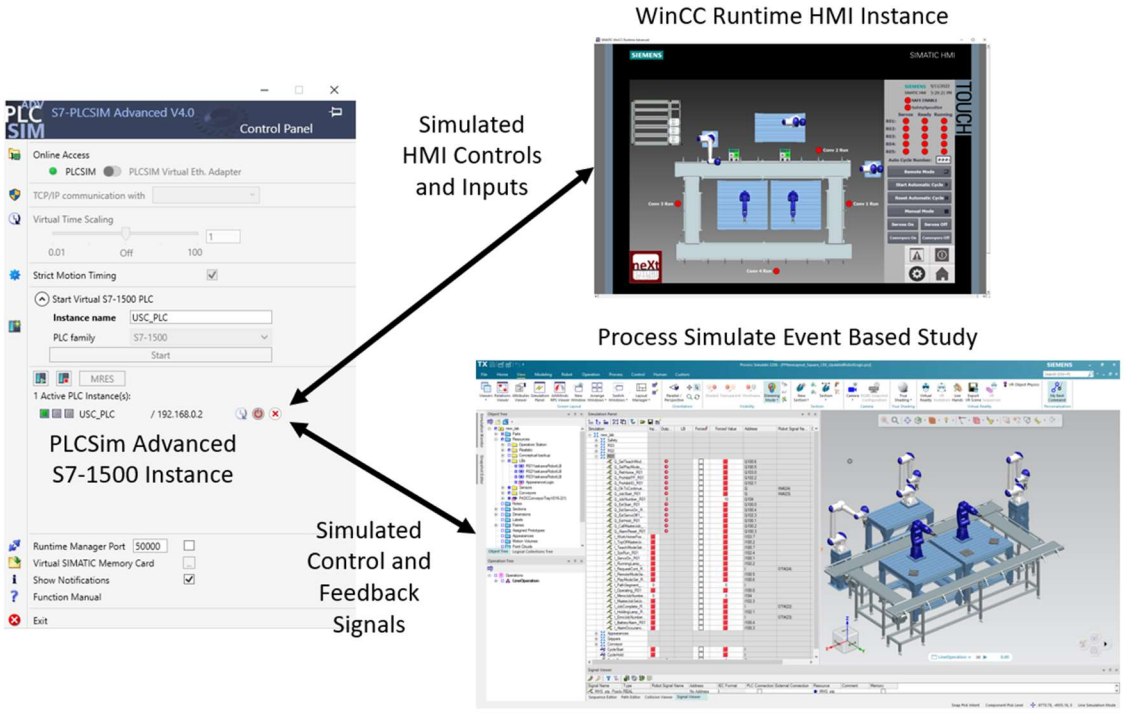

Figure 3.8. The System Architecture of the VC Model

Other logical components in the cell, such as robots, grippers, conveyors, and sensors are all simulated using Smart Objects inside the Process Simulate model. Smart Objects are components inside the Process Simulate model that contain logical behavior. They can perform basic logic operations, as well as trigger the device to perform some action, like closing a gripper or setting a conveyor speed. Smart Objects contain a list of "entries" and "exits" that act as input and output signals for the device. The Smart Object performs operations using these entries and provides feedback to the controller on the exits. Based on the current state of the entries and parameters of the device, various actions may be performed. The actions performed can utilize previously defined kinematics to move components as required and specified by the kinematic model. An example overview of a smart device is shown in Figure 3.9. Smart Objects were defined for all components inside the cell that required logical behavior, and these definitions were created to match the

behavior of the device as closely as possible. Often this definition step required information from the device manual or device manufacturer.

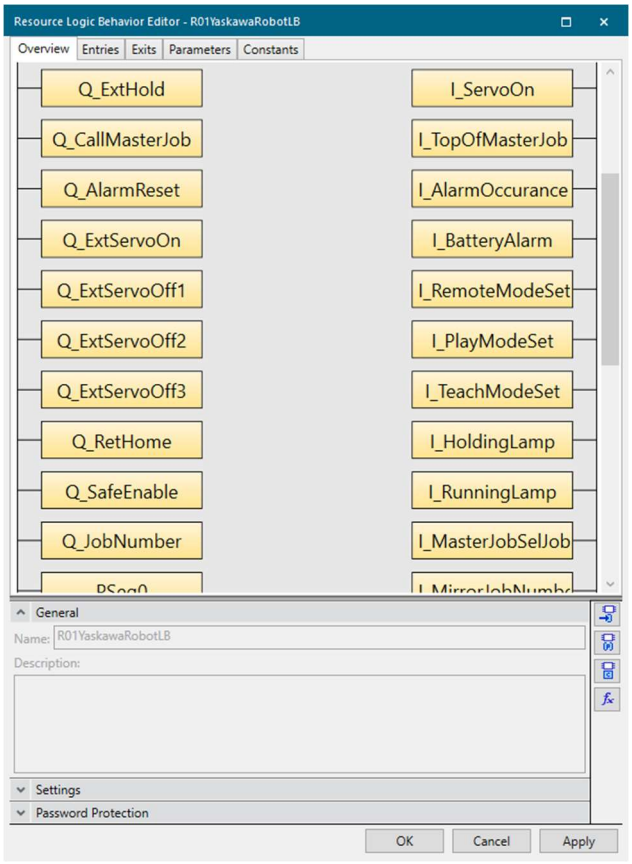

Figure 3.9. The Smart Object Overview for a Robot in the FF Lab.

Sensors inside Process Simulate require an additional definition step. These components require the definition of a sensor type and, if applicable, a detection zone. Proximity sensors are heavily used inside the FF lab, these are easily defined inside PS by modeling the detection zone of the sensor using a CAD system and configuring a proximity sensor inside the model with that detection zone. The sensor is then configured to change the associated signal when an object enters the detection zone. A similar process is followed for any photoelectric sensors, except the detection zone is simply a line in this case. Other sensor types, such as encoders or other joint sensors are created using Smart

Object parameters such as “Joint Value Sensors” that can record the current value of a joint from the kinematic definition of the component.

3.3.4. Notes on Accuracy

As previously discussed, the DT model of the FF lab inside Process Simulate was used to design the physical layout of the system. Many components of the 3D layout inside the DT model were based on supplier dimensions and CAD models, whose dimensions should be fairly accurate to the delivered product. The accuracy of these individual components is discussed more in chapter 5 of this thesis. The location of these components inside the physical system is based on the design discussed in section 3.1 of this chapter. When building the physical cell, the experimenters tried to maintain accuracy as close as possible to the DT design, but some variance is expected. The accuracy of this layout was assessed with multiple methods, including 3D scanning, and is discussed in more detail in chapter 5 of this thesis. At this stage, the researchers expected the DT cell model to have sufficient dimensional accuracy for OLP and VC.

The accuracy of the kinematic definitions in the cell is also of importance to the performance of the DT, and for OLP of the FF lab. For simple devices, such as grippers, the kinematic definitions were derived directly from the original CAD models, and joint limits were derived from measurements of the physical system. This yielded sufficient accuracy for these devices. For more complex devices, such as robots, the kinematic definitions were based directly on the CAD models provided by the supplier, and joint limits were taken from the documentation provided with the robot. These kinematic definitions should very closely match the definitions of the real system; the accuracy of these robot definitions is discussed more in chapter 5 of this thesis.

The accuracy of logical devices in the FF lab heavily impacts the behavior of the VC DT model. As such, care was taken to ensure Smart Objects match the behavior of their physical counterparts as closely as possible. Many of the Smart Object in the lab are derived directly from documentation of the real device, and as such their behavior should closely match the behavior of the real device. Most devices, such as conveyors and solenoids, are simple enough that variance in behavior is not expected. Some more complex devices, such as the robot control Smart Objects, derive their logic from experimentation with the actual system in addition to documentation. These devices may have behavior that varies from the actual system behavior. The accuracy of these behaviors in a VC environment is discussed further in chapter 5 of this thesis.

3.4 DIGITAL TWIN MODELING FOR AUTOMATED FIBER PLACEMENT

Process Simulate was also used for DT modeling for the AFP lab. Unlike the FF lab DT, the DT of the AFP lab was purely a kinematic DT, containing no logical component. This DT was designed to allow for machine and robot programming, and reconstruction of machine movement after layup, and as such did not require a logical component. A similar process was used to model this lab, however in this case, the DT was modeled after an already existing physical system, containing both the IMT Lynx fiber placement machine and the ACSIS robotic inspection system. Working from an already built physical system presented additional challenges when creating the physical layout and kinematic definitions of the cell. These steps are described in the sections below.

3.4.1 Physical Layout

The layout of components in the DT of the AFP system was determined by the already existing physical system in place in the AFP lab. CAD models for the Lynx

machine in the cell were obtained directly from IMT and used to position both the machine and 7th axis mandrel in the Process Simulate model. The geometry in these models was used directly without modification to create as accurate of a DT model as possible. Additionally, the AFP tool used in this experiment was also mounted on the mandrel according to the mounting dimensions in the provided CAD.

Other components in the cell, such as fencing, walls, and columns were located using traditional measurement and layout methods, such as tape measures and levels. CAD models for these components were obtained directly from vendors, when possible, or in some cases were modeled from physical measurements of components in the cell. The exact location of these components is less critical to the overall integrity of the DT. As such, these lower precision methods could be used without loss of functionality.

The ACSIS composites inspection system was also included in the physical layout of the AFP lab. The ACSIS system utilizes a KUKA KR120 robot mounted on a linear rail with four profilometers mounted on the end of the arm that can scan the surface of an AFP manufactured part for defects. Models for the robot and rail were obtained directly from KUKA, through their website. Models for both the robot and rail were natively compatible with Process Simulate and already contained accurate kinematic definitions to be used in the simulation. Models for the scanning end of arm tooling were derived from drawings of the profilometers obtained from Keyence, and from physical measurements of the mounting plate on the end of the arm. These methods were able to accurate reproduce the dimensions and location of the end of arm tooling on ACSIS. The location of the ACSIS system inside the cell was determined by physical measurement of the rail location relative to the mandrel of the Lynx machine. Later this position was refined using tooling results

from ACSIS. The complete physical layout of the AFP lab in the DT model is shown in Figure 3.10 below.

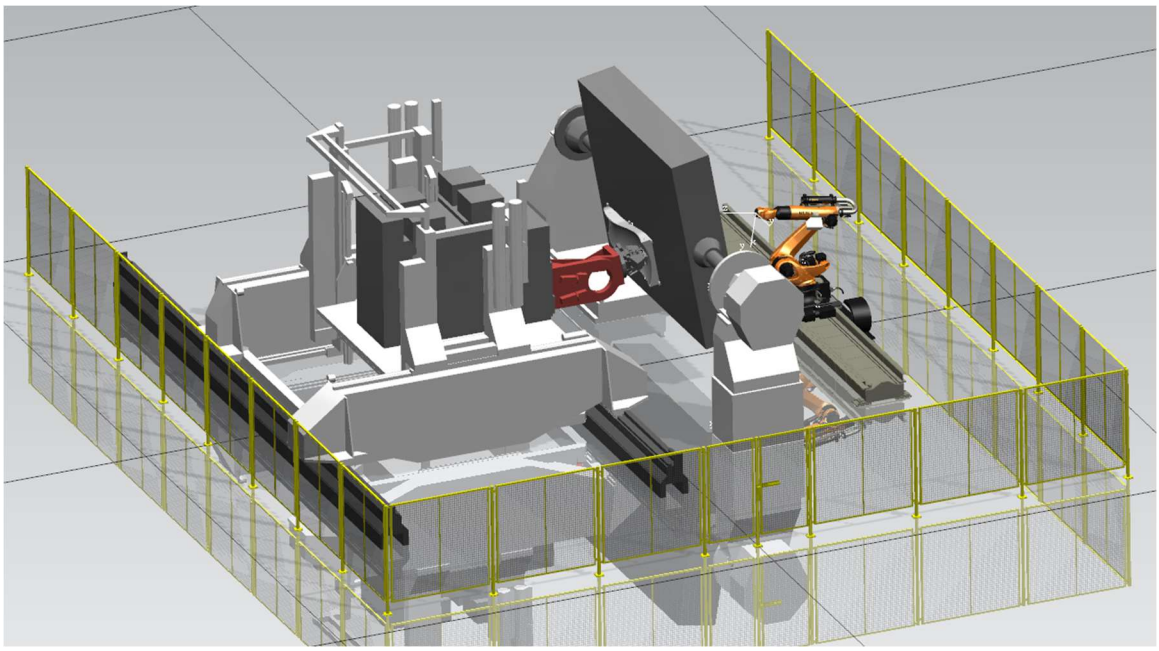

Figure 3.10. The AFP Lab Modeled in Process Simulate

3.4.2 Kinematic Definitions

Using the same procedure described above for the FF lab, accurate machine kinematics for the Lynx machine were created inside Process Simulate. Elements of the machine assembly were associated with links of the kinematic tree, as shown in Figure 3.11. These links were then connected with joints that describe how they can move with respect to each other. These relations are shown in the kinematic tree. The exact location, limits, velocity, and acceleration values for each joint were set to IMT specifications. Some additional specifications required to inform Process Simulate on how to simulate the motion of the machine and how to select inverse kinematic solutions for each location were included with the model in a separate file called "**motionparameters.e.**" The content of this file was built according to the syntax described in the Process Simulate help

documentation and should ensure a closer match of kinematic behavior between the DT and physical system. As described in section 3.2 of this chapter, a TOOLFRAME and BASEFRAME were defined for the AFP machine to allow for inverse kinematics calculations. This step also defined the AFP machine in the Process Simulate study as a robot. These frames were defined at the tool center point (TCP) of the rollers, as well as the origin of the machine as specified by IMT.

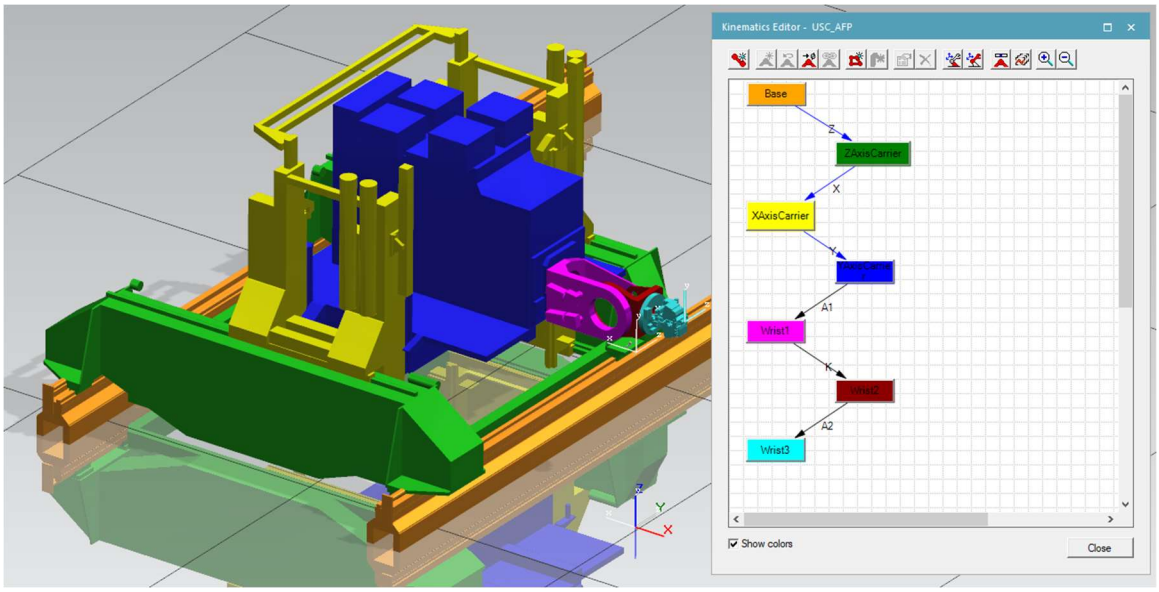

Figure 3.11. The Kinematic Model of the AFP Machine Inside of Process Simulate

The machine's mandrel and tool were then added to the DT environment, using the same procedure as described above. Kinematics were added to the mandrel, and the mandrel was then added as an external axis to the AFP system. The resulting system is shown in Figure 3.12.

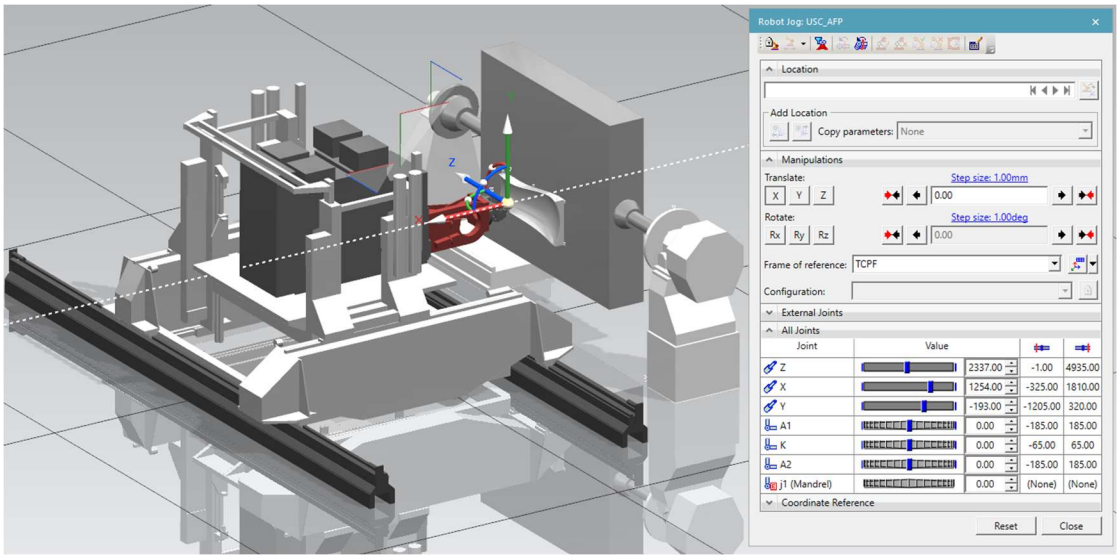

Figure 3.12. The Finalized Kinematic Model for the AFP System and Mandrel

As noted previously, kinematics for the ACSIS inspection system were obtained directly from the original equipment manufacturer, KUKA. KUKA's models included kinematic definitions for both the robot and the rail, as well as accurate TOOLFRAME and BASEFRAME definitions for the robot arm. To complete the kinematic configuration of ACSIS in the DT, the rail was added as an external axis to the robot inside the Process Simulate model, and user tool and base definitions were retrieved from the robot to further enhance the accuracy of the DT model.

3.4.3 Notes on Accuracy

AFP is a dimensionally sensitive process, so creating an accurate DT of the system was critical to the performance of the system. As discussed previously, CAD models provided by IMT were used for definition of both the physical layout and kinematics of the Lynx AFP machine. The location of the mandrel and machine in the cell was directly based on the models provided by IMT, and as such was expected to closely match the installed configuration. The accuracy of these placements in discussed in chapter 5 of thesis, with comparisons based on 3D scanning results. The kinematics of the Lynx machine are

assumed to be accurate, as their definitions are based on CAD models and manuals provided by IMT for the machine. The accuracy of these definitions is also discussed in chapter 5 of this thesis. The kinematics of the ACSIS robot and rail were provided directly by KUKA in a compiled format that is compatible with Process Simulate and are assumed to be accurate. Frame definitions for the robot and rail were imported directly into the model from the robot system.

The placement of additional components in the cell, such as fencing, walls, and the ACSIS inspection system relative to the rest of the cell was based on hand measurements taken with tape measures and rulers. We expect some level of error with these measurements. The final accuracy of these placements does not affect OLP or motion reconstruction inside the cell, and as such their impact on DT performance is minimal. The accuracy of these placements relative to the machine is discussed in more detail in chapter 5.

3.5 CONCLUSIONS ON DIGITAL TWIN MODELING

The creation of DT models for the FF and AFP labs was successfully completed using Siemens Process Simulate as DT backbone. Additional customization and modeling was performed to enhance the capabilities of the DT models for both the AFP and FF labs. These models provide many elements of a DT of manufacturing in one integrated environment. Both models provide a positional layout of elements in the manufacturing workspace, kinematic modeling of moving components, modeling of logic driven components, programming and virtual commissioning capabilities, and an environment for analysis of real results from the manufacturing system. The use of and results from these DT models is discussed in Chapter 5 of this thesis.

CHAPTER 4

IMPLEMENTATION OF DIGITAL TWIN SYSTEMS FOR FUTURE FACTORIES AND AUTOMATED FIBER PLACEMENT

The DT models described in chapter 3 of this thesis were implemented across both the Future Factories lab and the Automated Fiber Placement lab, incorporating many of the techniques previously described. This implementation presented many challenges and opportunities to prove the effectiveness of the DT. This chapter outlines this implementation, including the development of physical platforms, DT systems, and many challenges we encountered along the way. Additionally, this chapter describes the testing methodologies used to validate the effectiveness of the DT systems and provides many examples where the DT offers additional capabilities that improve the operation of the system.

4.1 DIGITAL TWIN IMPLEMENTATION FOR FUTURE FACTORIES

4.1.1 Physical System Development

As described in chapter 3 of this thesis, the FF lab was designed using DT tools with the intent of providing a test bed for various advanced manufacturing techniques. Various possible layouts were assessed in the DT tools before a final approach was selected. The selected layout was then built in the physical lab space at the McNair Center. Traditional measurement tools such as tape measures and scales were used to align the physical construction with the planned layout. Tables and fixtures were secured to the floor of the lab using concrete screws, ensuring that their relative positions would be fixed.

Safety fencing and an overhead gantry were then erected around the lab, to complete the space. The accuracy of the physical construction could then be assessed using Off-Line Programming and 3D scanning techniques.

After most of the large components in the cell were placed, logic driven components such as the PLC, HMI, and IO devices were mounted in reasonable locations inside the cell. IO modules were mounted on standard rail close to robots, to handle incoming IO from end of arm tooling and the robots themselves. The robot controllers were placed under the two tables in the middle of the lab, allowing easy access for wiring and IO connections. A main controls cabinet was placed in the back of the lab, allowing for power distribution and drives for the conveyors, as well as a secure location to mount the main PLC. A large amount of space was left unused inside this controls cabinet, to allow for future expansion and integration of additional devices should the need arise.

Safety devices were also installed in the FF lab to ensure the safety of students and staff, as well as to protect the equipment in the lab space. Emergency stop buttons were placed at strategic locations in the lab, including one on the controls cabinet inside the cell, one at the operator station, and one on each robot teach pendant. All emergency stops were wired as normally closed circuits (in accordance with SIL 3 standards) and each was programmed using the S7-1516F PLC to be able to independently stop all moving devices in the cell, including robots, conveyors, and end of arm tooling. All emergency stop systems were designed to the SIL 3 standard. Door sensors were also installed at each entrance to the enclosed area of the cell. These sensors were programmed using the safety PLC to limit the maximum speed of the robots in the cell to 250mm/s when the doors are open.

4.1.2 3D Scanning

After the physical construction was completed, 3D scans of the space were taken to measure how accurately the real cell was built to the original virtual model. 3D scans were taken using a Faro Focus 70M scanner, which can produce room scale scans with an accuracy of ~ 2mm over a 20m range. This level of accuracy is sufficient to determine roughly how accurately the components in the cell were positioned, however more sophisticated measurement techniques were required to determine the relative locations of robots for collaborative robot operations and other position critical robotic tasks. These positions were eventually determined using the robots themselves as probing devices, and by measuring the accuracy of OLP tasks.

20 scans were taken of the FF lab from 20 different scanner positions. These individual scans were then composited into one 3D point cloud through a process called scan registration. The registration was performed using Scene, a proprietary software provided by Faro. During the registration process, a maximum point error of 1.1mm was reported, indicating that the individual point locations inside the point cloud should vary from the physical by no more than 1.6mm. The final point cloud used in this implementation consisted of 87 million points. An image of the entire final point cloud is shown in Figure 4.1 below. As previously discussed, this data was then used to compare the physical construction of the FF lab to the planned layout. The results of this comparison are discussed in chapter 5 of this thesis.

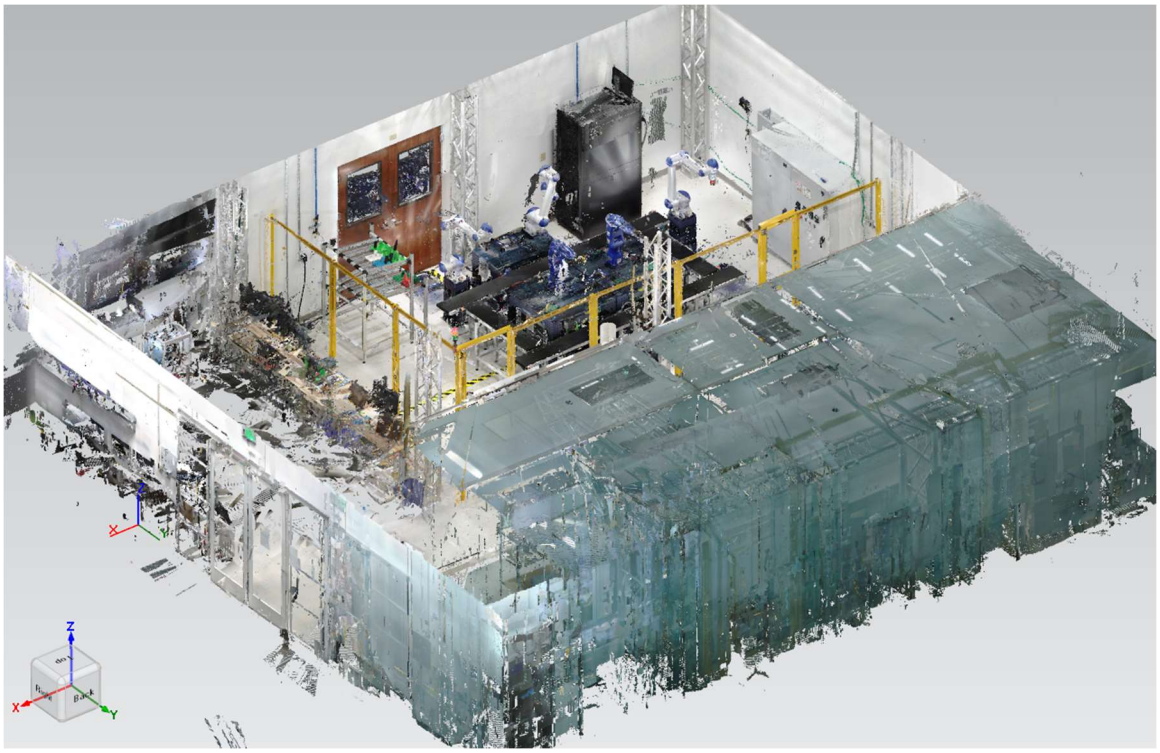

Figure 4.1. The Point Cloud Produced by the 3D Scan of the FF Lab

4.1.3 Virtual Commissioning

To complete the commissioning of the FF lab, a Virtual Commissioning (VC) process was used. VC, as described in chapter 3 of this thesis, refers to the process of building and testing automation control logic and programs in an integrated environment that allows for communication between a virtualized PLC and other cell components. The VC environment used in this project is described in more detail in chapter 3 of this thesis. The PLC used for the FF lab was a Siemens S7-1516F fail-safe PLC. This PLC was virtualized using Siemens PLCSim Advanced software, and the physical and other logical components of the cell were simulated using Siemens Tecnomatix Process Simulate.

4.1.3.1 PLC Programming

PLC programming for the FF lab was performed using the Siemens Totally Integrated Automation (TIA) Portal programming software. TIA Portal allows users to

author PLC programs in many languages, including ladder logic (LAD), functional block diagram (FBD), statement list (STL), and structured control language (SCL). The use of LAD or FBD was required for safety programming, as these two languages are the simplest to understand and least prone to programming errors, although they also are the least capable of the programming languages available. LAD was used for the safety programming on this project. SCL was used for the bulk of the control logic authored in this project, as it is the most versatile, and very similar to many text-based object-orientated languages that the researchers were already familiar with.

TIA Portal allows for a logical structuring of PLC code for easy editing and reusability. While this thesis will not cover a complete description of the PLC programs authored for the FF lab, a brief description of the programming structure used in the lab is provided in Figure 4.2 below.

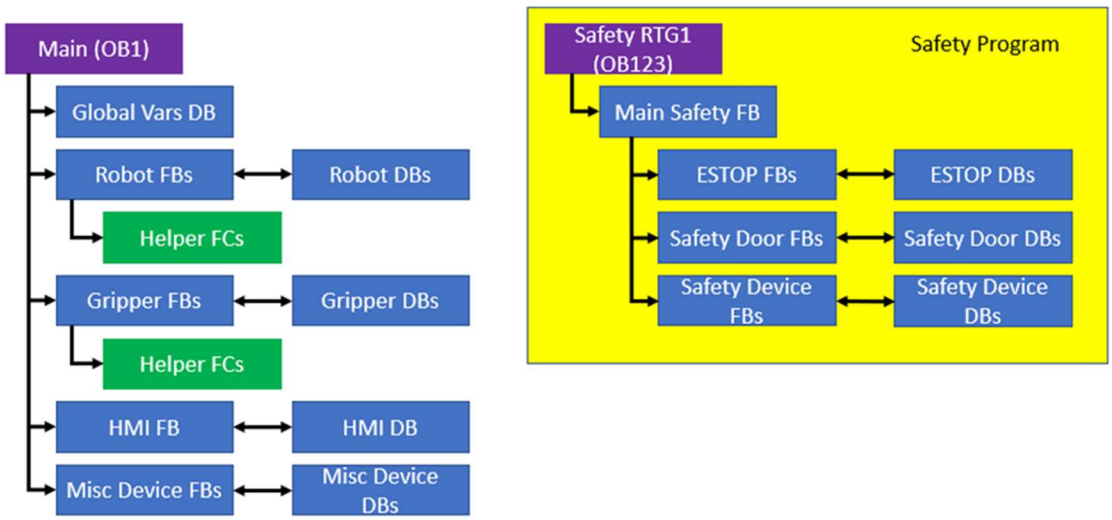

Figure 4.2. The PLC Programming Structure Used in the FF Lab

The TIA Portal project used in the FF lab consists of Organizational Blocks (OBs), Function Blocks (FBs), Function Calls (FCs), and Data Blocks (DBs). OBs specify a block of code that will run under a certain condition in the PLC. OB1, the “main” OB, is a cyclic

execution block that runs every execution cycle of the PLC. OB1 was the main OB used in the FF Lab PLC program, however other OBs, such as cyclic interrupts, were used for various edge cases. OB1 contains calls to nearly every other piece of code that is executed in the project. FBs contain the bulk of the PLC code in the FF lab project. FBs, when invoked with a corresponding DB that contains the data for the FB call, act somewhat similarly to classes in traditional object-oriented programming languages. FBs in the FF lab contain logic pertaining to a specific type of device in the FF lab and can be reused for every instance of that device in the project. In the FF project, FBs were used for conveyors, grippers, robots, and any other device that requires associated logic in the PLC project. FCs are similar to FBs, but do not contain any instanced data, and as such do not contain any information about a represented object or device. FCs simply contain logic that will be executed with the FC is invoked. In the FF project, FCs were used for simple operations that were reused and did not require any associated data. Finally, DBs are objects inside a TIA Portal project that act as data holding objects, allowing a programmer to access data in a managed manner. DBs were used alongside FBs to describe data pertaining to a specific device in the cell. DBs were also used in a standalone fashion to hold data pertaining to the cell, such as operational status and other global variables.

The PLC used in the FF lab is a S7-1516F CPU, capable of executing fail-safe "safety" programs in addition to the main PLC program. This capability was used in the FF lab to manage safety devices and safety logic in the cell. The safety program exists in a mostly isolated environment from the main PLC program and requires that all inputs and outputs are from safety rated devices that are double evaluated, redundant, and fail-safe. As previously noted, the safety program for the FF cell was written in LAD, as required by

TIA Portal. The safety program runs in OB123, a specific OB for safety operations. Most of the safety code exists in FB1, which is invoked from OB123 and runs completely isolated from the remaining PLC code. OB123 also guarantees that the main safety loop will run every 100ms. This timing is user adjustable if a faster response time is required.

This PLC program was developed alongside the simulation model developed in Process Simulate and was tested at various points during development using a VC methodology. When a particular block of the PLC program was first coded, that block was integrated into the project and tested against the DT model using VC. This process allowed the researchers to find and correct many problems with the PLC program before the code was ever deployed to the physical system. Additionally, this process allowed the team to develop code for components that had not been physically integrated into the system yet, further accelerating the development process. The use of VC also allowed the team to develop and test PLC blocks remotely on the DT, with confidence that these new program blocks would perform identically when deployed to the physical system. Other components of the control logic in the cell were also validated using the VC based DT, including HMI screens and robot programs.

4.1.3.2 Off-Line Programming

The DT described in chapter 3 of this thesis also acted as a development and test platform for robotic programs in the FF lab. The DT allowed for rapid development of robotic operations in a simulated environment for use on the physical system. These operations included tasks such as picking and placing components, assembly, and inspection. Operations were created in the Process Simulate environment and consist of a list of target locations in 3D space for each robot. For the purposes of other ongoing

research in the lab, robotic tasks were purposely broken down into short operations that can be individually invoked by the PLC to achieve an overall goal. For example, the task to retrieve a parts tray from an incoming goods station and place that tray on a conveyor was split into two separate robotic operations: picking the tray from a designated position in the station and placing that tray onto the conveyor. These two operations are shown in Figure 4.3 below.

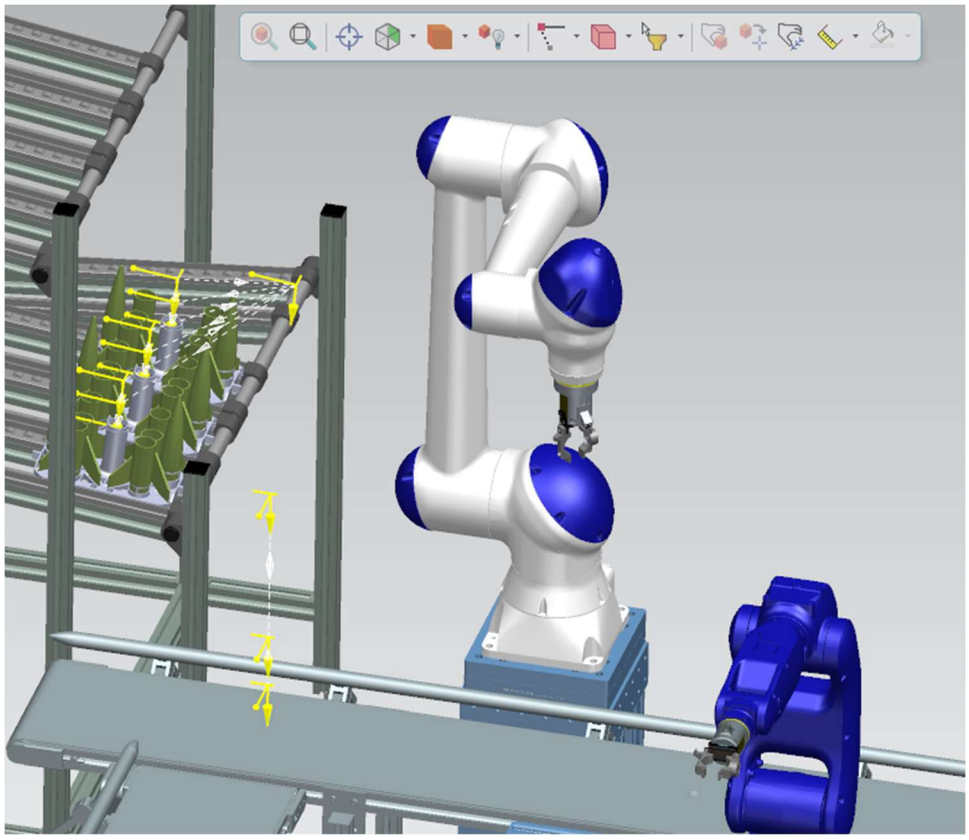

Figure 4.3. Pick and Place Operations for R01 in the FF DT Model

The locations for each robotic operation are driven based on the CAD of the system in the DT. Each location is specified interactively by the user inside Process Simulate, and these locations are based on the positions of the individual objects inside the DT. As such, the accuracy of the DT is paramount to the accuracy of the robotic programs. An example of location creation and edition is shown in Figure 4.4 below.

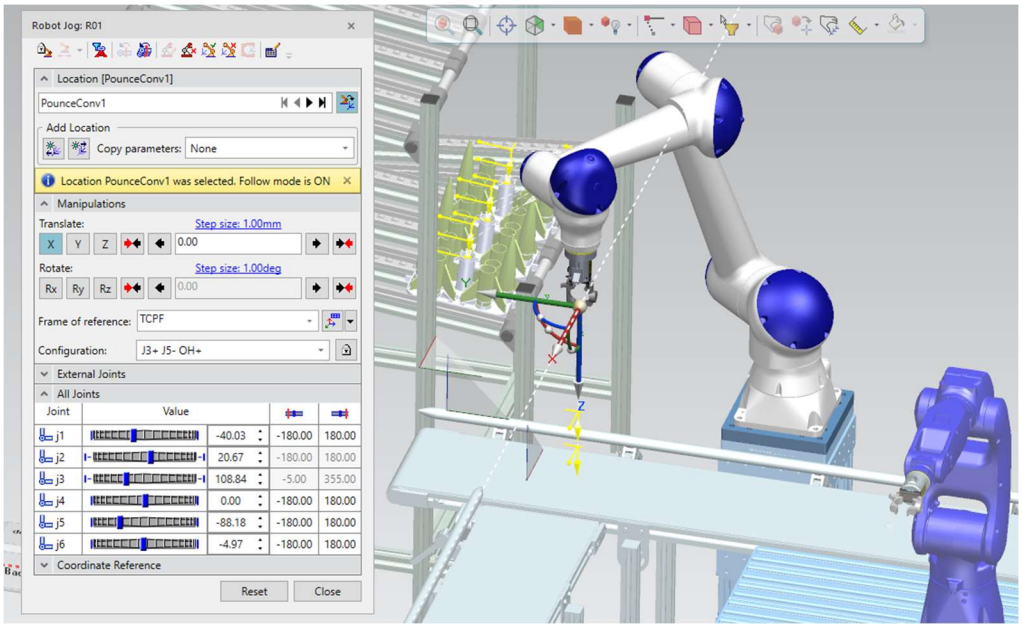

Figure 4.4. Location Creation Inside the FF DT Model

Each location inside a robotic program must also contain motion parameters that specify how the robot will move to the target. These properties were set using the Path Editor or the Teach Pendant inside the Process Simulate application. Examples of these properties include motion type (MOVJ, MOVL, MOVC, etc.), speed, acceleration, storage type, reference frames, etc. Additionally, the user must specify the kinematic solution, or "configuration", that the robot will use for each location. These configurations were specified by "teaching" a simulated configuration using the DT, which was then saved to native robot syntax inside Process Simulate. Examples of different robot configurations for the same target are shown in Figure 4.5.

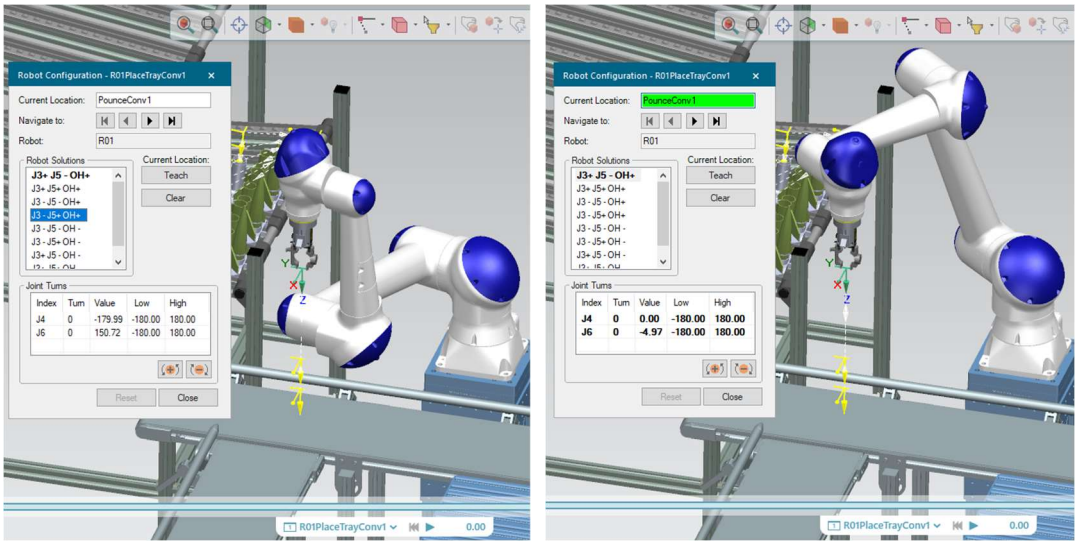

Figure 4.5. Different Robot Configurations Inside the FF DT Model.

Of key importance for OLP of these robotic tasks was specifying the correct reference frames for each location. For the Yaskawa robots used in the FF lab, these reference frames include specifying both the active UTOOL and UFRAME for every target. The UTOOL frame informs the robot of the tool offset from the flange of the arm, controlling the “commanded” point at the end of the arm. This UTOOL selection allows us to change the active Tool Center Point (TCP) during operations so different gripping modes could be utilized. In addition to specifying the UTOOL frame, the user also must specify the active UFRAME for each location in the robotic path. The UFRAME acts as an origin to which all specified locations are referenced. This ensures that so long as the UFRAME is calibrated correctly and all points are referenced correctly to the UFRAME, the resultant path will be accurate in the physical system. UFRAMEs function similarly to a work offset in the NC programming space, allowing the same path to be executed with the correct offset to position the path appropriately in the physical system.

UFRAMEs were calibrated on the physical system by performing a touch off procedure provided in the robot control software by Yaskawa. This touch off calibration

procedure was also used to validate the physical layout of the lab alongside the 3D scanning results. The calibration of UTOOLs and UFRAMEs using the robot itself allowed the researchers to establish the position of key landmarks in the cell (such as tables or fixtures) relative to the base of each robot. Custom designed 3D printed calibration tools were used for this touch off procedure. The tip of the tool was designed to be exactly 100mm from the flange of the robot, and directly centered on the axis of rotation of joint 6. This positioning allowed for a well-defined UTOOL for the UFRAME calibration procedure. The results of this calibration are discussed in more detail in chapter 5 of this thesis.

4.1.4 Motion Capture

The Optitrack based motion capture system described in chapter 3 of this thesis was used to record the motion of the robots in the FF lab and validate the accuracy and calibration of these robots. Motion capture tracking targets were placed on the existing calibration end effectors, as well as a custom end effector purpose designed for mounting the retroreflective motion capture targets. These two configurations for end effectors are shown in Figure 4.6.

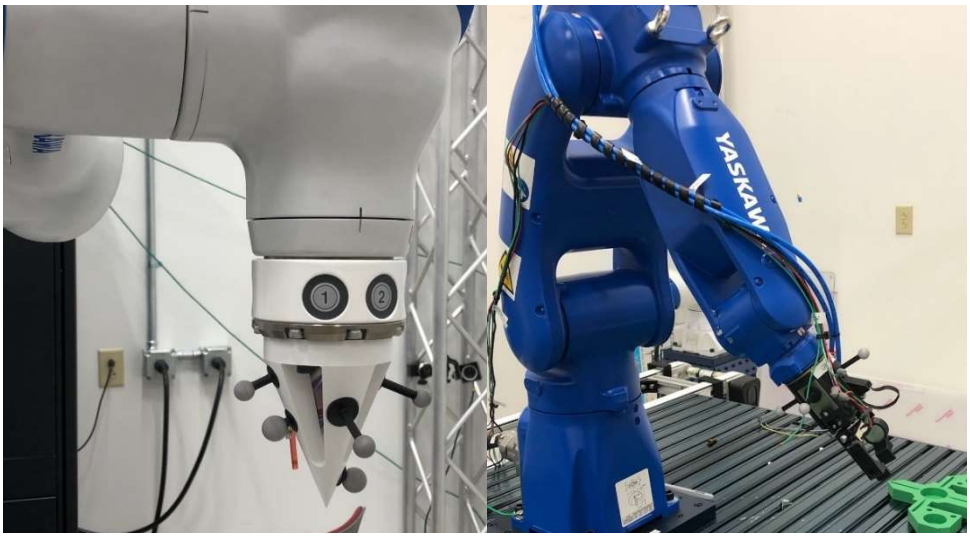

Figure 4.6. Two Motion Capture Target Configuration for End Effectors in the FF Lab

The offset of each tracking target relative to the TCP of the tool was measured in CAD and then recorded. A rigid body was then created for each tracked robot using the Motive motion capture software. The tracked point of each rigid body was placed at the TCP of the tool, allowing for an easy comparison against the original paths authored in the DT. The Optitrack system was then calibrated using the "wanding" procedure provided by the Motive software, resulting in a calibration accuracy of ~ ± 0.2mm. The calibrated motion capture system was then used to record the motion of the rigid bodies during robotic operations. The delta between the theoretical paths described in the DT and the actual recorded paths was then analyzed, and the results of this analysis are discussed in chapter 5 of this thesis.

Instead of using the existing robotic paths from the rocket assembly process, specific robotic paths were created for motion capture analysis. The choice to use purpose-built paths was made because the rocket assembly process utilizes little of each robot's available workspace, and the experimenters wanted to analyze the accuracy of the robot

across its entire workspace, including various orientations and configurations. Additionally, test paths were selected for simplicity, to make the analysis of the motion capture results after recording easier and faster. Two types of paths were developed for motion capture analysis: linear and planar. Linear paths simply move the robot end effector along a straight line with a fixed orientation. Planar paths consist of multiple linked co-planar linear paths to explore a larger portion of the robot's workspace. These paths are described in more detail in Chapter 5 of this thesis.

4.1.5 Edge and IoT Systems

Siemens Industrial Edge and Mindsphere systems were utilized to capture and record process data from the FF lab during manufacturing processes. Various sensors, such as potentiometers, accelerometers, and vibration sensors were placed at key locations inside the cell to measure process parameters. Additionally, the robotic systems in the cell were configured to report their current position and operational status back to the PLC. The S7-1516F was used to ingest this data from various sensors and other sources and make real time process decisions based on the incoming data, such as opening or closing grippers, or starting robotic operations. The Siemens Industrial Edge system was then used to collect this data from the PLC using the native S7 protocol over Profinet. The Edge system can then ingest this data and post the raw data to a MQTT data bus, hosted on the Edge device. The Edge device can then perform operations and calculations locally on the device using containerized applications developed by the research team, or applications from the Industrial Edge Hub (a global Siemens database of Edge applications). The Edge device then exposes both the raw and processed data to Mindsphere using a connector application. The architecture of the IoT implementation at the FF lab is depicted in Figure 4.7.

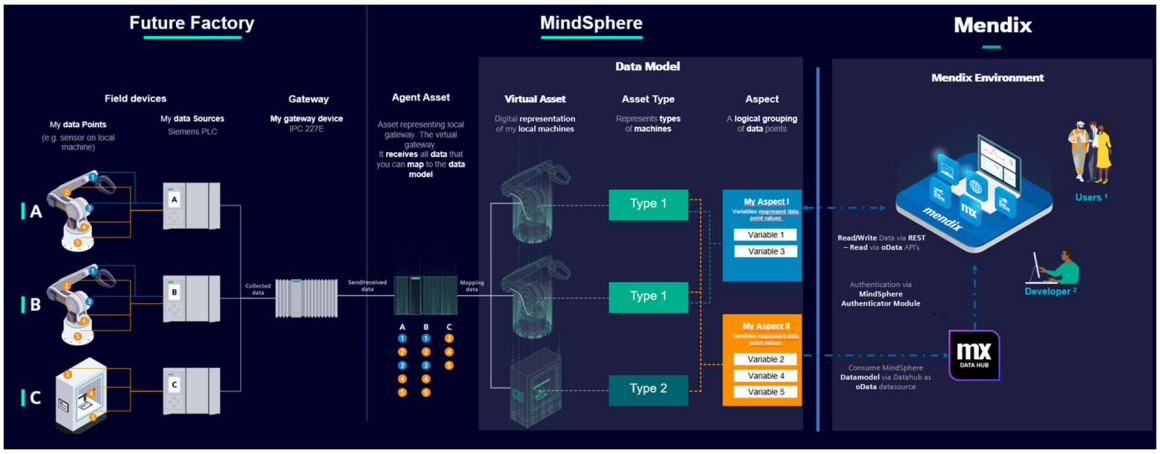

Figure 4.7. The IoT Architecture Used in the FF Lab

Siemens Mindsphere was used as a cloud-based data collection and analytics hub for manufacturing data coming from the Edge device. Data for the FF lab was hosted on an isolated Mindsphere "tenant" for the McNair center. This tenant includes data visualization and analytics tools. The data and apps available on this tenant are exposed to the internet, meaning that anyone with the proper credentials can access manufacturing data from the Mindsphere tenant from any location. Additionally, the data on Mindsphere is also exposed using APIs, allowing other applications such as Mendix to pull data from the tenant. Currently, the Mindsphere environment at McNair serves as a data collection pool, holding mostly raw unprocessed data and exposing some of that data on status-based dashboards. Future development for Mindsphere and data analytics includes identification of KPIs, energy consumption monitoring, custom app development, and production quality analysis.

4.2 DIGITAL TWIN IMPLEMENTATION FOR AUTOMATED FIBER PLACEMENT

4.2.1 Physical System Duplication

As discussed in chapter 3 of this thesis, the AFP lab and associated equipment was duplicated in the DT using CAD models provided by machine vendors and traditional

physical measurement techniques. When available, CAD models directly from equipment vendors were used, otherwise dimensions were taken from the physical equipment and parts were modeled using a CAD system. Based on the measurement techniques and layout drawings available of the lab space, it was assumed that the DT was accurate to within 10mm of the physical layout of the cell. This level of accuracy was deemed sufficient for many components in the cell, however greater detail was required for the AFP machine itself. It was assumed that the Lynx AFP machine model provided from IMT was accurate to within 1mm, which should be sufficient for all planned experiments. This assumption would be validated using 3D scanning techniques.

4.2.2 3D Scanning

In a similar manner to the application described above for the FF lab, 3D scanning was used to validate the DT layout of the AFP lab. 3D scans of the entire lab space were taken using the same Faro Focus 70M scanner, which produced scans with an accuracy of 7.7 mm over a 10m range. A total of 40 scans were taken of the lab from 40 scanner locations. These scans were registered using Faro's Scene software, and a mean point error of 1.7 mm was reported. The resultant point cloud was of sufficient quality to validate the location of all components inside the DT, as well as validate the dimensions of the Lynx machine model provided by IMT. Any required adjustments of the DT layout, as well as adjustments to machine models in the DT, were performed at this stage. These adjustments, their magnitudes, and the impact on the overall accuracy of the DT of the AFP lab is discussed in chapter 5 of this thesis. The resultant point cloud used in this implementation consisted of 203 million points. An image of this point cloud is shown in Figure 4.8 below.

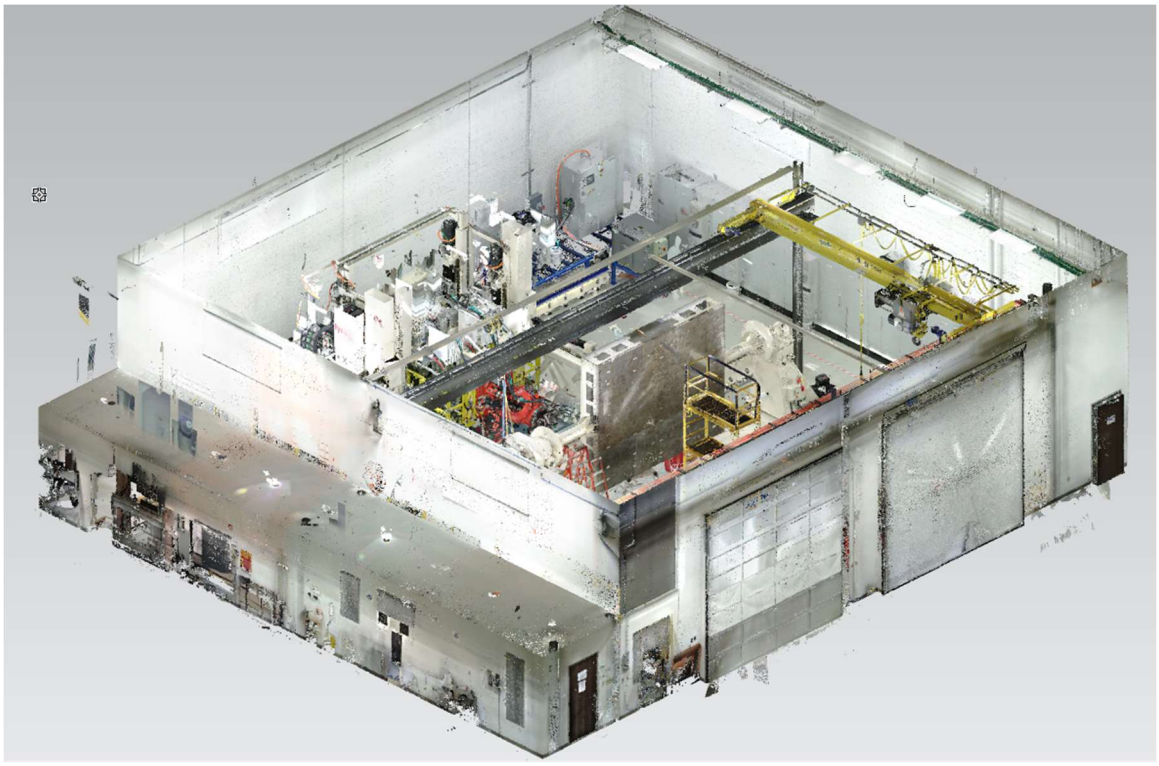

Figure 4.8. The 3D Scan of the AFP Lab

4.2.3 Off-Line Programming

Off-line programming techniques were used on two systems in the AFP lab: the ACSIS scanning system, and the AFP machine itself. The ACSIS system utilized a standard OLP workflow using Process Simulate, while the AFP machine required more customization and non-standard techniques to program layup paths. The techniques used for both systems are described below.

4.2.3.1 AFP Programming

To produce composites manufacturing programs for use on the AFP machine, feature definitions created in Catia V5 are created. Tool surface definitions, ply boundaries, starting points, and a directional rosette are created in Catia to define the plies and laminate. Further, the geometry of the tool and mandrel is used to accurately place the tool in the correct location relative to the machine rosette. These definitions are then exported from

the initial design environment into the neXt Composites (neXtC) and Computer Aided Process Planning (CAPP) tools described by Alex Brasington in [55] These tools work in conjunction with CGTech's Vericut Composite Programming (VCP) tool to create ply and course definitions.

The neXt environment can then be used to perform process planning, utilizing a combination of the CAPP software, VCP software, and the neXtC software. CAPP is used to import the original design data of the generated tool geometries through step files. A series of processes are then performed to split the surface at the ply boundaries, mesh the ply boundary surface, and find an initial starting point with a user defined layup strategy. CAPP then utilizes a back-and-forth connection with VCP where courses and defects are generated and analyzed. The VCP environment accepts an Excel export from CAPP that defines the parameters for generating the layup such as tow width, tow count, starting points and layup strategies. From this data, VCP generates the tows and courses, analyzing their geometry for overlaps, gaps, angle deviations, and steering. All of this is done with a batch process that creates files and directories that are imported back into the CAPP environment. CAPP then uses these files defining the defect occurrences to score the starting point based on user defined inputs prioritizing the different defect types. The user can then iterate this process to optimize the location of the starting point and layup strategy.

Using these optimized course definitions, neXtC then modifies the course centerlines for communication with Process Simulate. The courses are modified to include roll in and roll out beyond the ply boundaries, and layup directions are defined. Using the tow definitions from VCP, points are projected onto the course centerline that inform PS

where tows are either turned on or off using a tow mask. The geometry of the courses and the points is then exported to a STP file for importing into PS.

At this point in the process, the DT created for this system in PS acts as a Rosetta Stone for communication between the various software tools that have been employed previously. The output from neXtC is imported into PS using custom API tools that allow the courses and tow masks to be imported and visualized. An example of an imported course STP file and associated tow mask is shown in Figure 4.9.

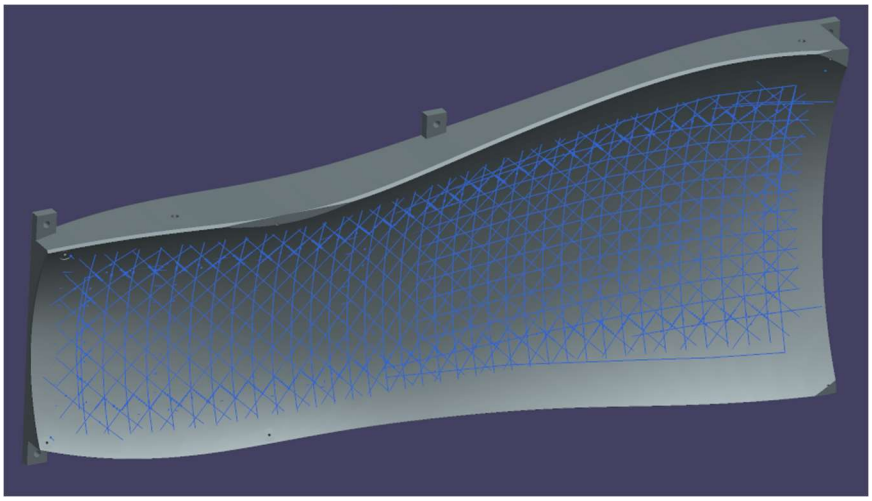

Figure 4.9. Image of the Imported Course STP File for AFP Programming

Projection tools inside PS are then used to create path locations and eventually machine motion simulation. The paths can then be edited inside PS to create off-part motion or adjust head angle at any path location. Tow adds and cuts are then added to the paths using a custom API tool that reads the tow mask from neXtC. This tow information is stored on the operations inside PS and will be used when the paths are exported for the machine. Machine motion simulations are then created from the finalized paths. Collision detection is performed between the machine head and the tool to ensure that the exported

machine paths will be collision free. An example of the collision analysis process is shown in Figure 4.10.

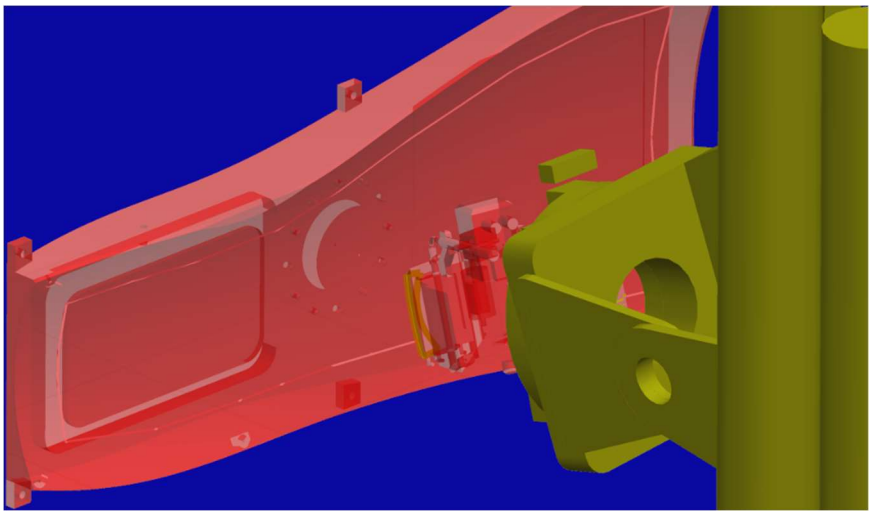

Figure 4.10. A Collision Analysis Performed Inside PS

Finalized and validated machine paths are then exported from PS as PF (Ply File) files using a custom tool developed in the PS API. These files contain the coordinates for machine motion for each ply, referenced to the origin of the tool surface. PF files contain both the cartesian position of the roller, the normal and travel direction vectors that define the orientation of the head, and P and Q angles that define the relative "tilt" of the head for collision avoidance. These points are referenced to the tool surface to allow for deviations between the machine model in PS and reality. The tool surface will be calibrated on the machine before these programs are run. These PF files are then used by the Gen 2 Simulation software produced by IMT to evaluate and finally post-process the paths originally created in VCP and CAPP. A description of the overall process flow is shown in Figure 4.11.

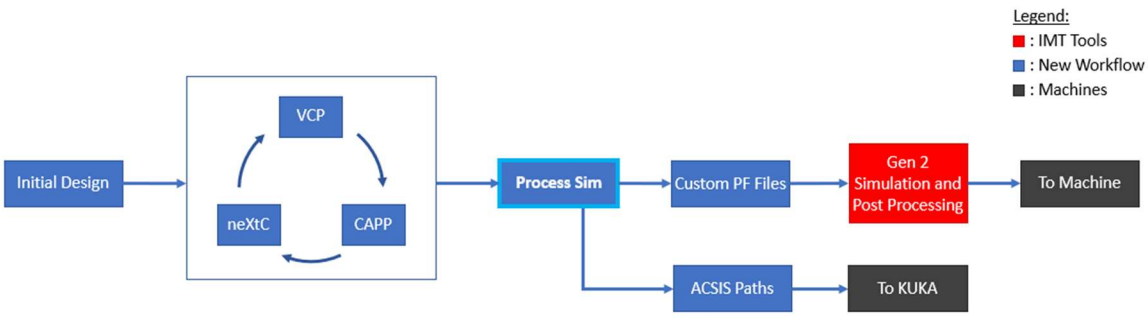

Figure 4.11. The Dataflow Between Applications for AFP and ACSIS Programming

The workflow described above was then validated through layup experiments with both a flat and doubly curved tool. For initial experiments utilizing the flat tool, the programmed laminate consisted of 4 plies with a stacking sequence of [0/90/45/-45], all with the same ply boundary. CAPP, VCP, and neXtC were utilized to communicate and provide course and tow definitions to PS. Simulation and machine code were then produced via PS for both the AFP machine and the ACSIS system. The Gen 2 simulation software was used to produce the final code to be sent to the AFP machine.

The same process was used to generate programs for a set of plies on a multi-curved tool surface. The curved tool surface was used to generate more complex paths for experimental verification of the DT techniques. A random set of ply angles were used for the curved tool laminate, as was required for other ongoing experiments. The final ply was set to be at 70°. The fiber angles of the remaining plies were randomly selected from a set of 0, ±45, and 90º angles. A total of 17 plies were generated for the curved tool, using the process described in Figure 4.11. Not all of these plies would be manufactured during the layup process, due to time constraints, however all were generated in the DT model.

Results of these experiments as they pertain to the accuracy and predictive value of the DT are discussed in chapter 5 of this thesis.

4.2.3.2 ACSIS Programming

In addition to AFP path programming capabilities, the DT model developed for the AFP space also allows for programming of the ACSIS (Automated Composites Structure Inspection System) robot. The same DT environment is used for AFP path planning and ACSIS programming, which allows for the ACSIS toolpaths to be generated using the same geometries (tool, paths, etc.) used for AFP. The previously defined kinematic mandrel is set to a 135° angle inside the simulation, and the same course definitions used for AFP programming are re-projected on to the tool surface to create paths for the ACSIS system to follow. A reduced number of paths are projected compared to the AFP courses, as the ACSIS scanning system can cover a wider area than one course at a time. Additional scanning passes are then added to incorporate the required 18mm stagger shift offset required for the scanning system. The resultant paths are then manipulated to correctly sequence the scanning process and add off-part motion. Custom xml-based path templates are used in PS to apply path properties such as active tool and base frame, speeds, accelerations, and to add non-motion commands to activate the profilometry scanners. After path templates are applied, the robotic paths are verified using the DT simulation environment to ensure collision free motion. The paths are then exported as a KRL (KUKA Robot Language) program to be run on the ACSIS robot. All the points in the exported program are referenced to a BASEFRAME on the tool surface that will be calibrated on the robot before the program runs to account for any differences between the virtual and physical systems. An image of ACSIS programming in PS is shown in Figure 4.12.

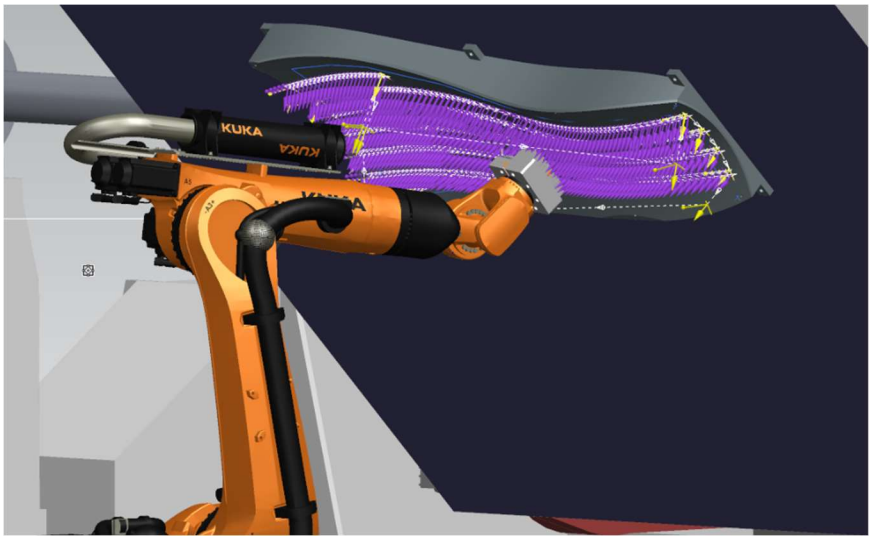

Figure 4.12. Programming of ACSIS in PS

4.2.4 Motion Capture

A motion capture system, similar to the system described in section 4.2.4, was used in the AFP lab to record the motions of the AFP machine during layup of composite parts. As described in chapter 3 of this thesis, an Optitrack system with multiple infrared cameras was used to track the locations of targets mounted on critical components in the cell. Cameras were mounted in the AFP lab on the two side of the AFP machine's Z axis travel, and above the AFP machine, such that the AFP head does not leave the vision range of the cameras. Tracking targets were mounted on the AFP head that physically performs the layout. These targets were grouped using the Motive software into a rigid body, whose location and orientation would be tracked during motion of the AFP machine. Motion capture recordings of the motion of the AFP machine were taken during dry runs of the plies described in section 4.3.3.1. Dry runs were used instead of live layup to avoid potential damage to and blinding of the motion capture system due to the intense light and IR emissions given off by the AFP heater during layup. The machine motion should be identical between dry runs and actual layup.

The recordings of AFP machine motion during dry runs would then be used to compare actual machine paths against planned machine paths and reconstructed machine paths using data collected from the AFP machine. This data could be used to predict defects, analyze potential motion problems, and measure head compaction distances. The analysis of this data is presented in chapter 5 of this thesis.

4.2.5 Layup and Scanning

Layup of the previously described plies was performed on both the flat and doubly curved tool surfaces to verify the accuracy and efficacy of the utilized DT techniques. An image of the layup process is shown in Figure 4.13. Plies planned and programmed with the DT model were manufactured using the Lynx AFP machine, and the motion of the machine during this layup was recorded using both motion capture equipment and by capturing machine axis coordinates from the motion controller. The machine axis data would later be used to reconstruct motion paths in the DT by utilizing the forward kinematics model of the AFP machine. The results of this layup and the recorded data is discussed in more detail in chapter 5 of this thesis.

Figure 4.13. The Layup Process for the Curved Tool

ACSIS was used to scan each ply after layup. The previously described programs were loaded to the ACSIS robot and ran to capture profilometry scans of each ply. These programs had been previously verified using the DT model and tested with a dry run on the actual system. The programs ran on the ACSIS system without any collisions or errors. The raw scan data would then be processed by a ML model developed by Sacco et al. and manually verified by the research team to classify defects in the ply [35].

CHAPTER 5

EXPERIMENTATION, RESULTS, AND CONCLUSIONS ON THE USE OF DIGITAL TWIN SYSTEMS FOR FUTURE FACTORIES AND AUTOMATED FIBER PLACEMENT

The previous chapters have described the modeling and creation of DT systems for both the Future Factories industrial technology test bed and the automated fiber placement lab at the USC McNair Aerospace research center. This chapter will discuss in detail the effectiveness of the two DT systems, as well as the results, both quantitative and qualitative, of the experiments described in chapter 4 of this thesis. Additionally, this chapter will discuss the implications of these results and how they relate to the effective of certain implemented DT techniques. In the author's opinion, these results show the efficacy of DTs for manufacturing systems across two specific applications and demonstrate viable paths for future DT implementations. Additionally, it is shown that the DT implementation developed for FF and AFP provides a solid foundation on which to build and test future manufacturing technologies and research topics. Finally, this chapter will discuss potential directions for future work in the DT for manufacturing research area.

5.1 DIGITAL TWIN EXPERIMENTS AND RESULTS FOR FUTURE FACTORIES

The implementation of the FF lab as described in chapter 4 provides a test bed to verify the functionality and usefulness of many DT for manufacturing systems. Two components of the DT were selected for verification: DT layout accuracy and DT path

programming and accuracy. These components were assessed with both qualitative and quantitative methodologies, and the results of these analyses are described below.

5.1.1 Digital Twin Layout Accuracy

The DT model of the FF lab was first evaluated based on the accuracy of the placement of physical components in the DT model as compared to their actual locations in the physical system. This assessment was performed using various techniques, including simple measurements with tape measures, motion capture based methodologies to find robot locations, and 3D scanning to provide a wholistic view of the accuracy of the entire cell. Each method had unique benefits and drawbacks, and the results of each analysis are discussed in detail below.

5.1.1.1 Cell Measurements

Physical measurements of the actual FF lab layout were taken using standard measurement techniques such as tape measures and meter sticks. Certain key dimensions were selected for measurement and comparison. These dimensions were selected for their relevance to collision avoidance, path accuracy, multi-robot collaboration, and tooling accuracy. In total, 34 dimensions were selected for comparison. These dimensions were separated into groups based on the type of objects they located and the relevance of the dimension to successful robot operation. The categories of dimensions chosen were as follows: fixture locating dimensions, conveyor locating dimensions, table locating dimensions, robot locating dimensions, and obstacle locating dimensions. Fixture locating dimensions describe the locations of assembly fixtures relative to other features in the cell. Conveyor locating dimensions describe the locations of the conveyors relative to the tables in the cell; these dimensions are less critical. Table locating dimensions describe the

relative locations of tables in the cell, referenced to each other and the walls of the cell. Robot locating dimensions describe the locations of robots relative to each other and relative to the tables on which they are mounted. Finally, obstacle locating dimensions describe the locations of key features in the cell that the robots need to avoid to prevent collisions. These categories of dimensions are shown in Figure 5.1 – 5.5.

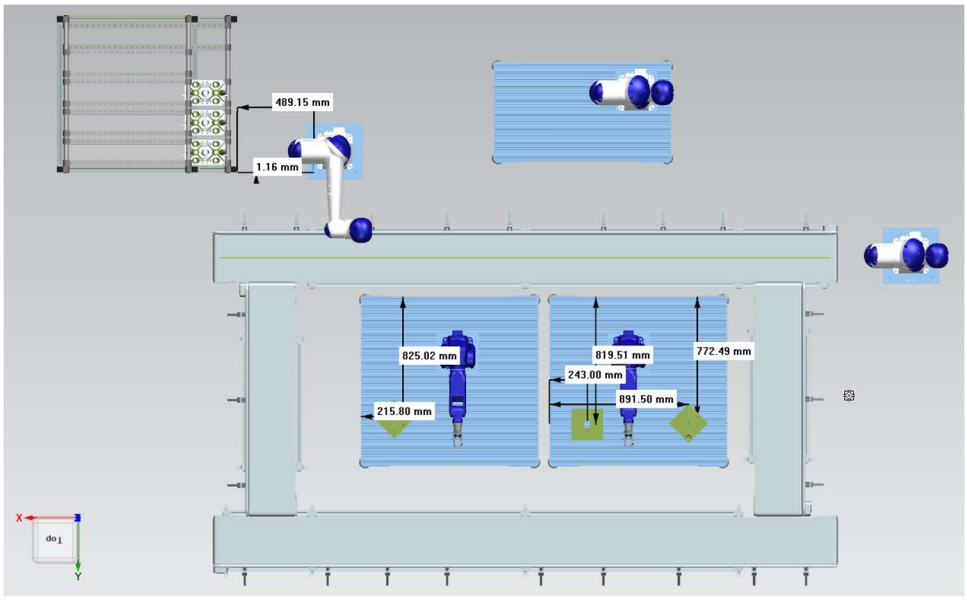

Figure 5.1. Fixture Locating Dimensions from the DT Model

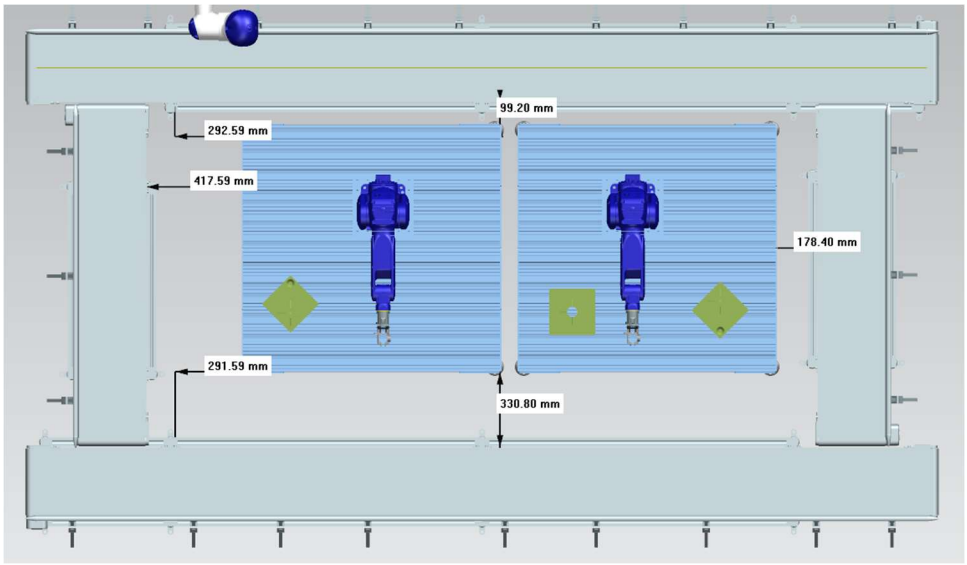

Figure 5.2. Conveyor Locating Dimensions from the DT Model

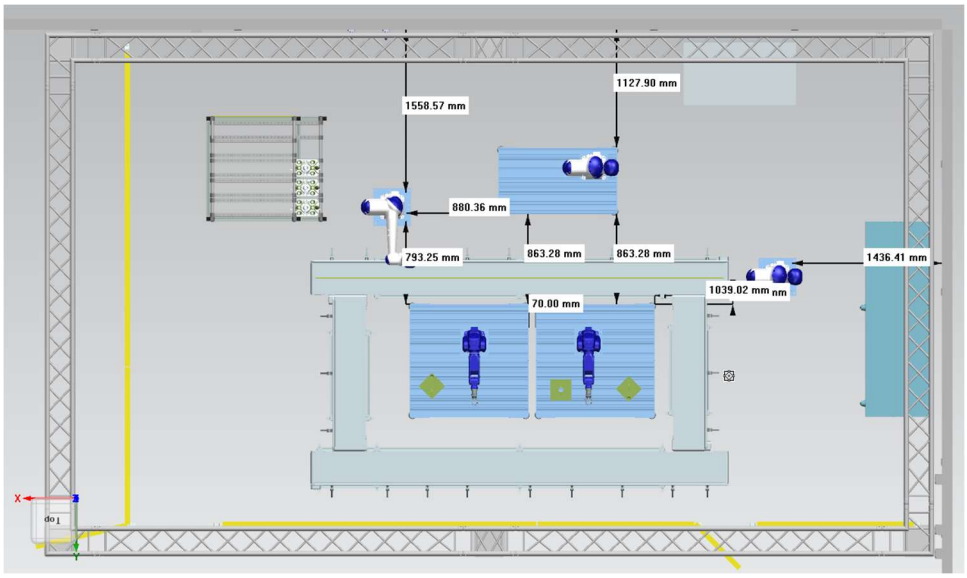

Figure 5.3. Table Locating Dimensions from the DT Model

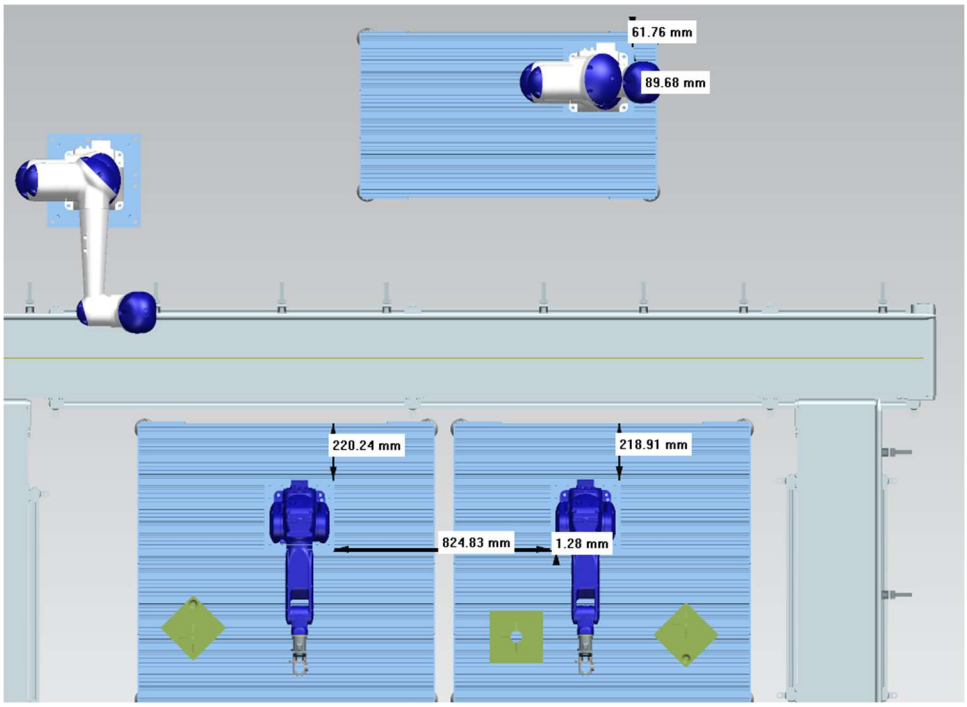

Figure 5.4. Robot Locating Dimensions from the DT Model

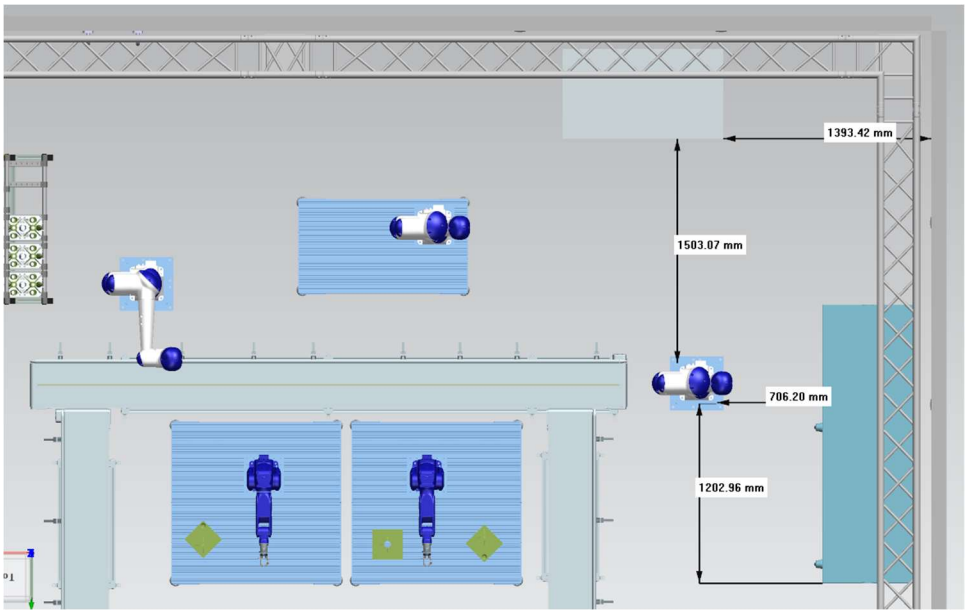

Figure 5.5. Obstacle Locating Dimensions from the DT Model

After measurement of each selected dimension in the physical system, the physical measurements were compared to the planned measurements from the DT model. The results of these comparisons are shown below in Table 5.1. Note that all hand measurements were taken with an accuracy of ±5mm unless otherwise noted. Some of these dimensions were identified as critical to path accuracy for the process identified in the cell. These dimensions pertain mostly to fixture and robot locations and are noted as critical to accuracy in Table 5.1.

Table 5.1. DT and Hand Measurements of Dimensions from the FF Cell

Dimension	DT Measurement (mm)	Hand Measurement (mm)	DT Error (mm)	Notes	Critical to Path Accuracy
Fixture Locating Dimensions					
R2TableFixDim1	825.02	825	0.02		x
R2TableFixDim2	215.8	214	1.80		x
R3TableFix1Dim1	243	243	0.00		x
R3TableFix1Dim2	819.51	817	2.51		x
R3TableFix2Dim1	772.49	770	2.49		x
R3TableFix2Dim2	891.5	888	3.50		x

MHSDim1	489.15	491	1.85		x
MHSDim2	1.16	3	1.84	Hard to measure	x
Conveyor Locating Dimensions					
Conv3Dim1	417.59	427	9.41		
Conv2Dim1	99.2	94	5.20		x
Conv1Dim1	178.4	269	90.60		
Conv4Dim1	330.8	340	9.20		
Conv2Dim2	292.59	320	27.41	±10mm	
Conv4Dim2	291.59	299	7.41		
Table Locating Dimensions					
R4TableDim1	1127.9	978	149.90		
R1TableDim1	1558.57	1183	375.57		
R1R4TableDim	880.36	372	508.36		
R4R2TableDim	863.28	638	225.28		
R4R3TableDim	863.28	589	274.28		
R1R2TableDim	793.25	813	19.75		
R2R3TableDim	70	27.5	42.50		
R3R5TableDim1	1039.02	975	64.02		
R3R5TableDim2	126.26	151	24.74	±10mm	
R5TableDim1	1436.41	1502	65.59		
Robot Locating Dimensions					
R4Dim1	89.68	1	88.68		
R4Dim2	61.76	5	56.76		
R2R3Dim1	824.83	797	27.83		x
R2R3Dim2	1.28	2	0.72		x
R2Dim1	220.24	231	10.76		x
R3Dim1	218.91	226	7.09		x
Obstacle Locating Dimensions					
R5CabinetDim1	706.2	882	175.80		
R5ServerDim1	1503.07	990	513.07		
R5CabinetDim2	1202.96	2042	839.04		
ServerWallDim	1393.42	1550	156.58		

5.1.1.2 3D Scanning Results

3D scanning techniques were used as described in Chapter 4 to provide another point of comparison between the DT and the physical system. As compared to the physical measurement technique described above, this method provides a more holistic analysis of

the layout of the location of objects in the DT. The same dimensions described above were used to analyze the differences between the 3D scan of the physical system and the DT. To take these measurements, the 3D scan was overlayed on top of the DT, aligned with a selected datum feature, such as the center of a robot. Measurements were then taken using the overlayed 3D scan, by selecting two relevant points from the point cloud and measuring the distance between them in software. As all measurements described above are relative distances between objects in the cell (not absolute measurements relative to a datum), the exact location of the 3D scan in the DT model does not affect these results. The 3D point cloud collected from the 3D scan was trimmed to only contain data from the FF lab and is shown in Figure 5.6. The resultant measurements from the 3D scan are presented in Table 5.2.

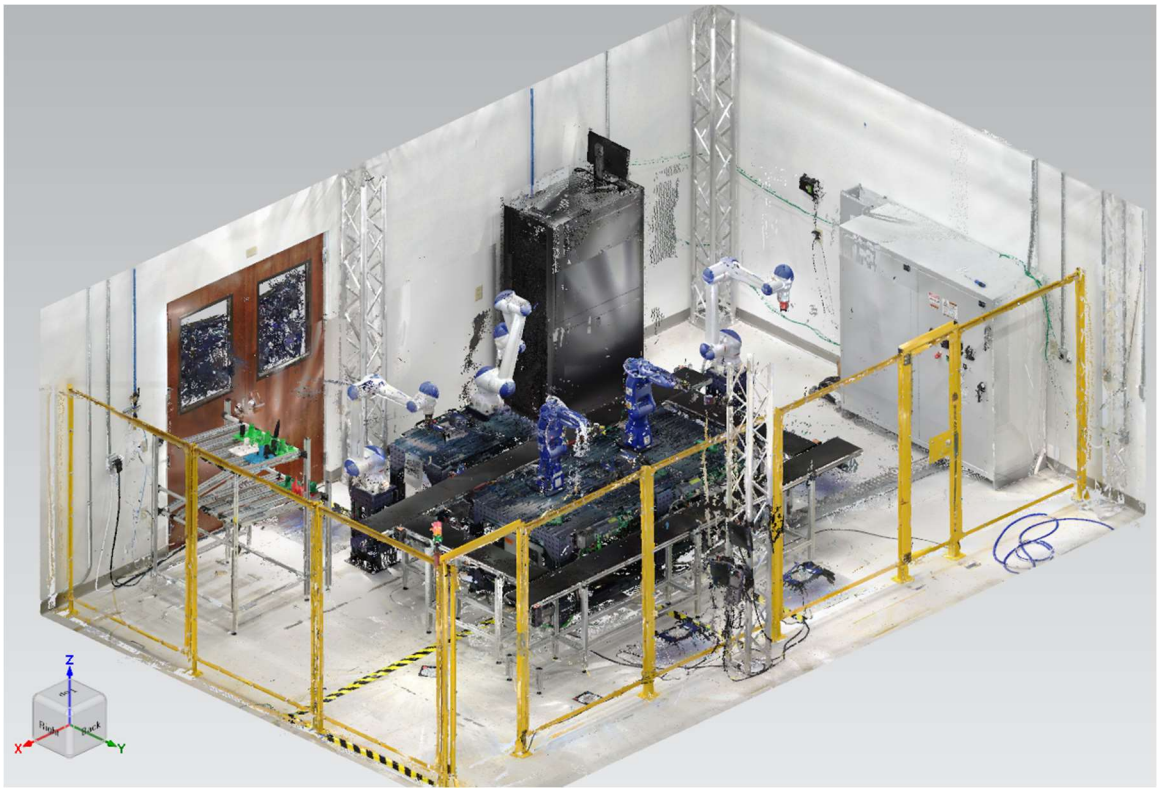

Figure 5.6. The Resultant Trimmed 3D Scan of the FF Lab

The 3D scan point cloud contained some areas where scan quality was reduced from nominal. These areas were primarily caused by reflections off reflective surfaces in the lab, such as glass windows, by occlusion of certain areas due to shadow casting, and by scanner placement choices resulting in some areas having better point coverage than others. Areas where reflection created false images in the point cloud were manually removed from the point cloud. An example of a false image resulting from reflection is shown in Figure 5.7. An example of a case where scanner placement and shadow casting resulted in a lower quality scan is shown in Figure 5.8, where R02 is shown to be substantially clearer from one side of the scan than the other. If these defects substantially affected the measurement of a dimension from the 3D scan, that measurement was noted as being potentially inaccurate in Table 5.2.

Figure 5.7. False Points Resulting from Reflections in the Glass Windows of the Doors in the FF Lab

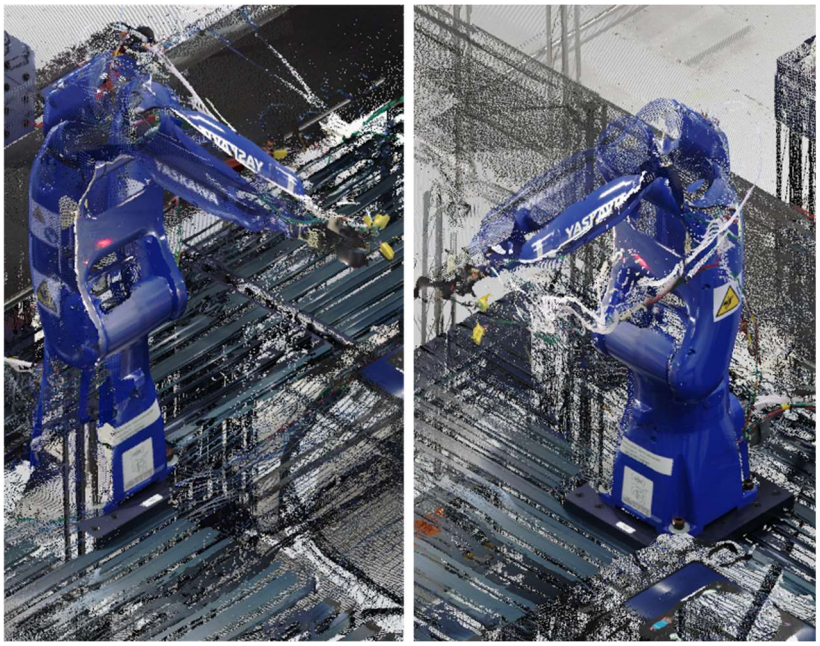

Figure 5.8. R02 As Viewed in the FF Point Cloud Data from the Left and Right

Table 5.2. DT and 3D Scan Measurements of Dimensions from the FF Cell

Dimension	DT Measurement (mm)	3D Scan Measurement (mm)	DT Error (mm)	Notes	Critical to Path Accuracy
Fixture Locating Dimensions					
R2TableFixDim1	825.02	824	1.02		x
R2TableFixDim2	215.8	212	3.80		x
R3TableFix1Dim1	243	243	0.00	Close to scanner position, low res	x
R3TableFix1Dim2	819.51	820	0.49	Close to scanner position, low res	x
R3TableFix2Dim1	772.49	767	5.49		x
R3TableFix2Dim2	891.5	891	0.50		x
MHSDim1	489.15	490	0.85	Very clean points	x
MHSDim2	1.16	1	0.16	Very clean points	x
Conveyor Locating Dimensions					
Conv3Dim1	417.59	437	19.41		
Conv2Dim1	99.2	101	1.80		x
Conv1Dim1	178.4	181	2.60	Probably out of date, conveyor has moved since scan	
Conv4Dim1	330.8	342	11.20	Very clean	
Conv2Dim2	292.59	325	32.41		

Conv4Dim2	291.59	310	18.41		
Table Locating Dimensions					
R4TableDim1	1127.9	967	160.90		
R1TableDim1	1558.57	1182	376.57		
R1R4TableDim	880.36	370	510.36		
R4R2TableDim	863.28	652	211.28	Low point density	
R4R3TableDim	863.28	610	253.28	Low point density	
R1R2TableDim	793.25	824	30.75		
R2R3TableDim	70	31	39.00	Hand measurement is probably more accurate at this scale	
R3R5TableDim1	1039.02	971	68.02	Very clean	
R3R5TableDim2	126.26	161	34.74	Very clean	
R5TableDim1	1436.41	1505	68.59	Very clean	
Robot Locating Dimensions					
R4Dim1	89.68	0	89.68		
R4Dim2	61.76	7	54.76		
R2R3Dim1	824.83	801	23.83		x
R2R3Dim2	1.28	7	5.72	Odd, this measurement looks clean, but 3D scan is not square, so that could be it	x
R2Dim1	220.24	226	5.76		x
R3Dim1	218.91	238	19.09		x
Obstacle Locating Dimensions					
R5CabinetDim1	706.2	876	169.80	very clean	
R5ServerDim1	1503.07	987	516.07	very clean	
R5CabinetDim2	1202.96	2044	841.04	very clean	
ServerWallDim	1393.42	1551	157.58		

5.1.1.3. Discussion of Cell Measurement and 3D Scanning Results

In general, the layout of components inside a DT model is critical to the success of that model as both a programming tool and an analysis tool. Both hand measurement and 3D scanning techniques provide a means for verification of the accuracy of a layout inside the DT model. A comparison of the results from Table 5.1 and Table 5.2 show that both the hand measurements and 3D scan measurements agree closely on most dimensions,

often to within the margins of error of the measurement techniques. In this experiment, hand measurement was assumed to have a higher margin of error (±5mm) than the 3D scan data (±2mm), however these error ranges vary dramatically depending on the dimension being measured. For instance, short dimensions between flat surfaces separated by large distances (such as the R2R3Dim2 dimension in the images and tables above) are difficult to measure by hand but trivial to measure using 3D scanning techniques. 3D scanning techniques, however, suffer from issues stemming from scanner placement and false image generation due to the optical nature of the device. 3D scanning is also a time consuming and potentially expensive process that is not easily repeated without disruption to the manufacturing process. As such, one must be careful to ensure they are working with up-to-date 3D scan data or be aware of where the 3D scan may differ from reality, as evidenced by measurements of Conv1Dim1 in Table 5.1 and Table 5.2. Care must be taken to avoid these potential errors when utilizing a 3D scanning system.

These drawbacks of a 3D scanning system do not substantially reduce the usefulness of a 3D scanned point cloud in validation and use of a DT system. In general, we found that the 3D scanning system utilized in this experiment produced an accurate 3D point cloud representation of the cell that was usable to verify and update the DT layout. Additionally, we propose that such a system is often easier to use than manual measurement of the same cell and allows for a more holistic view of the entire manufacturing workspace. This allows an engineer using the DT system to measure dimensions using the 3D scan that may have been omitted from physical measurements of the cell. This holistic view of the system allows users to build their DT model in the context of the point cloud data, and even incorporate that data into the model itself. This allows for easier evaluation of layout

alternatives, what-if scenarios, and placement options in the context of real-world data from the physical space (such as the locations of walls and columns).

In evaluating the accuracy of the layout of the DT model itself, both hand measurement and 3D scanning techniques provide data indicating that in some dimensions the DT model matches reality very closely, while in others it varies wildly. On dimensions labeled as "critical to path accuracy" as described above, we see much closer agreement between the DT model and both physical measurement techniques. These dimensions and their hand measurement and 3D scanning results are shown in Table 5.3 below.

Table 5.3. Critical Dimensions and Deltas for Both Hand Measurement and 3D Scan Measurement of the FF Cell.

Dimension	DT Measurement (mm)	Hand Measurement Delta (mm)	3D Scan Measurement Delta (mm)
R2TableFixDim1	825.02	0.02	1.02
R2TableFixDim2	215.8	1.80	3.80
R3TableFix1Dim1	243	0.00	0.00
R3TableFix1Dim2	819.51	2.51	0.49
R3TableFix2Dim1	772.49	2.49	5.49
R3TableFix2Dim2	891.5	3.50	0.50
MHSDim1	489.15	1.85	0.85
MHSDim2	1.16	1.84	0.16
Conv2Dim1	99.2	5.20	1.80
R2R3Dim1	824.83	27.83	23.83
R2R3Dim2	1.28	0.72	5.72
R2Dim1	220.24	10.76	5.76
R3Dim1	218.91	7.09	19.09

Most dimensions marked "critical to path accuracy" have a small delta between the DT model measurement and the actual measurement, as measured by either hand or 3D scan. Most dimensions are within ~5mm of their actual value, based on either measurement technique. Two dimensions in Table 5.3 are outliers to this rule. One, R2R3Dim1, appears to have shifted from its design value by ~25mm when built in the physical system. This error was noticed and corrected for during robot calibration as described in Chapter 4. The

other outlier dimension, R3Dim1 appears to vary depending on the measurement technique, with hand measurement agreeing more closely with DT model. The 3D scan measurement of this dimension was annotated as having "low point density" and as such this could explain this variation. Additionally, the difference between the two measurement techniques could be due to the fact that the 3D scan and the cell measurements were not taken as the exact same time, and a such some elements of the cell may have shifted.

Dimensions not marked as "critical" often vary substantially more from their expected DT measurements. For instance, all of the obstacle locating dimensions in Tables 5.1 and 5.2 vary from their expected values, using either 3D scanning on hand measurement, by over 150mm, and in one case, over 800mm. These errors are shockingly large and would not be immediately apparent from simply viewing the DT model and the physical cell in person. While these dimensions (and others with large variations) are not immediately critical to path accuracy, as they do not locate fixtures or robots directly, these large errors do pose a threat for robot collision avoidance. As such, after these errors were discovered, the DT model was updated using the 3D scan data to position these components more accurately in the cell, to within ~5mm of their actual location. These large, unexpected, errors provide more evidence for the use of a 3D scanning system to provide an accurate assessment of a DT model of a manufacturing system.

5.1.1.4. Motion Capture Results

To provide a final comparison point for layout accuracy of certain components in the DT, motion capture was used to derive physical locations for the two GP8 robots on the tables in the middle of the cell. The relative location of each of the two GP8s was calculated using the motion capture system described in Chapter 4. Motion capture targets

were attached to specific locations on each of the end effectors of the robots, and these targets were used to construct rigid bodies for each end effector in the Motive software used for motion capture. Tracking the end effector of each robot while the robot followed a specific path allowed the researchers to calculate the location of the base of each robot relative to the motion capture datum. The path selected for this calibration was a simple semi-circular sweep around the base of each robot, turning only the first joint, S, during the path. These semi-circular paths were programmed in the DT model using the OLP tools presented in Chapter 4 and are represented in Figure 5.9.

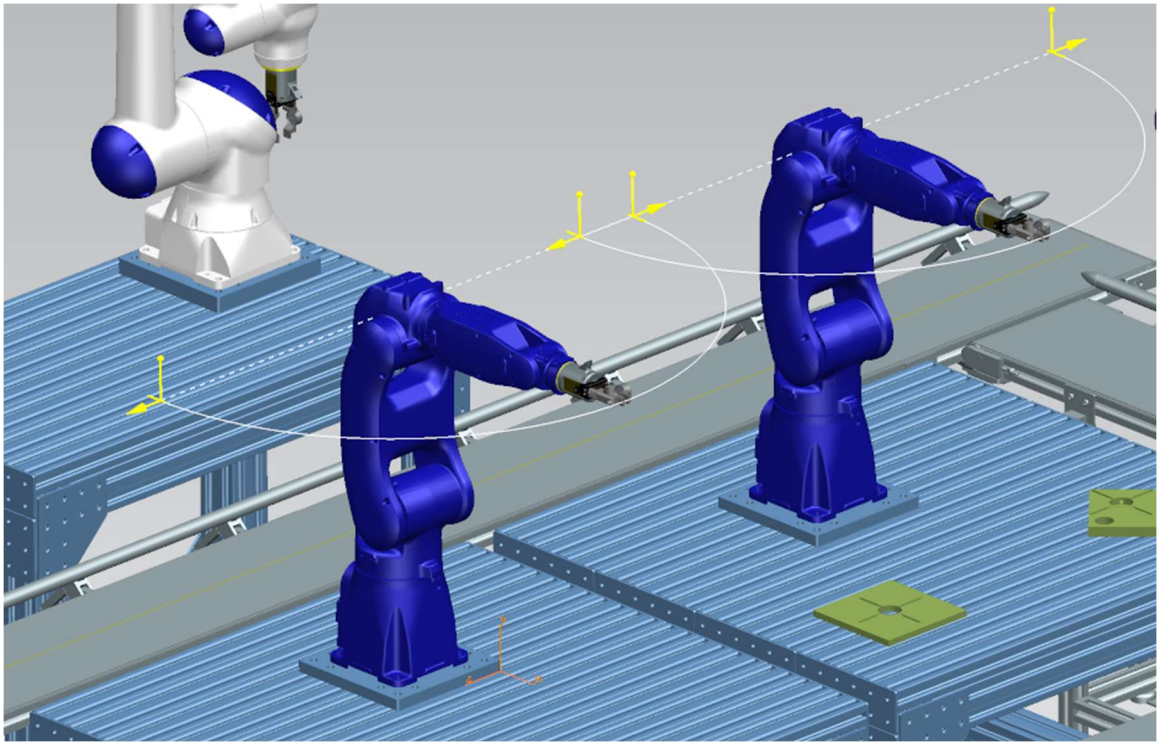

Figure 5.9. The Semi-Circular Paths Used for Motion Capture Calibration of R02 and R03

After programming, the robot motion during these paths was recorded using the motion capture system described in Chapter 4. These paths were recorded relative to the global motion capture datum, visualized as the orange frame in Figure 5.9 above. The

center of each path was found by fitting a circle to each path in three dimensions using a python script. Fitting the circle in three dimensions allowed the researchers to determine not only the circle center in 3D, but also the orientation of that circle, allowing for analysis of the relative rotations of each robot around the X and Z axes of the motion capture datum. The motion capture and circle fit results for R02 and R03 are presented in Figure 5.10 and Figure 5.11 respectively, where the blue points represent the imported motion capture paths, and the black curves represent the circle of best fit for each robot.

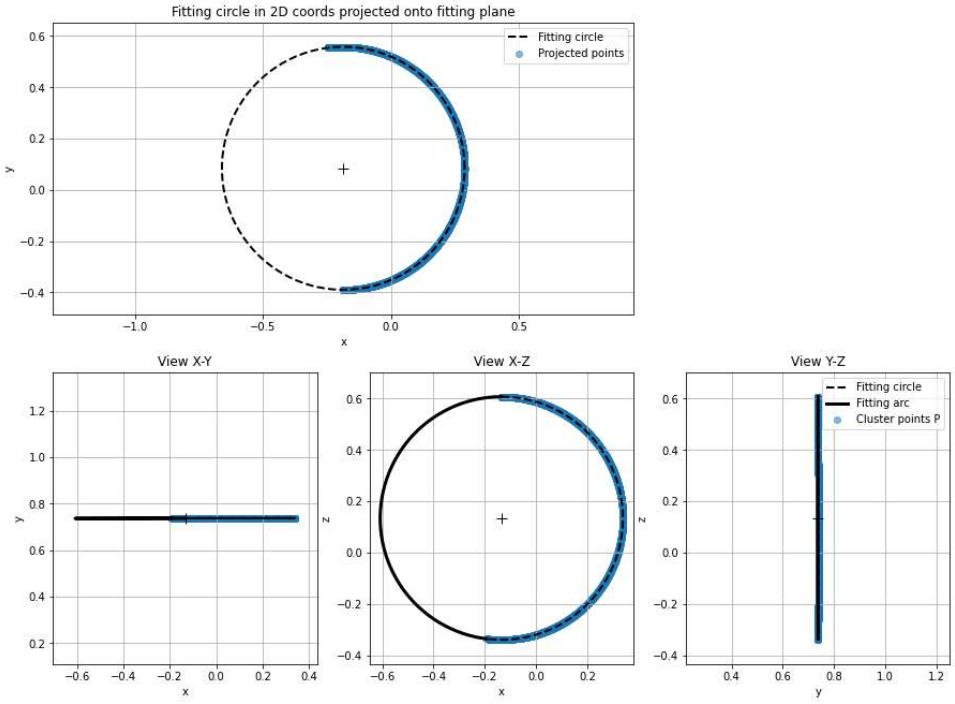

Figure 5.10. R02 Circular Path Curve Fitting Results

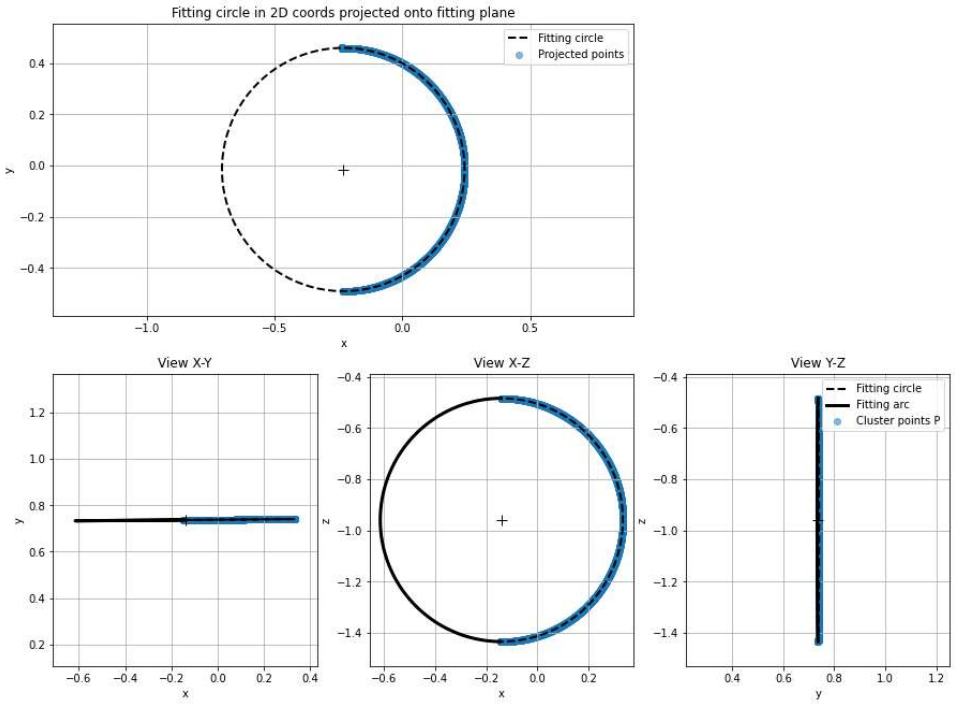

Figure 5.11. R03 Circular Path Curve Fitting Results

The numerical results from the circle fitting operations shown in Figure 5.10 and Figure 5.11 are shown below in Table 5.4. N_x, N_y, and N_z represent the normal vector of the plane of best fit for the circle, and thus the orientation of the circle in 3D space. C_x, C_y, and C_z are the coordinates of the center of the circle of best fit for each robot, relative to the motion capture datum previously described.

Table 5.4. Combined Motion Capture Data for R02 and R03 Circle Fitting

Robot	N_x	N_y	N_z	C_x (m)	C_y (m)	C_z (m)	R^2
R02	-0.00179	0.999998	0.000385	-0.1346	0.7375	0.1337	0.99999977
R03	0.007886	-0.99997	-0.00031	-0.1406	0.7365	-0.9391	0.99999992

The data shown in Table 5.4 above was then used to reconstruct the circle center points and orientations inside the Process Simulate DT model. After creation of these locations inside the DT, all coordinates were transformed to align with the global origin of the DT. These circle centers were then used, along with the CAD models of the robots in

the cell, to locate the base of each robot according to the circle fit data. These results are presented in Table 5.5 below.

Table 5.5. Coordinate Data of Resultant Locations Relative to Cell Origin

Location	X (mm)	Y (mm)	Z (mm)	Rx (deg)	Ry (deg)	Rz (deg)
R02 Circle Center	-7682.4	-257.2	1641.1	0.11	0.22	0.08
R03 Circle Center	-8755.2	-263.2	1640.1	0.45	0.17	0.81
R02 Base	-7685.2	-255.9	926.1	0.11	0.22	0.08
R03 Base	-8757.5	-257.5	925.1	0.45	0.17	0.81

5.1.1.5 Discussion of Motion Capture Results

The use of motion capture along with the circle fitting methodology described above provides an accurate way for locating the base of a robot arm in the context of the entire lab. As described in Chapter 4, the motion capture system records paths with a noise level of approximately 0.2mm on all linear axes. This low noise level, along with the very good fitting results shown in Table 5.4 results in a very accurate circle center location. Additionally, any systematic error resulting from the orientation of the tracked rigid body would be averaged out, as during the semi-circular path the end effector sweeps through a full 180° motion. These factors result in a circle center location that, in the worst case, is accurate to the noise level of the motion capture system, ±0.2mm on all axes.

Calculation of the accuracy of the rotational measurements is slightly more complicated but can be derived using the same values for positional accuracy of the motion capture system described above. In the worst case, the circle is systematically high by 0.2mm on one side of the sweep, and low by 0.2mm on the other side. The circle radius was 473.7mm. The angular deviation on R_x and R_y can then be calculated with the simple trigonometric relation below:

$$\theta = tan^{-1}\left(\frac{h}{R}\right) = tan^{-1}\left(\frac{0.2mm}{473.7mm}\right) = 0.024°$$

The accuracy of the rotation of the circle around the normal axis of the fitting plane, R_z, can be calculated using the exact same relation, as this rotation was simply based on the starting point of the sweep. A similar trigonometric relation can be found for this rotation, resulting in Rx, Ry, and Rz all having an accuracy of ±0.024°. Application of these rotational and positional variances to the location of the robot model in the DT results in a final positional accuracy of the robot placement of ±0.5mm on all axes and a final rotation accuracy of ±0.024° on all axes.

To validate these motion capture results, the DT model was updated with the final mounting locations of the two robots, and the two dimensions describing the relative locations of the two robots (R2R3Dim1 and R2R3Dim2) were updated and recorded. A comparison of these two dimensions across all three measurement methodologies is provided in Table 5.6 below.

Table 5.6. A Comparison of Measurement Techniques for Dimensions Locating R2 and R3

Dimension	Motion Capture Measurement (mm)	3D Scan Measurement (mm)	Hand Measurement (mm)
R2R3Dim1	802.7	801	797
R2R3Dim2	1.7	7	2

As shown in Table 5.6, the motion capture results agree with hand measurements for both dimensions to within the margin of error of the hand measurements. Additionally, the motion capture results agree closely with the 3D scan results for R2R3Dim1, however R2R3Dim2 shows a substantial difference between motion capture and 3D scanning. It was noted in Table 5.2 that this dimension as measured with 3D scanning may be uncertain due to differences in rotation between the DT and the physical system, as the point selected along the edge of the robot base for measurement has a substantial impact on the result of

that measurement. In this case, we trust that the motion capture and hand measurements are more accurate to the physical system. In total, these results show that motion capture techniques along with careful path selection and curve fitting can be used to record accurate (±0.5mm) robot locations from a physical system.

5.1.2 Off-Line Programming and Path Accuracy

Path programming accuracy and ease of use was also assessed using the DT model of the FF lab. The accuracy of paths programmed using the DT model was assessed using both qualitative and quantitative means. Qualitative analyses were performed by assessing ease of programming paths in the FF lab using the DT model, as well as assessing how often a path fails due to differences between the DT and the physical system. Quantitative analysis was performed using the motion capture system described in Chapter 4 of this thesis.

5.1.2.1 Qualitative Analysis

The qualitative analysis of the DT model in its usefulness in path programming and accuracy was mostly assessed through use of the DT and testimony from those who use the DT system. In general, the use of the DT for path programming resulted in accurate and collision free paths. As of the writing of this thesis, no path programmed using the DT has resulted in collision between robots or robot end effectors and any other component of the cell. Path accuracy, however, was not always perfect, and the layout of components inside the DT had to be adjusted multiple times to correctly mirror the locations of critical components such as assembly fixtures and robot mounts.

Calibration of the cell was performed as described in chapters 3 and 4 of this thesis using custom 3D printed calibration tools and the on board UFRAME calibration software

available on the Yaskawa robots. This calibration resulted in generally accurate path locations (usually within ~1mm of physical target), however some components that were not calibrated using this technique had to be adjusted in the DT to match their physical location. This adjustment was performed by running paths programmed using the DT and then measuring the approximate offset between the robot location at a given target and the desired location, and then moving the component inside the DT accordingly. Once these adjustments were performed, robot path programming was much faster and led to accurate paths most of the time. Most path programming inaccuracies after calibration were the result of components of the cell being inadvertently moved in the physical system.

5.1.2.2. Motion Capture Results

Motion capture techniques were used to provide a quantitative analysis of robot path accuracy in addition to the qualitative analysis provided above. Motion capture was used to analyze the positional accuracy of robot workspaces in the FF lab. Until this point, these workspaces were assumed to be perfectly accurate; a commanded robot position from the DT should exactly match the position of the real robot, relative to the base of said robot. Additionally, it was assumed that the robot's workspace would be perfectly cartesian, that jogging the robot along any of its primary axes would results in perfectly straight motion, and that these three axes are all perfectly perpendicular. Further experimentation with motion capture using the HC10 robots led to preliminary results indicating that these assumptions might not hold for the robots in our cell. These tests were designed to verify these assumptions.

The Yaskawa HC10 robot in the back in the FF lab, R05, was selected for these experiments. R05 was selected as the HC10 robots had previously shown some positional

inaccuracy during testing, and it was suspected it would produce the clearest results when analyzed with motion capture. R05 was instrumented for motion capture similarly to R02 and R03, by attaching motion capture targets to specific locations on the end effector, and mounting motion capture cameras on the truss in the FF cell. A rigid body was then constructed for the end effector using the Motive motion capture software.

Two tests were developed to evaluate the linearity and accuracy of R05's coordinate space. In the first test, a linear robot path for R05 was developed using the DT model and deployed to R05. This path moves the robot constantly along its X axis and stops at multiple locations on this line. Each move along the path was a PTP movement, and all robot locations were referenced to the robot's base coordinate system (without using any custom UFRAMEs). PTP movements were used to allow the robot to move through singularities and other difficult locations along the path, while still allowing us to collect location data at the individual points. A pause was added at each location such that the robot would stop at each point so a clean capture of the point could be taken using the motion capture system. The resultant linear path in the DT model is shown in Figure 5.12.

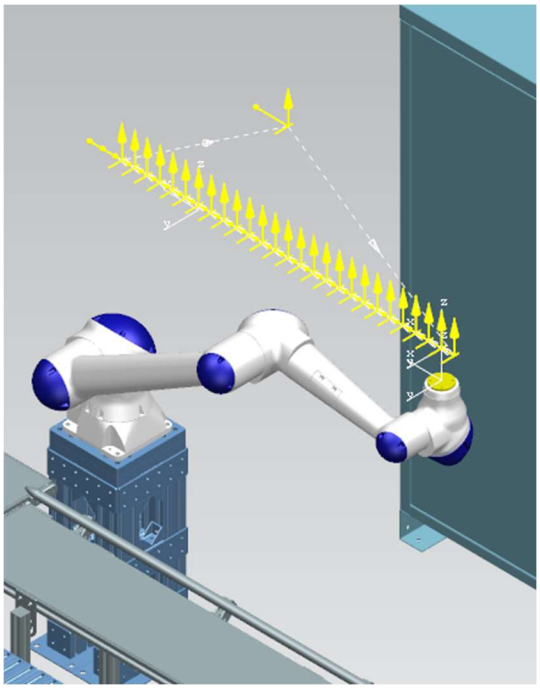

Figure 5.12. One of the Linear Test Paths Used on R05

This linear path was then post processed using the OLP tools in the DT, and then loaded to R05. The resultant motion was recorded using the motion capture system in the FF lab, and the coordinate data from the system was processed and visualized using a Python script. The resultant motion is visualized below in Figure 5.13 and Figure 5.14.

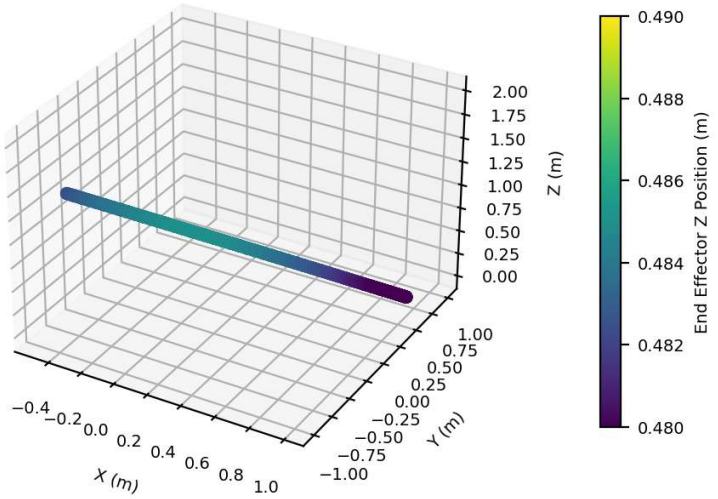

Figure 5.13. Raw Motion Capture Results from the First Linear Test on R05

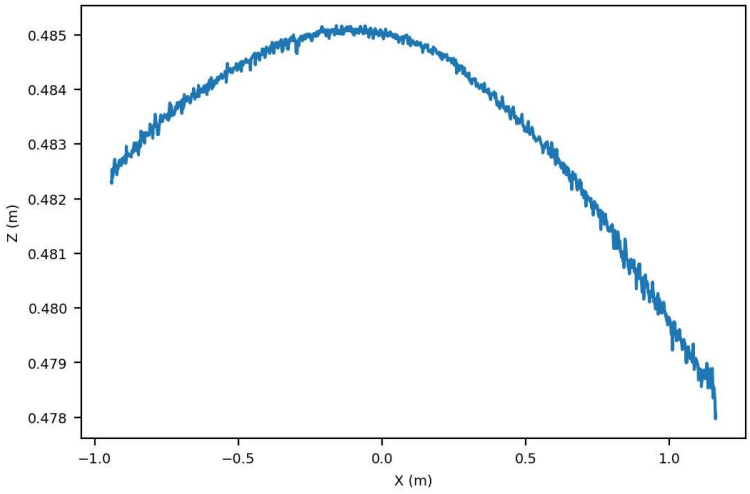

Figure 5.14. The X-Z Projection of R05's Robot Motion During the First Linear Test

This data was then discretized using the pauses added to the motion path. This discretization was performed to allow direct comparison back to the original planned motion path, and eventual correction of robot motion to recreate a linear path. Figure 5.15 and Figure 5.16 below show the discretized data from this test.

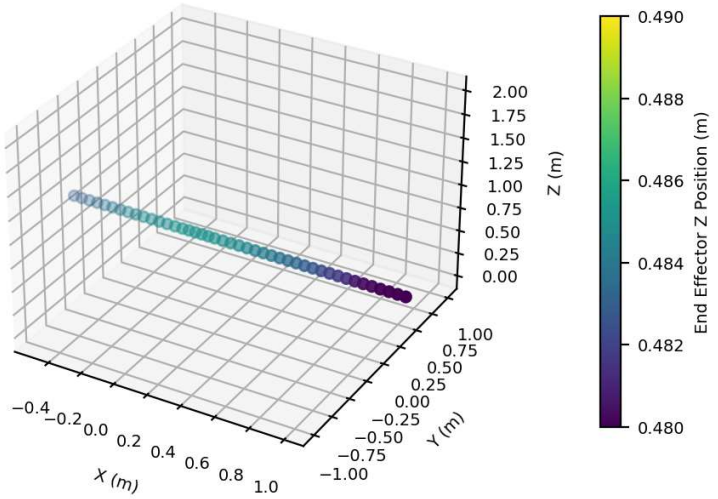

Figure 5.15. Discretized Motion Capture Results from the First Linear Test on R05

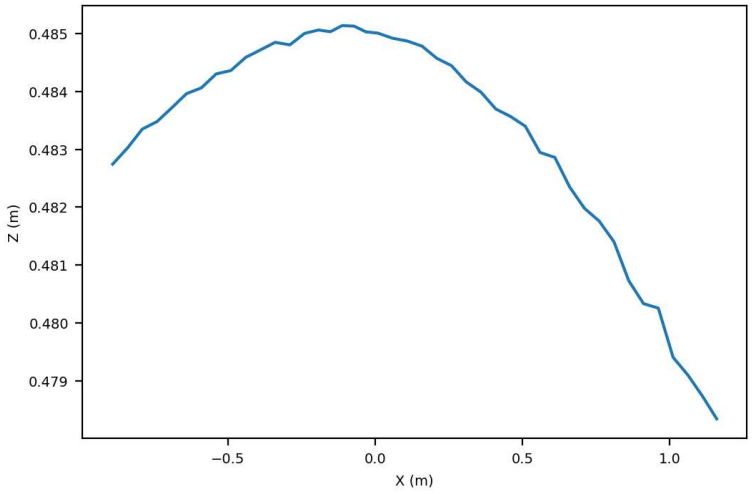

Figure 5.16 X-Z Projection of the Discretized Results from the First Linear Test on R05

Both the raw and discretized data above show clearly non-linear motion along a path that was programmed as perfectly linear. Additionally, these curves show a local high spot at X = 0m, at roughly the base of the robot. The discretized data was also processed to locate the highest and lowest Z values across all stopping locations in the path. The

highest recorded Z value was 0.4851m, and the lowest recorded value was 0.4783m. These values give a total Z deviation of 6.8mm across the entire linear robot path.

This linear test was then repeated using 6 different end effector orientations. These orientations are shown in the DT in Figure 5.17.

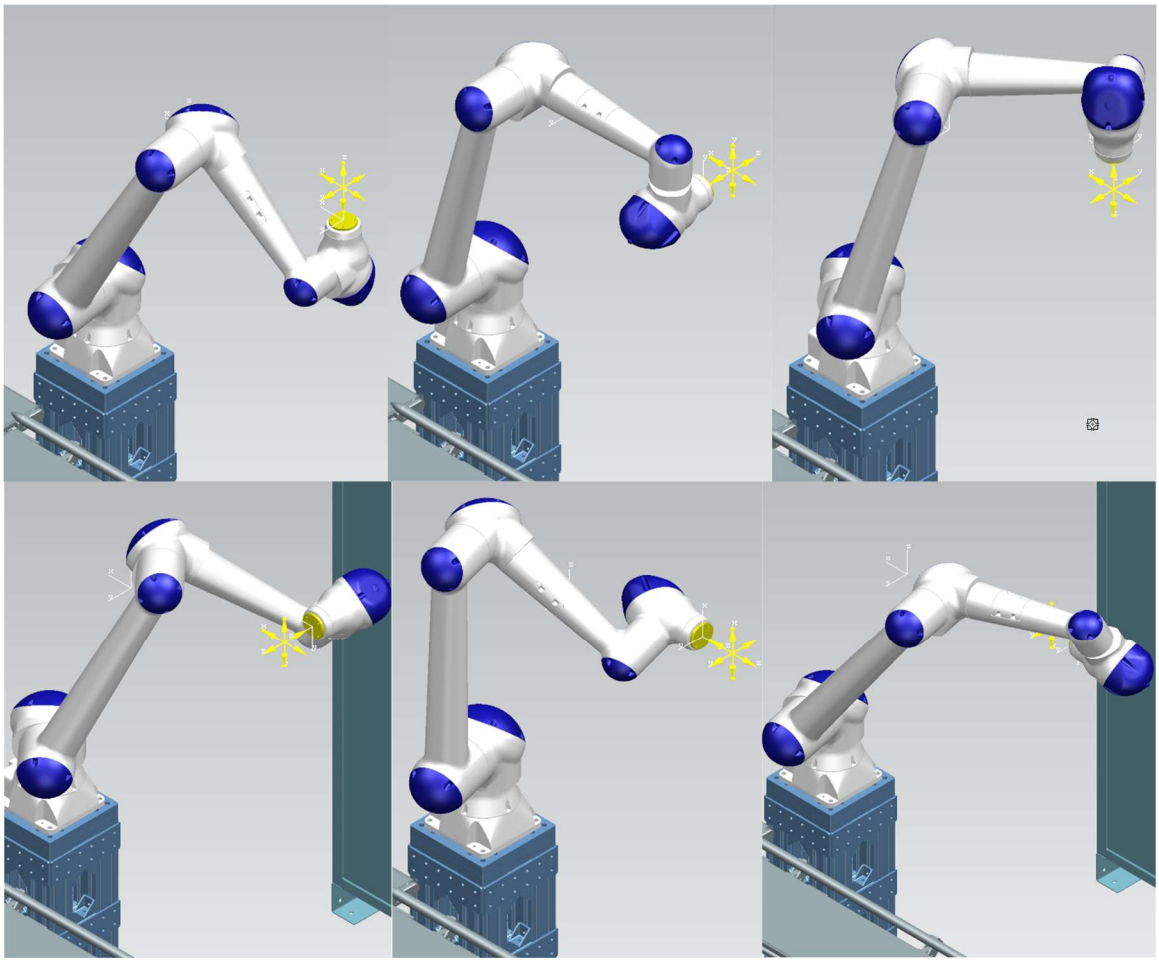

Figure 5.17. The 6 Different Orientations Tested in Linear Motion Capture Paths. From Top Left to Bottom Right: Up, Right, Down, Left, Forward, Backward.

The discretized results from these 6 tests are visualized in Figure 5.18 - Figure 5.23 below. Note that some tests have gaps in the middle of the path, as some locations are not reachable in the target orientation.

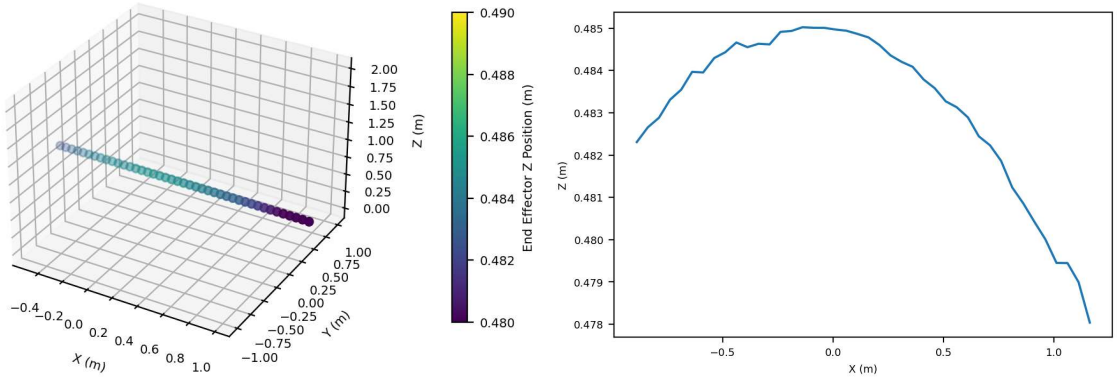

Figure 5.18. Up Orientation Linear Test Discretized Results

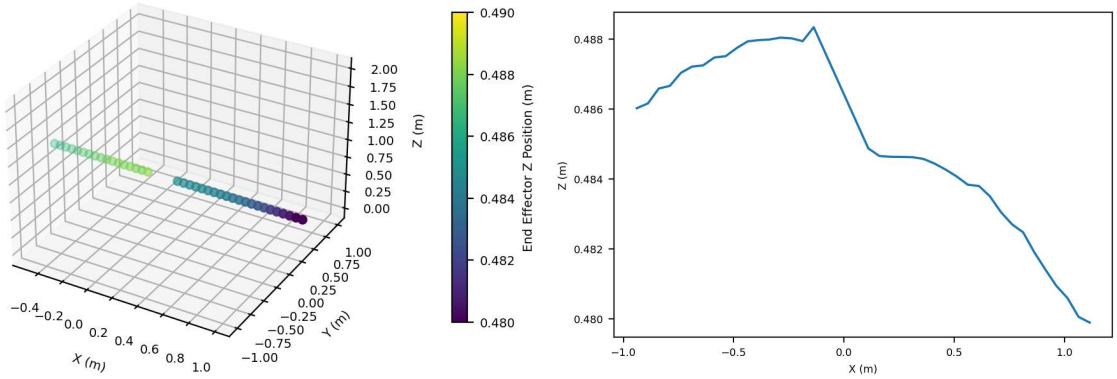

Figure 5.19. Right Orientation Linear Test Discretized Results

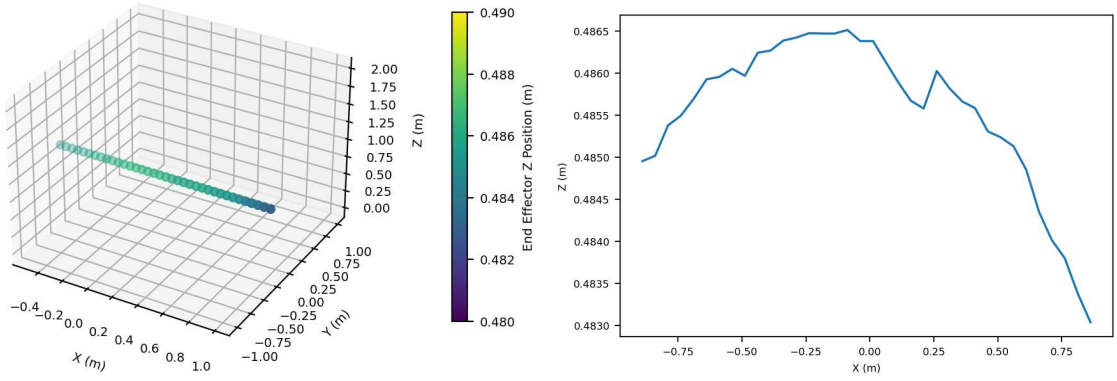

Figure 5.20. Down Orientation Linear Test Discretized Results

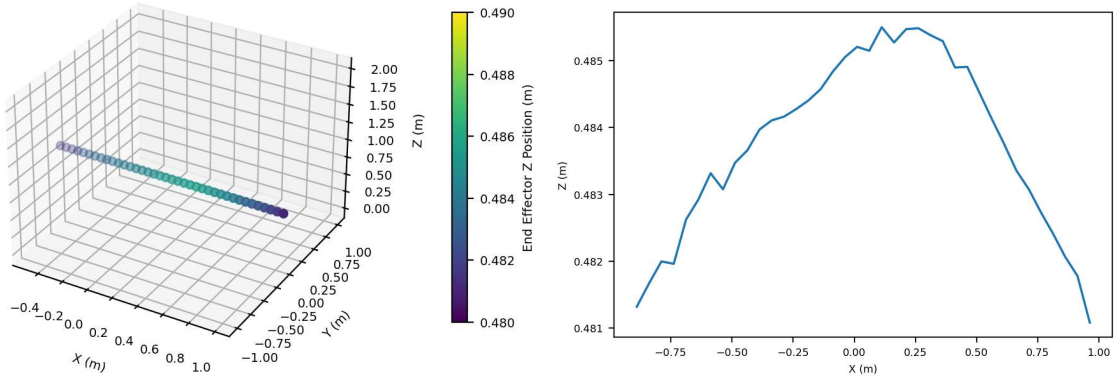

Figure 5.21. Left Orientation Linear Test Discretized Results

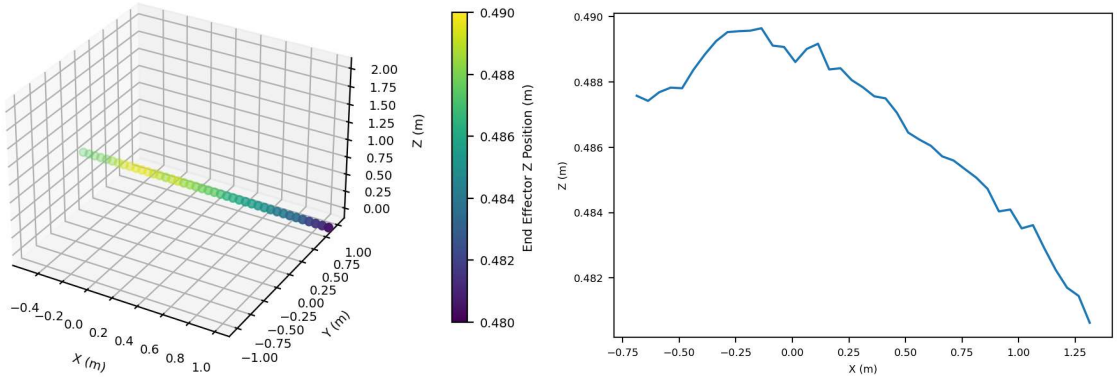

Figure 5.22. Forward Orientation Linear Test Discretized Results

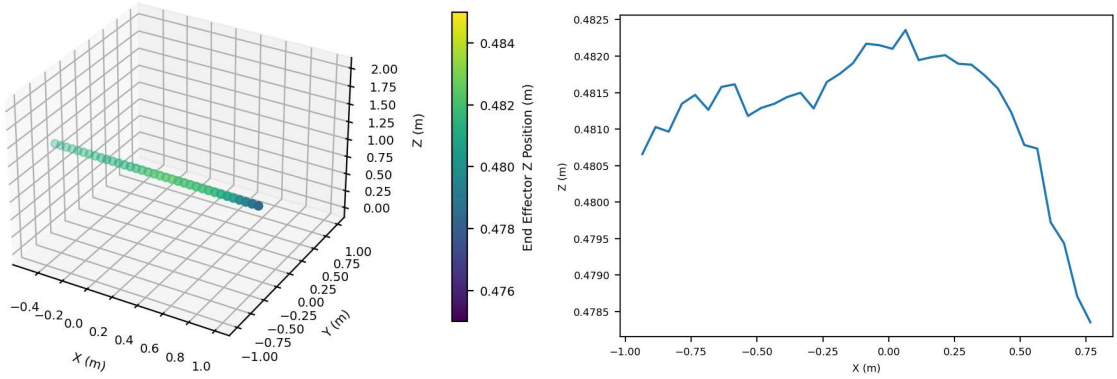

Figure 5.23. Backward Orientation Linear Test Discretized Results

The minimum and maximum Z coordinates recorded from the discretized data for all 6 orientations, as well as the total Z deviation for each test, is reported in Table 5.7.

Table 5.7. Z Deviation Data for All 6 Tested Robot Orientations

Orientation	Z_{min} (m)	Z_{max} (m)	ΔZ (m)
Up	0.4780	0.4850	0.0070
Right	0.4799	0.4883	0.0084
Down	0.4830	0.4865	0.0035
Left	0.4811	0.4855	0.0044
Forward	0.4806	0.4896	0.0090
Backward	0.4784	0.4824	0.0040

In addition to quantifying the non-linearity of the robot motion, the motion capture system was also used to correct robot programs to be linear within the motion capture coordinate system. To perform this correction, a linear test path was first generated and record using the procedure above. This path was referred to as the "calibration path". The Z deviations from the expected coordinates in the calibration path were tabulated. A new path was then generated that moves along the same line as the calibration path with the same orientation, however with different target locations along that line. For each target location in the new path a required Z correction value was calculated by interpolating the Z deviation values of the nearest two points from the calibration path. This correction factor was then added to the Z coordinate of the new path, and the updated coordinates were written back to the robot job file. To test this methodology for path correction, an "uncorrected" and a "corrected" path were both recorded using the motion capture system. The test path selected for this test was in in the Up orientation, and the correction was based on a previous recording in the Up orientation that moved along the same line as the test path. Figure 5.24 and Figure 5.25 display the results of this motion capture based path correction for a linear path.

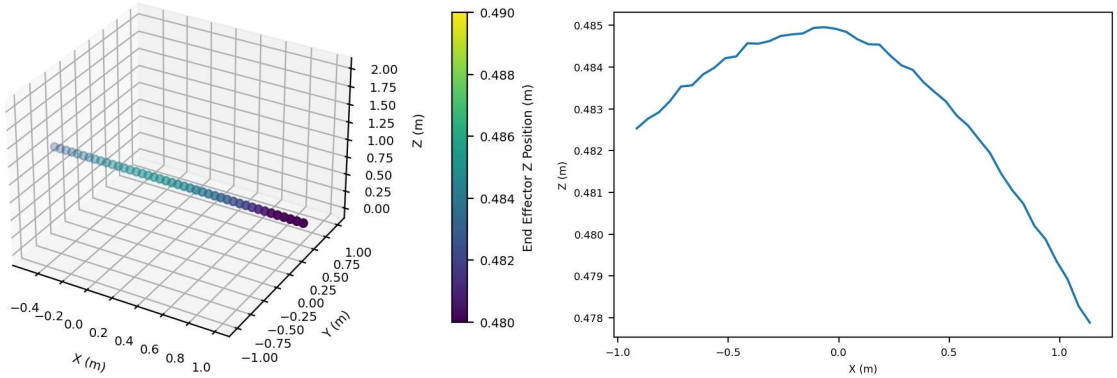

Figure 5.24. Uncorrected Linear Test Discretized Results

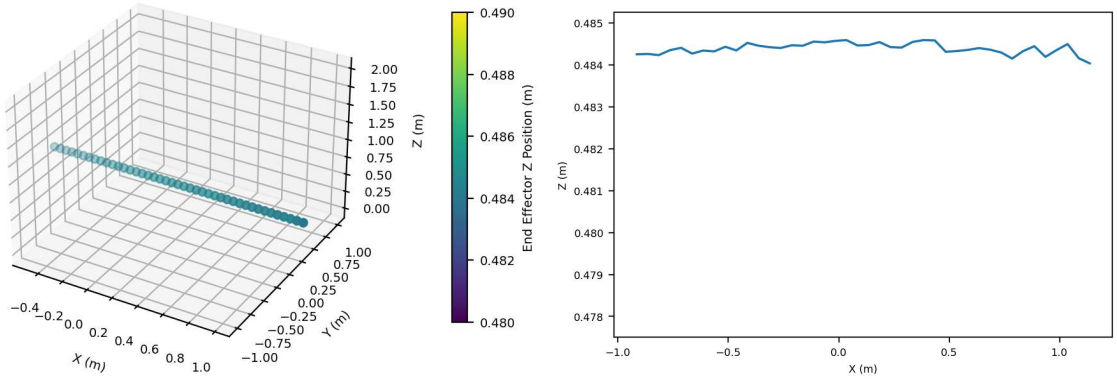

Figure 5.25. Corrected Linear Test Discretized Results

The uncorrected test resulted in a total Z deviation of 7.0mm across the path, while the corrected path resulted in a total Z deviation of 0.56mm across the path.

Similar tests were performed for robot motions within a plane at a set orientation. A planar path was constructed for R05, using the DT to avoid collision and configuration issues with the robot over the course of the path. The path was created with even 100mm spacing between target locations, and with a pause at each target location to allow the robot to stabilize and to make discretizing the data easier. The resultant path is shown in the DT in Figure 5.26.

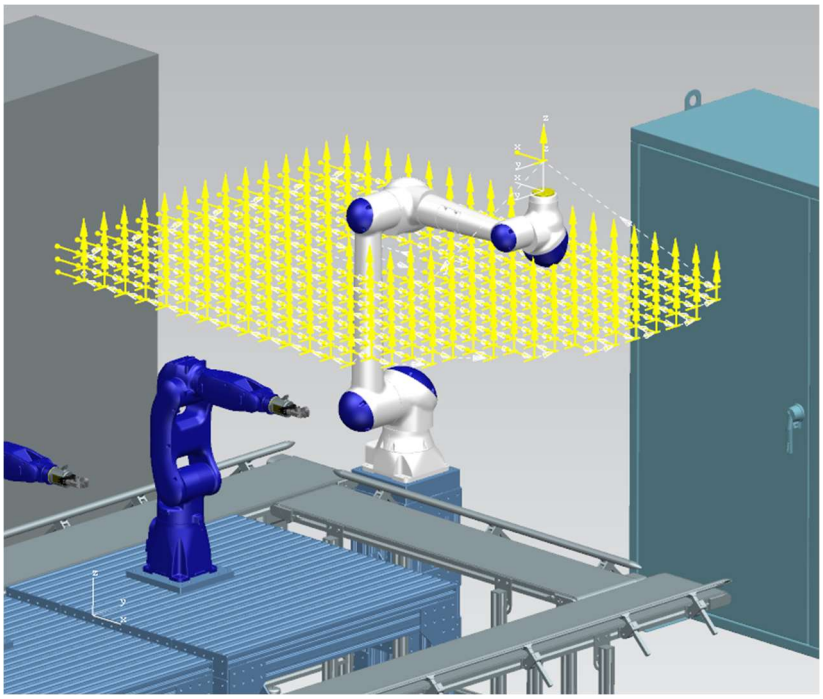

Figure 5.26. The Planar Test Path for R05

In a similar manner to the linear tests, the robot was allowed to run through this path while its motion was recorded using the motion capture system. These results were then discretized and are presented in Figure 5.27.

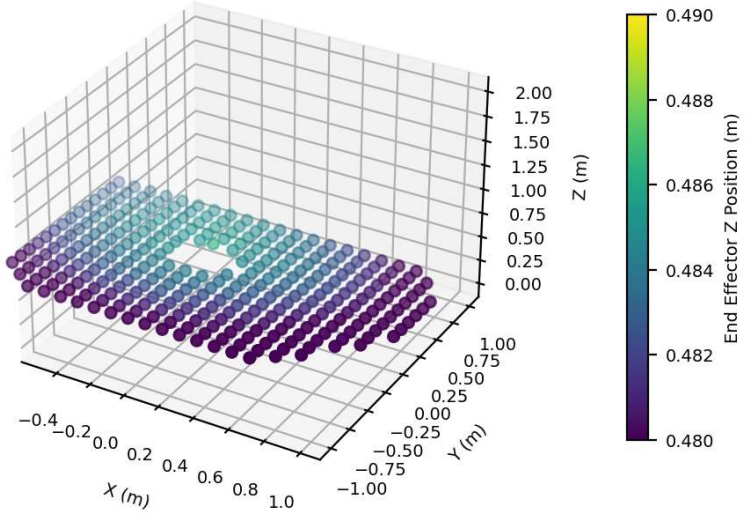

Figure 5.27. Results from the Planar Test in the Up Orientation

Across this planar test, the lowest Z coordinate recorded from the discretized data was 0.4768m while the highest Z coordinate recorded was 0.4862m. This results in a Z deviation across the plane of 9.4mm for the Up orientation.

Similarly to the linear case, a path correction technique utilizing previous path recordings as a calibration path was demonstrated for planar paths. An uncorrected planar path was recorded, and this path served as a calibration path for any future paths that will be in the same plane in the same orientation. Z deviation values were calculated and recorded to create an offset map for any path in the plane. A new path was then created in the same plane, however all of the coordinates of the new path were offset in X and Y from the calibration path. A 2D interpolation algorithm was then used to calculated the appropriate offset for each point in the new path based on the results from the calibration path. The interpolation algorithm used in this experiment was scipy.interpolate.griddata(). The calculated Z offset for each point in the path was then added to the Z coordinate of each point, and then written back to the robot job file. This job file was then run on R05, and the resultant motion was recorded. The results from the uncorrected planar path are shown in Figure 5.28, and the results from the corrected planar path are shown in Figure 5.29.

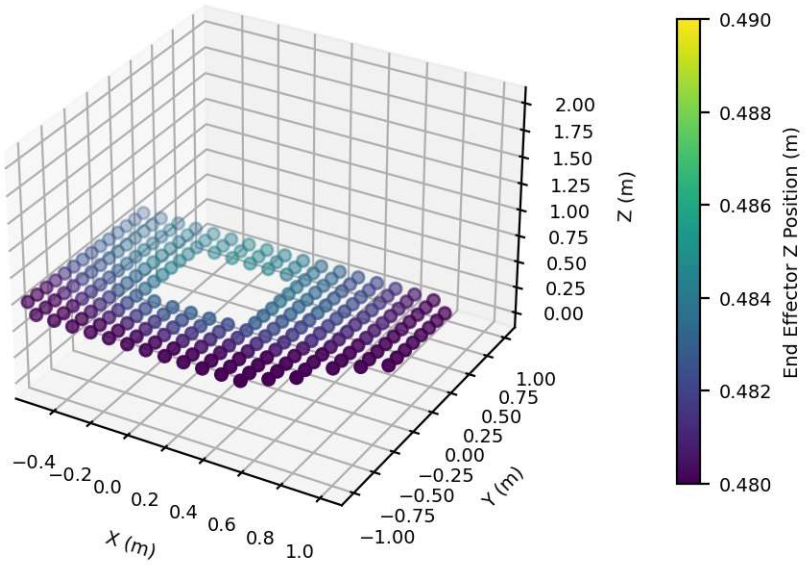

Figure 5.28. Uncorrected Planar Path Discretized Results

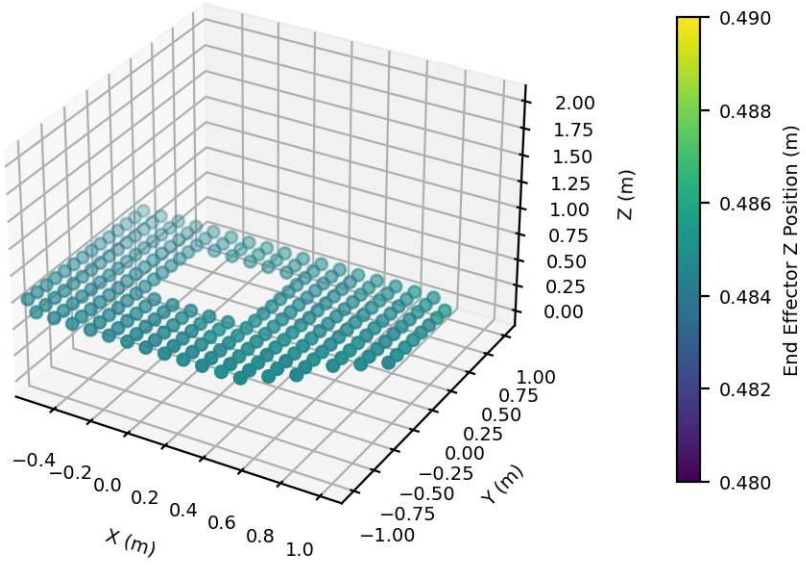

Figure 5.29. Corrected Planar Path Discretized Results

The final recorded Z deviation for the uncorrected planar test was 7.0mm, while the final recorded Z deviation for the corrected planar test was 1.0mm.

5.1.2.3 Discussion of Motion Capture Results

The linear and planar motion capture tests described above refute the assumption that the robot workspaces are perfectly cartesian and accurate. The first linear tests performed using R05 demonstrated clearly non-linear performance along paths specified as linear motion. This test showed a variation in end effector height by 6.8mm from high to low across the entire 2.1m long path. Additionally, this drop in Z height appeared to be somewhat symmetric around the base of the robot, dropping the full 6.8mm on one side of the robot, over approximately 1.2m. The other side of the path demonstrated a lower drop, but the path did not extend quite as far from the base of the robot. The planar tests showed similar results, indicating a total Z deviation over the tested space of approximately 9.4mm, with a similar rotationally symmetric drop off around the base of the robot. These results lead us to believe that the symmetric drop in Z around the base of the robot is due to the weight of the robot arm itself, and that this drop in Z would be more exaggerated if the robot were carrying more load. We also expect that this result is more exaggerated on long arm robots such as the HC10 due to the large extent of motion possible by the arm and the relatively low structural rigidity of the system. We also note that these results affect only the accuracy of the system to the planned coordinates, not the repeatability of the system which appeared to be within manufacturer specifications.

Additional linear tests with different end effector orientations demonstrated similar results to the first linear test. A similar drop in Z position of the tracked end effector was observed across 6 different test orientations, as shown in Figures 5.18 – 5.23. For all tested orientations, the drop in height was most extreme at the limits of motion of the arm, in locations where the end effector is farthest from the base. The observed curve, however,

varies substantially between different orientations. The various tests demonstrated different high points in their motion, different total Z deviations across the path, variations in smoothness of path, and discontinuities that show that the results from one robot orientation are not transferable to another. This means that a calibration based on one of these curves is only applicable to paths along the same line and with the same orientation.

Successful correction of both a linear path and a planar path was demonstrated using the motion capture system. The uncorrected path and corrected paths shown in Figures 5.24 and 5.25 and in Figures 5.28 and 5.29 for linear and planar tests respectively demonstrate a drastic reduction in variation between the uncorrected and corrected paths. This test demonstrated the effectiveness of a simple interpolation model in correcting a robot path based on a previously calibrated workspace. While these 1 and 2-dimensional tests are simplistic and limited to paths within their calibrated areas, they demonstrate the potential for a motion capture based system to drastically increase the positional accuracy of a relatively inaccurate robot system, for minimal additional cost. This motion capture based calibration technique also shows potential to increase the usefulness of the DT model, as programs authored from the DT model in the motion capture workspace could be made far more accurate by this technique. Should this technique be made applicable to the entire robot workspace instead of just linear paths, this increase in accuracy will in turn result in a decrease in setup time and a decreased reliance on touch based calibration of UFRAMEs and UTOOLs, as well as a decreased reliance on manual touch up of robot paths after programming.

5.2 DIGITAL TWIN EXPERIMENTS AND RESULTS FOR AUTOMATED FIBER PLACEMENT

In addition to the implementation tested in the FF lab, the DT developed for the AFP lab was also tested. The AFP DT system was evaluated across three main criteria: DT layout accuracy, DT motion accuracy, and ACSIS scanning. These components were assessed according to both qualitative and quantitative methodologies, and the results of these analyses are described below.

5.2.1 Digital Twin Layout Accuracy

Similarly to the FF lab, the accuracy of the DT model of the AFP lab was paramount to the success of the DT system. The accuracy of the physical layout of components in the DT model of the AFP lab was evaluated using two techniques: 3D scanning and reconstruction of AFP motion in the DT model. As shown in section 2.1.2 of this chapter, 3D scanning provides an accuracy positional analysis of components in the cell. Reconstruction of the AFP motion from recorded machine coordinates provides a means to assess the accuracy of the kinematic model in the DT by reproducing machine paths as recorded in the context of the machines and tooling in the DT. Deltas between observed and expected machine paths provides a means to assess the layout accuracy of the cell, and any non-linear deltas between the observed and expected paths provide feedback on the accuracy of the kinematic model of the AFP machine in the cell.

5.2.1.1 3D Scanning Results

The Faro 3D scanning system described in chapters 3 and 4 of this thesis was used to create a composite 3D scan of the AFP lab. To ensure this scan provided useful information on the accuracy of the model of the Lynx machine, the machine was jogged to

a specified location on each axis before the scan was taken. These axis values were then entered into the DT to ensure the position of each axis inside the DT matched the values of the physical machine. The resultant raw scan was first presented in chapter 4 of this thesis. This raw scan was then refined to remove unnecessary point data. This trimmed scan is presented below in Figure 5.30.

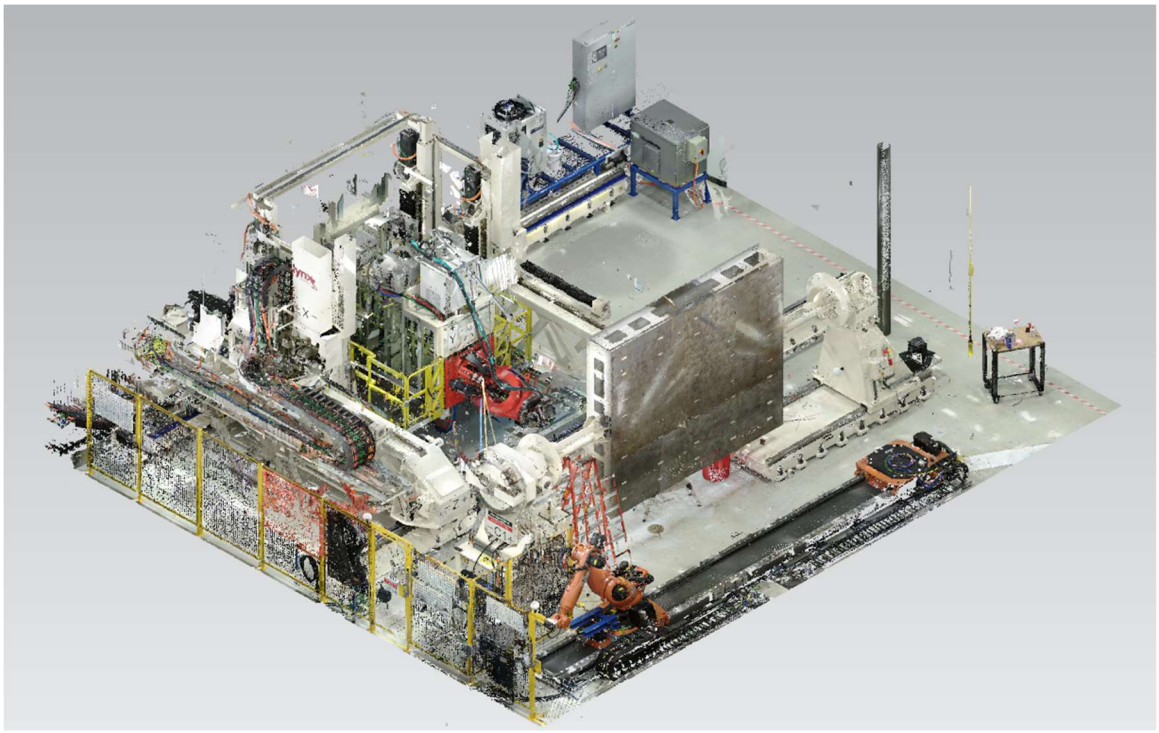

Figure 5.30. The Resultant Trimmed 3D Scan of the AFP Lab

The AFP lab had many more reflective surfaces and cases for obscured details than the FF lab. These cell features caused many scanning artifacts in the raw scan data. A clear image of "phantom" points resulting from reflectivity can be seen in in Figure 5.31 below, where multiple instances of the AFP machine's Z carriage can be seen offset from the original, true, location. These duplicate points of already scanned objects were removed wherever possible, to provide easier analysis of the 3D scan data.

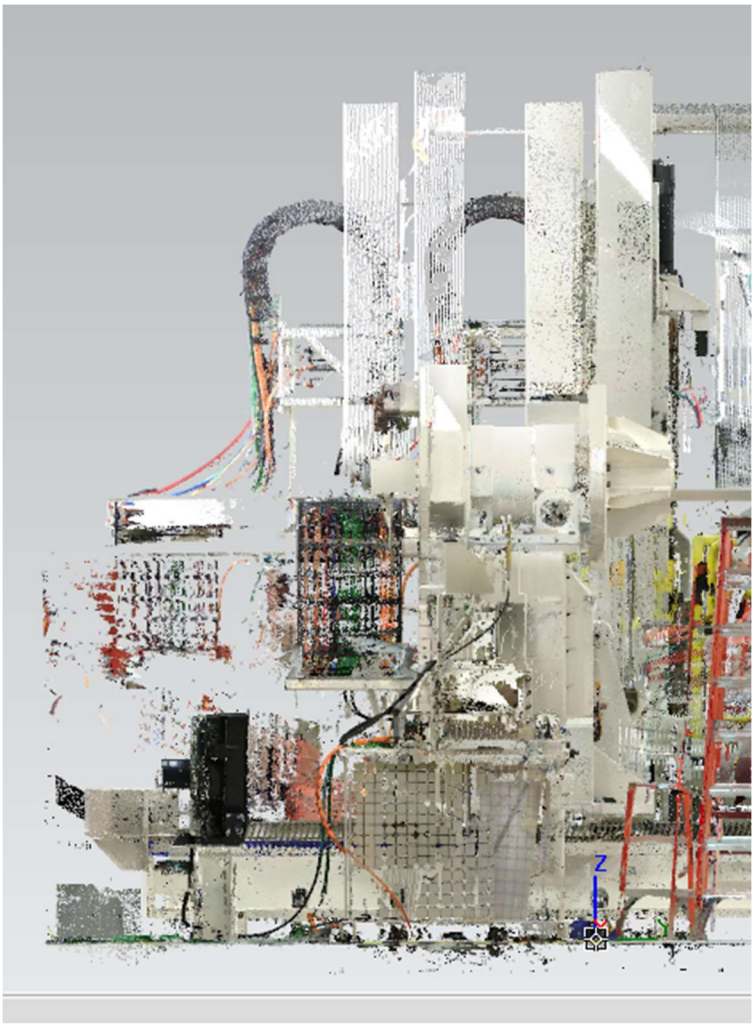

Figure 5.31. “Phantom” Images of the Lynx Z Carriage in the 3D Scan

In the AFP lab, the 3D scan data was primarily used to analyze the placement of key components inside the cell, and to verify the accuracy of the kinematic model of the Lynx AFP machine used. This analysis was performed by overlaying the 3D scan on top of the DT model of the AFP lab and measuring key distances between points in the DT model and their corresponding location in the 3D scan. The 3D scan was aligned with the DT model by aligning a key component in the 3D scan with the same component in the DT. In this case, the floor mounts of the Lynx AFP machine were used, as these locations were clear in both the 3D scan and the DT, and the location of the AFP machine itself is a

critical dimension in the context of the AFP lab. Additionally, the use of a large component like the base of the AFP machine allows for alignment of the 3D scan against the DT model in both position and rotation. Figure 5.32 shows the cleaned 3D scan overlayed on top of the DT model.

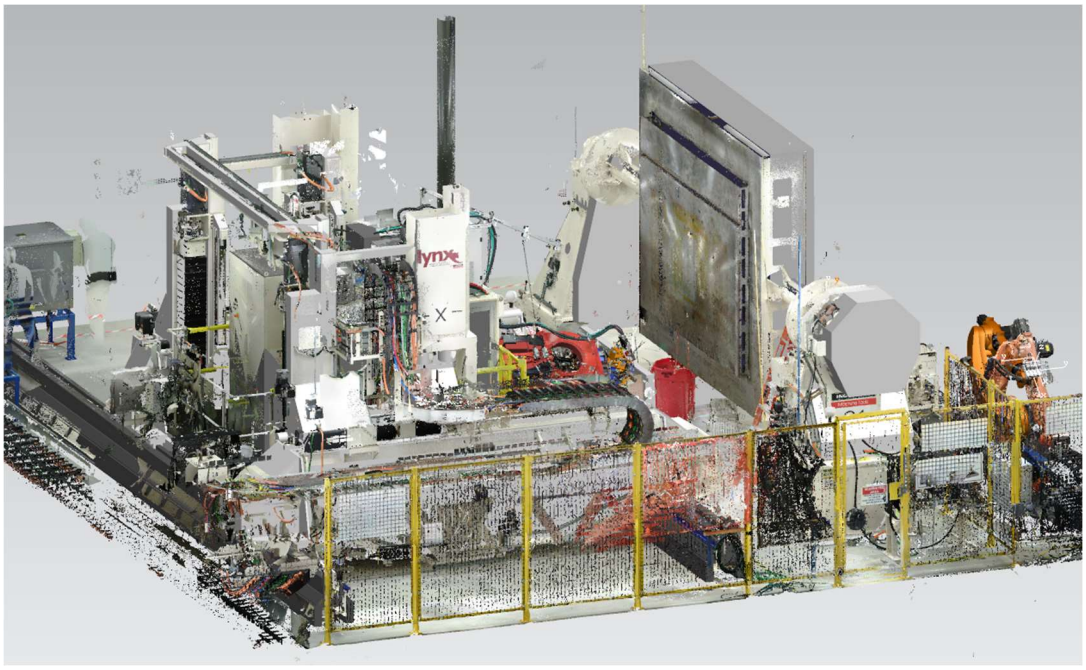

Figure 5.32. The 3D Scan Overlayed On Top of the DT Model of the AFP Lab

Analysis of the overlayed scan in Figure 5.32 revealed several inaccuracies in the DT model of the cell. While many components are accurately located in the context of the scan, some components vary substantially from their expected location. Of note are the locations of the mandrel relative to the base of the AFP machine, and the resultant location of the AFP head relative to the base of the AFP machine. These discrepancies are shown in Figure 5.33 below.

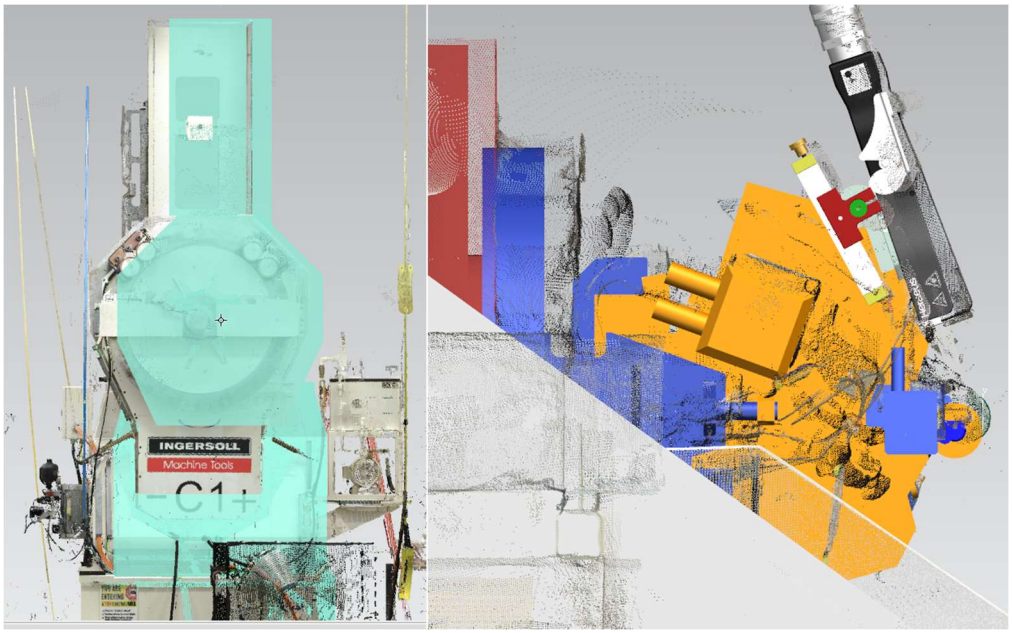

Figure 5.33. Discrepancies Between the DT and the 3D Scan for the Mandrel Location and AFP Head Location

Overlaying the 3D scan with the DT allows for a relatively easy correction for the mandrel placement: the mandrel location inside the DT can simply be adjusted to align with the mandrel location in the 3D scan. The results of this shift are discussed in section 5.3.1.3 below, with analysis based on path reconstruction from machine coordinates. A similar shift was performed to correct the locations of the ACSIS scanning system and rail relative to the mandrel location, however this shift is less impactful, as the tooling process used for ACSIS accounts for these offsets when programming.

Correcting for the delta in head position was more difficult. The error in head position could be the result of any number of inaccuracies in the kinematic model of the Lynx machine. To locate potential errors, the location and length of each axis of the machine in the DT model was compared against the 3D scan. Through this analysis, a difference in the X axis of the machine between the DT and the 3D scan was observed. This delta is shown in Figure 5.34 below.

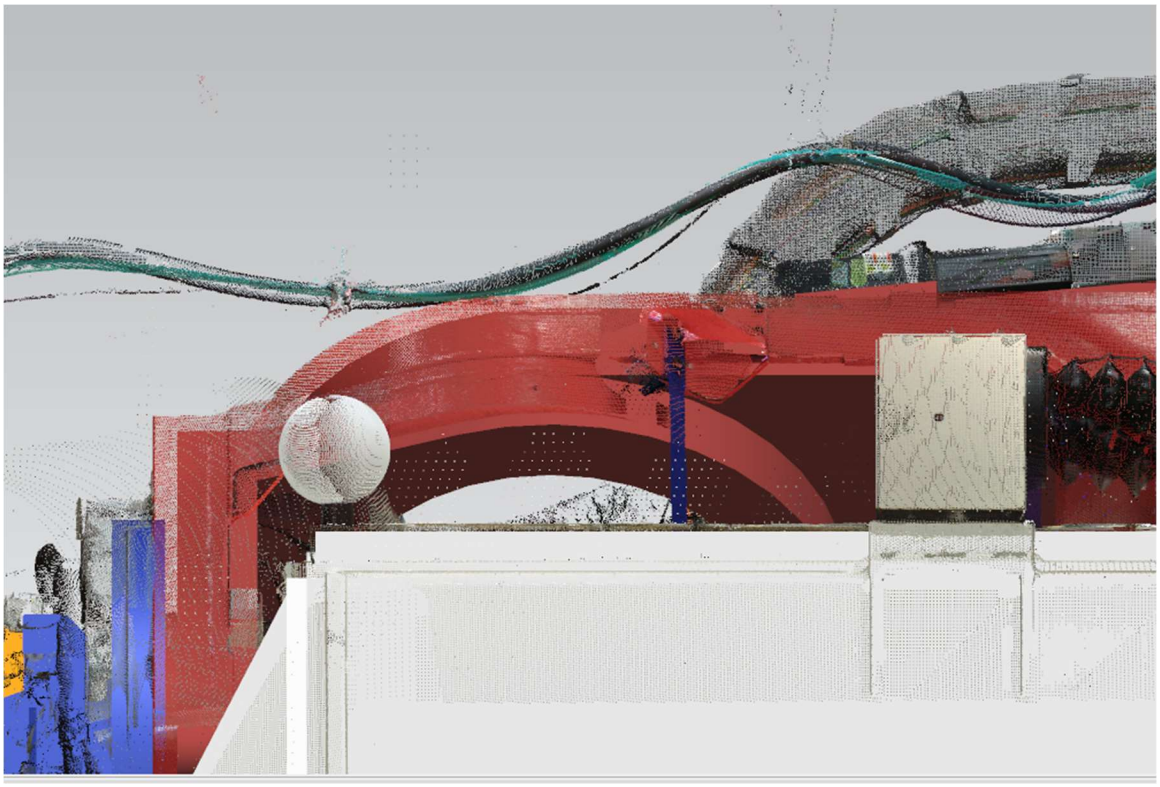

Figure 5.34. The Delta Observed Between the DT Model and the 3D Scan for the X Axis of the Lynx Machine

This delta was corrected in the kinematic model of the Lynx machine, and this updated model was added to the corrected version of the DT. Various other corrections were performed for location of non-critical components in the cell (such as fencing, walls, desks, etc.), and the final updated DT model is shown with the overlayed 3D scan data in Figure 5.35 below.

5.2.1.2 AFP Machine Motion Results

The analysis of recorded motion from the Lynx machine during layup provides another comparison point for the accuracy of placement of components in the DT cell. Additionally, as the paths reported by the machine are reliant on the kinematic model in the DT, reconstruction of these paths in the DT model provides another means to assess the accuracy of the kinematic model of the Lynx in the DT model.

To reconstruct the motion of the Lynx machine during layup, coordinates from all machine axes were recorded over OPC every 100ms. These coordinates were stored in a SQL database and retrieved later for analysis. Each timestamp contains coordinates for all machine axes, as well as the current layup speed, heater power, and compression set point. The axis values for each timestamp were extracted and saved as a comma separated value file, and these coordinates were then used to recreate the machine motion inside the DT model. For each ply, the relevant axis data csv file was loaded into the Process Simulate model using a custom API script that automatically set the joint values for the Lynx model inside Process Simulate to the values from the csv, and recorded the resultant TCP location as a path location inside the model. The script then repeated this process for every timestamp in the ply, and structured the resultant locations inside PS as a sequenced Ply operation assigned to the Lynx model. This use of the Process Simulate API as a "forward kinematics engine" allowed for the recreation of machine paths in the DT model, however this approach relies on the accuracy of the kinematic model of the Lynx machine in the DT. The recreation of the machine motion for ply 0101 from the curved tool layup is shown in the DT model in Figure 5.35 below. Note that this recreation was performed using the original uncorrected version of the DT that did not use the 3D scan results to refine the cell layout and kinematics.

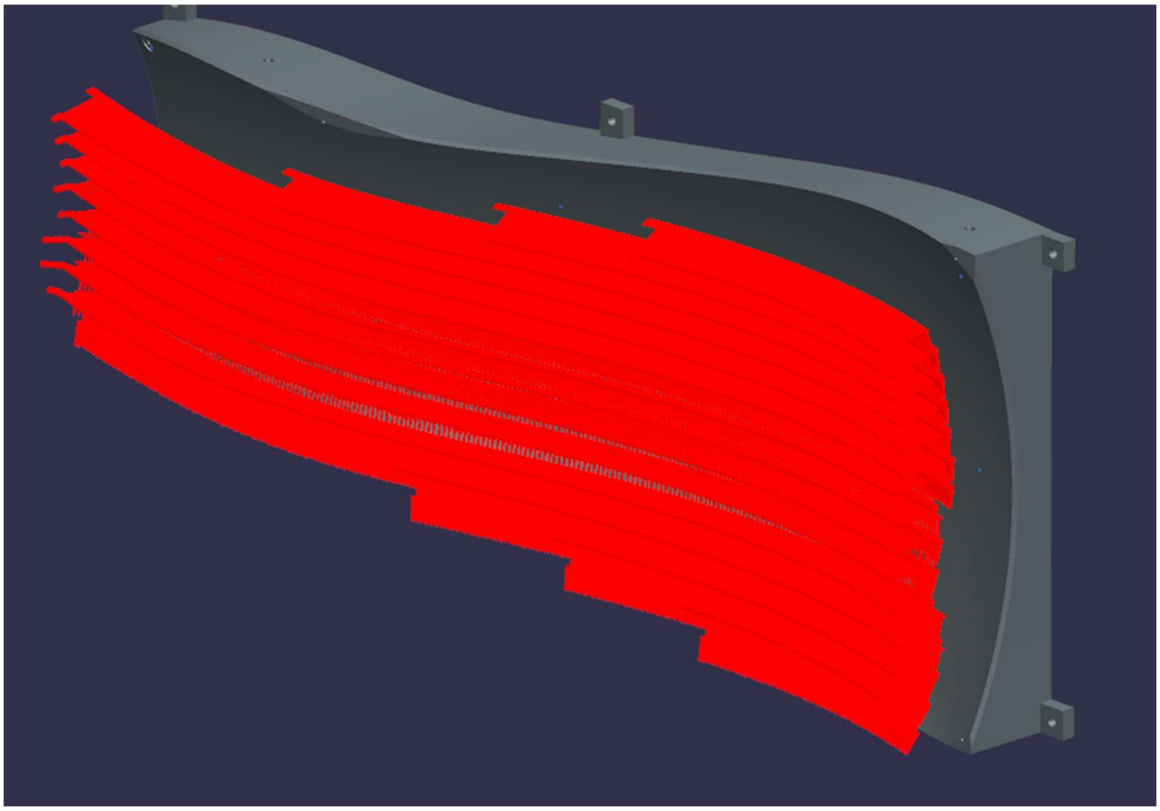

Figure 5.35. The Reconstruction of Ply 0101 Using Forward Kinematics in the Uncorrected DT Model

Figure 5.35 shows a clear offset between the tool surface and the reconstructed machine path. Using the Process Simulate API to iterate over all recorded points, this offset was calculated to have a mean point error of 112.6 mm. This suggests a substantial inaccuracy between the layout of the DT and the actual AFP lab. Additionally, the delta between the expected and reconstructed paths is not constant; the reconstructed paths have a different "shape" than the original paths. This suggests a kinematic inaccuracy in the modeling of one of the four rotational axes (A1, K, A2, or C1) of the Lynx machine. This inaccuracy, however, is expected based on the previous results of the 3D scanning tests.

The sources of this inaccuracy could likely be determined purely from the machine data and an iterative refinement process using the DT, however the 3D scanning results described above provide a great starting point for refinement. Repeating the path

reconstruction of Ply 0101 described above with the updated DT model based on the 3D scanning results yielded the path reconstruction shown below in Figure 5.36.

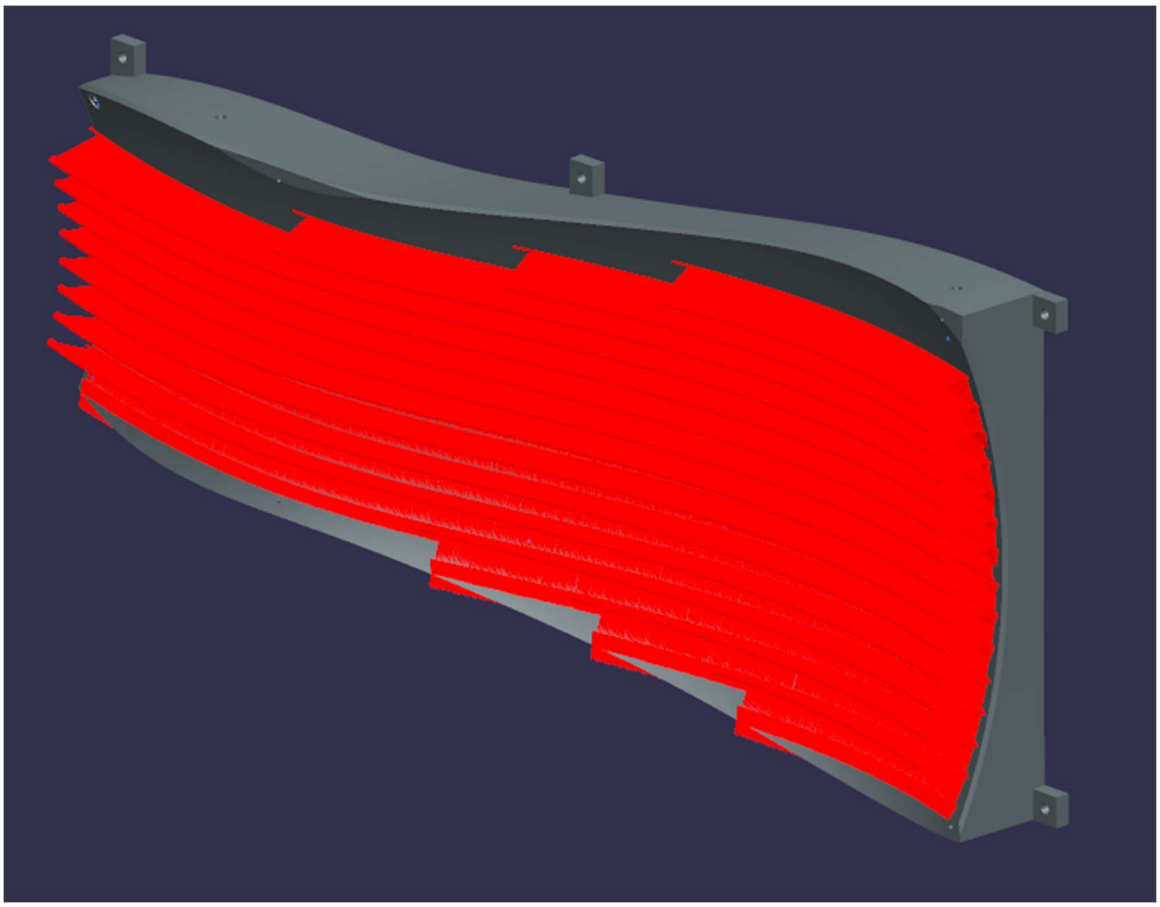

Figure 5.36. The Reconstruction of Ply 0101 Using Forward Kinematics in the Corrected DT Model Based on 3D Scanning Results

The path reconstruction shown above in Figure 5.36 demonstrated a drastic reduction in mean point error. Using the updated DT model based on 3D scanning results yielded an average point error of 1.0 mm over the entire ply.

5.2.1.3 Discussion of Layout Results

The results from both the 3D scanning and motion reconstruction methodologies demonstrated a useful means for assessment of DT model positional and kinematic accuracy. The use of 3D scanning provides a relatively easy to use environment for

comparison where discrepancies between the real and virtual systems can be easily identified. Two major differences between the virtual and physical systems were identified using 3D scanning, and these discrepancies were corrected relatively easily in the DT model. The integration of 3D scanning data directly into the DT environment made this refinement process fast and intuitive.

The use of motion reconstruction techniques in the DT model also validated the changes made based on the 3D scan data. When path reconstruction was performed using the original unedited DT model, substantial discrepancies were observed between the predicted and reconstructed paths. After updating the model using the 3D scan data, these discrepancies dropped substantially and thus it can be assumed that the layout and kinematic accuracy of the DT model was significantly improved. With the model in this more calibrated state, path reconstruction based on machine motion coordinates demonstrates a viable methodology for comparing physical system performance against predicted DT models. In addition to providing a secondary check to the 3D scanning results, path reconstruction can also independently offer refinement to the layout of the DT system, as well know sample paths could be constructed and used a calibration set for the DT model.

5.2.2 Path Accuracy and Motion Comparison – Layup

As described in chapter 4 of this thesis, an example layup was performed to test the efficacy of the DT system in a real world manufacturing case. This layup allowed for testing of many systems, such as motion reconstruction using the DT, comparison of machine motion against motion capture results, and programming of a robotic inspection

system in the integrated DT system. One ply from this layup was previously used in the motion reconstruction described in section 5.3.1.2 above.

5.2.2.1 Layup Experiments

Using the techniques described in chapter 4 of this thesis, a sequence of plies was created and programmed using the DT tools. The stacking sequence chosen for this experiment was [0, 0, -45, 90, 70]. This sequence was selected randomly, per requirements from other research utilizing the same equipment. The last ply in the sequence (70°) was selected as a non-standard ply angle to test models developed by other members of the research team. For the purposes of the research described in this thesis, any ply angle would suffice.

After programming and simulation in the DT, the plies were posted to the Lynx machine as .PPF files (Posted Ply Files). These PPFs contain all required machine motion to manufacture the ply, and would only be edited by the tooling process on the machine. The tooling process on the Lynx was performed using an automated laser tooling system to locate the tool surface in the cell. These tool definitions were applied to the PPFs, and were also exported to the DT to help locate the tool surface in the DT model. After programming and tooling were performed, dry runs of the machine were performed on each ply to ensure that the programs would execute without collisions. These dry runs were performed without collision across all plies, however in some cases it appeared that the physical system was closer to collision than predicted by the DT.

Layup of the programmed and tested sequence was then performed. An image of the machine during layup is shown below in Figure 5.37. During layup, machine coordinates and various parameters were recorded and stored using the OPC and SQL

system described above in section 5.2.1.2. These parameters would later be imported into the DT system and used to characterize the system in the context of the virtual model.

Figure 5.37. An Image of the Lynx Machine Performing Layup on the Curved Tool Surface

5.2.2.2 AFP Machine Motion Results

Using the updated DT model based on the 3D scan results, AFP motion path reconstruction was performed using the same methodology described in section 5.3.1.2 for all plies in the layup. Paths were reconstructed using the forward kinematic model inside the Process Simulate DT system, and the process of path reconstruction was automated using the Process Simulate API. As an example, the recreation of the motion of ply 0105 is presented below in Figure 5.38, and the intended motion for ply 0105 is presented in Figure 5.39. The same analysis was performed for all plies in the layup, and these results are presented in Appendix A of this thesis. Also note that while all paths are presented as images in this report, they are available as playable simulations in the DT model.

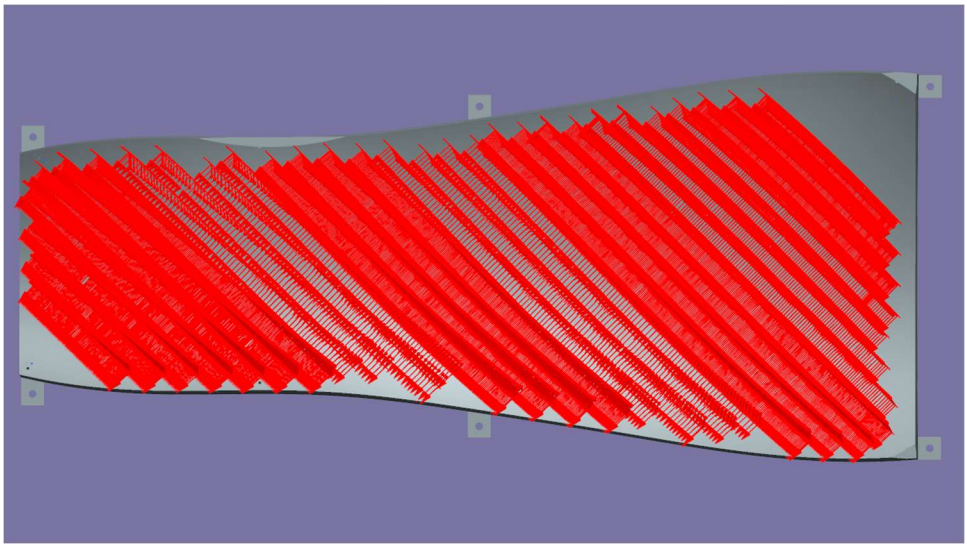

Figure 5.38. Machine Motion Reconstruction for Ply 0105

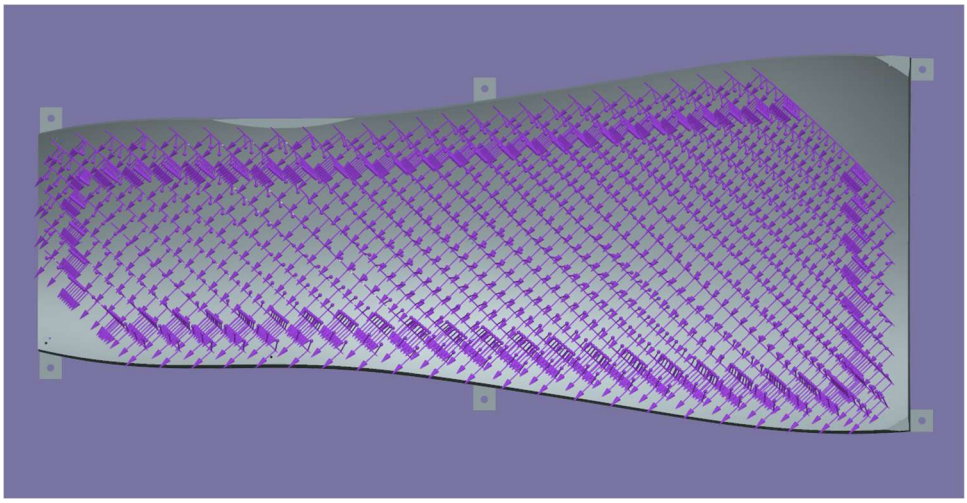

Figure 5.39. Planned Machine Motion for Ply 0105

After path reconstruction, all plies were compared with their expected paths from the programming stage. This comparison was performed using an API script that measures the delta between each point of the reconstructed path and the closest point on the original programmed curve. The direction of this delta was also recorded, and this data was used to create a vector field representing the error of the reconstructed path as compared to the original path. This comparison is shown for Ply 0105 in Figure 5.40 below. For ply 0105, this resulted in an average error over the course of the ply of 1.3 mm. The results for all

remaining plies are included in Appendix A of this thesis. The mean point errors for each ply are shown below in Table 5.8.

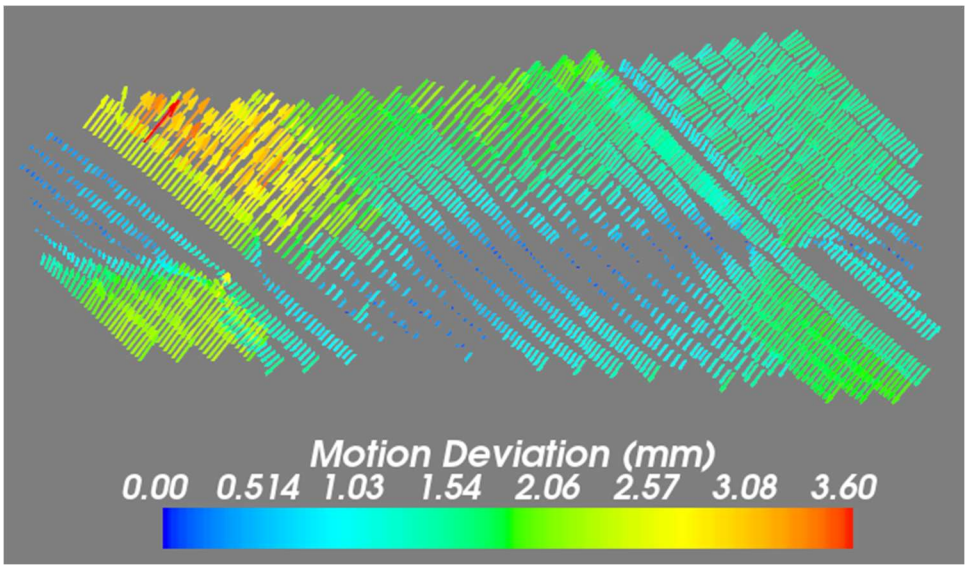

Figure 5.40. Error Vector Field for the Motion Reconstruction of Ply 0105

Table 5.8. Average Point Errors for Motion Reconstruction of All Plies

Ply	Average Deviation (mm)
0101	0.99
0102	0.92
0105	1.26
0112	3.53
0117	1.23
Average	1.59

5.2.2.3 Motion Capture Results

Motion capture techniques were also used to analyze path accuracy of layup paths performed by the Lynx AFP machine. The system described in chapter 4 of this thesis was used to perform this analysis. Before capture, the tracked point on the rigid body representing the AFP head was adjusted to coincide with the TCP of the machine. This adjustment allowed for direct comparison of motion capture paths with all other TCP

referenced paths inside the DT (such as the original programmed paths and the reconstructed motion paths described above). As mentioned in chapter 4, motion capture was only performed during dry runs of plies to avoid any potential damage to the tracking system from the intense light and IR emissions during the layup. Additionally, due to time constraints during the experiments, motion capture results were only collected for the first two plies of the layup: Ply 0101 and Ply 0102.

The analysis of motion capture data recorded during the dry runs of Plies 0101 and 0102 was performed using similar tools to those employed in the FF lab. The data collected in the AFP lab, however, was far noisier than the data from the FF lab, and as such had to be smoothed before it could be worked with in the DT. The built-in smoothing tools in the Motive software were used to reduce noise in the motion capture path, and the resulting data was saved as a .csv file. This file was then loaded to the DT environment using a custom tool built in the Process Simulate API. The resultant curve is visualized in the DT environment as a polyline, and rendered using the DT. This curve can then be used by other analysis tools inside the DT environment. The recorded and smoothed motion capture paths for plies 0101 and 0102 are represented in Figure 5.41 and Figure 5.42 below.

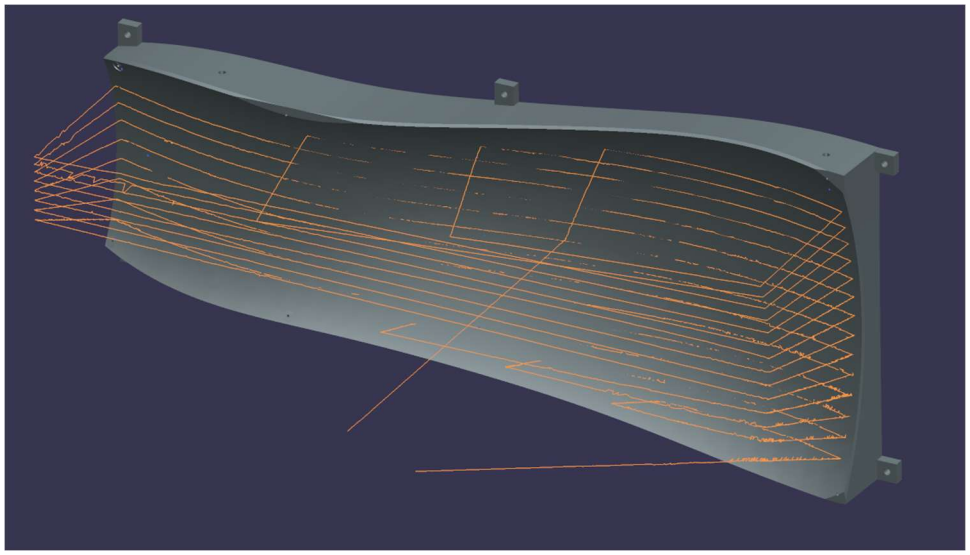

Figure 5.41. The Motion Capture Recording for Ply 0101

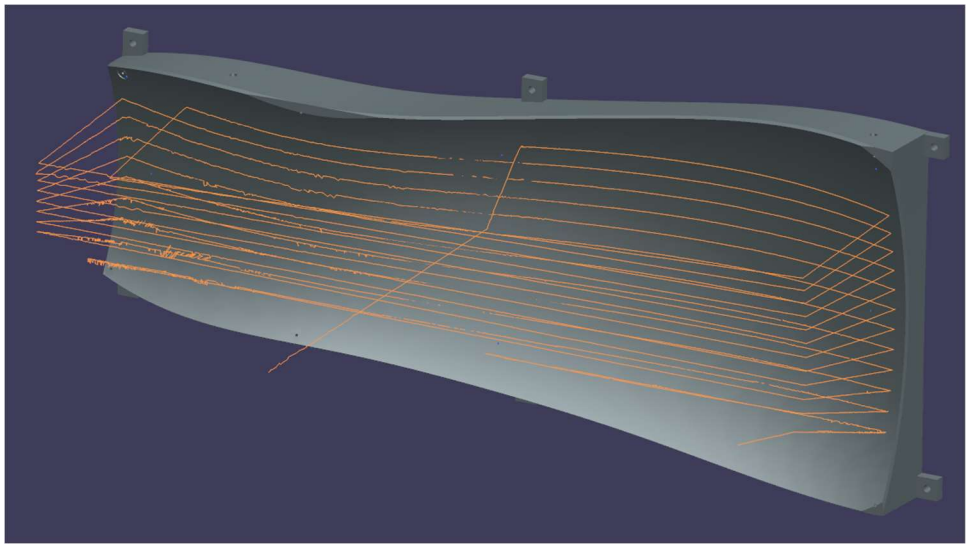

Figure 5.42. The Motion Capture Recording for Ply 0102

A similar analysis to that applied to the motion reconstruction results was then applied to the motion capture results. A custom-built API tool was used to compare the recorded paths with the expected (planned) paths and report the deltas in both magnitude and direction. This data was then used to create a vector field representing the error for both plies 0101 and 0102. These vector fields are shown below in Figure 5.43 and Figure

5.44. Analysis of these error fields yield average point errors of 1.72 mm and 1.65 mm for plies 0101 and 0102 respectively.

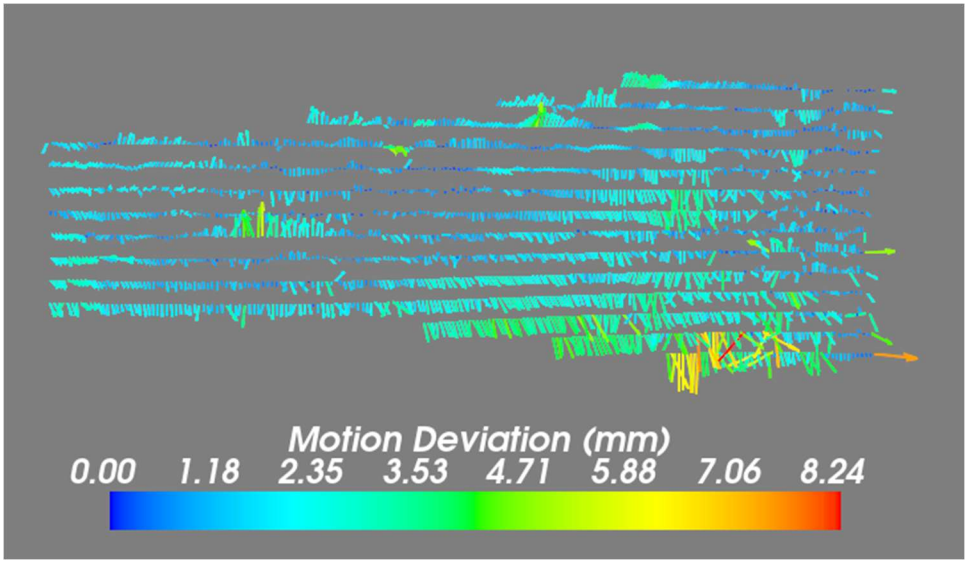

Figure 5.43. The Error Vector Field for the Motion Capture from Ply 0101

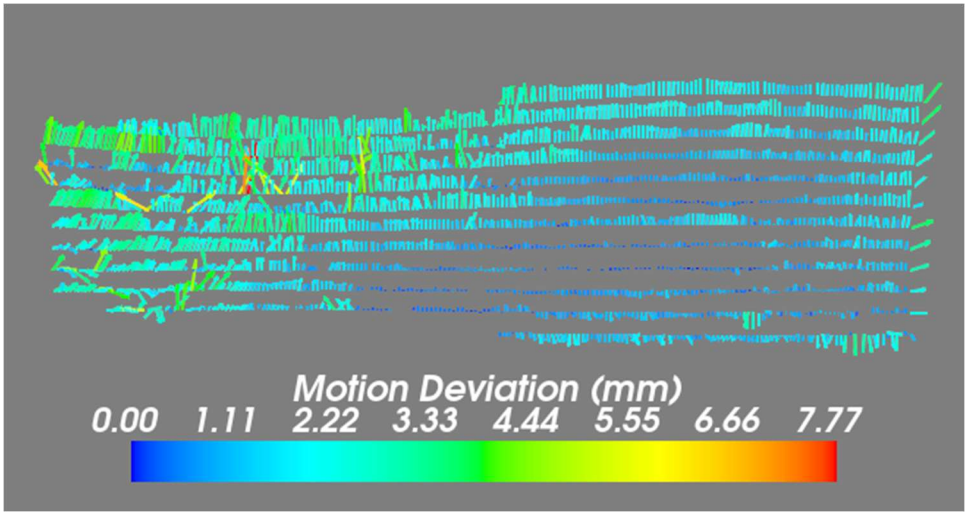

Figure 5.44. The Error Vector Field for the Motion Capture from Ply 0102

5.2.2.4 Discussion of Motion Results

Analysis of the machine path reconstruction results presented above indicates that the DT model can successfully be used as a forward kinematics engine to recreate toolpaths from axis coordinates from the AFP machine. These path reconstructions, on average, were

accurate to within 1-2mm of the expected locations of the machine, across the entire path. While this level of error is far higher than the expected accuracy of the machine itself, these reconstructions can be successfully utilized to correlate various process parameters recorded during the layup with their cartesian locations in the actual part, by providing a tool for translation between the time and positional domains. This 1-2mm error is also sufficient for visualization of paths and process parameters in the DT model and contextualization of this data against the original designed paths.

This path accuracy error, however, is not sufficient for machine motion quality analysis or comparison of actual machine motion against planned values. Additionally, the results presented in the error vector fields in Figure 5.40 and in Appendix A of this thesis show clear trends indicative of systematic errors in this analysis of machine axis data. Analysis of the error vector field for ply 0105 shows a clear line of symmetry through the field, approximately halfway up the ply. Similar symmetric features were observed in the vector fields for all other analyzed plies and can be seen in Appendix A of this thesis. These symmetric features likely indicate an error in the kinematic definition of the Lynx AFP machine, either in mandrel placement or in the axis definitions of the machine itself. As the error is not simply a linear shift, we expect this error to manifest in the definition of at least one of the rotational axes of the DT model.

Analysis of the motion capture results leads to similar conclusions for motion capture-based path reconstruction. The relatively small average deltas between the motion capture results and the original planned motion paths indicates a relatively high degree of accuracy for the motion capture results of the two plies. Average errors of 1.72mm and 1.65mm for plies 0101 and 0102 respectively indicate that the motion capture system is

successfully recording positional data of the AFP head and indicating a relatively accurate position of that head over the course of the layup. These errors, however, are substantially higher than the errors observed in the FF lab and are prohibitively high for performing machine motion analysis based solely on the motion capture results alone. The AFP process requires precision down to roughly 0.1mm to ensure positional accuracy of the placed tows, and as such, these motion capture recordings do not provide accurate enough data to draw conclusions on the accuracy of the machine motion or provide input on the prediction of defects in the final part.

Compared to the resultant motion captures from the FF lab, the AFP motion capture results display a far higher noise level, even after post-processing. In particular, the recording of ply 0102 shows numerous cases where the motion path appears to "jump" from one location to another, substantially off the expected path. This "jumping" occurs when tracking targets on the rigid body come into and out of vision of the cameras, and as reflections in the AFP lab create false images of targets in the Motive software. Motive attempts to correct for these factors through post processing, however in this case these corrections were not enough to increase the accuracy of the motion capture system to be sufficient for motion analysis.

Analysis of the vector field results presented in Figures 5.43 and 5.44 indicates that the positional error of the motion capture system is not constant across the plies. This could be due to errors in the AFP machine motion, however the high noise level of the data indicates that this error is likely due to the motion capture system itself and not indicative of an issue with the machine. The fact that these vector fields do not show a uniform direction or magnitude across the ply may be indicative of changing tracking conditions

during each capture. Additionally, unlike the results from the machine motion reconstruction, these motion capture vector fields show no consistent patterns in the field, indicating that this error is not due to some consistent systematic source. Areas with lower error and constant direction error had lower noise levels in the motion capture, as shown in the right-hand side of the capture for ply 0102. Areas with higher error magnitudes and more random directions tend to be indicative of higher noise levels, as shown in the left-hand side of the capture for ply 0102. Comparing the vector fields for plies 0101 and 0102 indicate that no area of the capture had a consistently lower error, and indicates that changing lighting conditions, shadows, occlusions, and other non-constant factors likely heavily influenced the motion capture results.

Both kinematic motion reconstruction using the DT model and motion capture techniques yielded similar levels of positional error and present similar capabilities for motion recreation in the DT model. Neither technique could produce low enough levels of error to provide analysis of the motion itself, however both techniques were capable of visualization of the paths in the DT context and providing a means for understanding the impact of other various process parameters on the part program. The kinematic motion reconstruction techniques could be improved by improving the DT model itself, and by calibration of the Lynx AFP model. Future work should include the creation of a more accurate DT model of the Lynx or use of a calibration technique to validate and improve the accuracy of the DT. Motion capture results could be improved through use of more consistent lighting, better camera placements and more visible target locations. Future work in motion capture should involve the improvement of the capture itself. If the capture can be improved to the quality seen in the FF lab, motion capture could be used to provide

accurate path analysis down to 0.1mm, which would be sufficient for motion quality analysis.

5.2.3 ACSIS Scanning

During layup, inspection of each completed ply was performed using the ACSIS (Automated Composite Structure Inspection System) inspection system. The results of this inspection are not relevant this thesis, however, as described in chapter 4 of this thesis, programming of this system was accomplished using the DT system. The results of this programming process and the ease of use of the system are discussed below.

5.2.3.1 Programming Quality of Life Improvements

Utilization of the DT system to program ACSIS provides clear benefits over the existing process. ACSIS was traditionally programmed as a "black box" system, using custom code to post the exact course center lines to KUKA syntax, without validation or simulation. This technique is prone to errors and limited in capabilities, as the system will only ever post inspection programs that follow course centerlines exactly and does not allow for optimization or customization of scanning paths. The existing methodology also lacks in interoperability: paths created using this automated script or created by teach pendant are difficult to export and interface with other systems that require use of robot coordinates from paths.

Utilizing the DT techniques described in chapter 4 of this thesis for programming of ACSIS substantially improved the programming process for ACSIS. Programs were validated virtually in a calibrated DT model, ensuring that the programmer knows exactly how the physical system will perform. In our experiments, touch up or reprogramming of ACSIS was never required after a program was generated from the DT; all paths from the

DT functioned exactly as intended. This accuracy is due to the nature of the calibrated DT model and the use of aligned $BASE frames in the KUKA programs. Additionally, the use of the DT allowed the programs to be better optimized; the simulation allowed programmers to accurately predict scan coverage and reduce the number of required passes when possible. This optimization allowed scan paths to run faster while still maintaining adequate coverage of the ply. Finally, the use of the DT allowed other researchers easy access to ACSIS toolpaths for additional work, as the Process Simulate API exposes this data in an easy to access format.

5.3 CONCLUSIONS ON THE USE OF DIGITAL TWINS FOR MANUFACTURING

The creation, use, and testing of DT systems for manufacturing in both the FF and AFP lab has shown that DTs of manufacturing are powerful systems that can improve the output, usability, safety, and commissioning time of manufacturing systems. Both systems demonstrated the capabilities and usefulness of DT models and, in our opinion, provide justification for the creation of such systems for future manufacturing systems.

In the case of the FF lab, the DT provided an incredibly useful model for planning and testing of the lab before any hardware had even been purchased. The DT provided tools for offline programming and virtual commissioning of manufacturing equipment, improving the commissioning process, and reducing the time to production. Additionally, the DT model provided a means for testing manufacturing strategies in the virtual world, ensuring that techniques would work reliably and safely before they were implemented on the physical system. These testing and commissioning tools drastically improved quality of life while working on the physical system, as compared to simply using CAD tools.

Testing of the DT model of the FF lab against the physical system after construction additionally showed the usefulness of the DT system, as well as some potential drawbacks and pitfalls to avoid. The DT model of the FF lab was generally positionally accurate to the physical system where accuracy was important (robot and fixture locating dimensions, etc.), however accuracy was lacking on other components where location was less critical. It was important to be aware of the positional accuracy of various components in the lab when utilizing the DT model, as these layout errors could lead to path inaccuracies or potential robot collisions. The use of motion capture and 3D scanning systems drastically improved the accuracy and usability of the DT model. Additionally, the use of 3D scanning system also revealed inaccuracies in robot motion that were not predicted by the DT. Future work should involve correction of these inaccuracies using the motion capture system and the DT model in tandem.

The results from the AFP lab show similar utility of the DT model in a real manufacturing context. In the case of the AFP lab, the construction and layout of the physical system was already complete before the DT was developed, so in this case we did not see benefits of creating the DT before the physical system existed. The DT model of the AFP lab did, however, provide meaningful benefits during the programming and execution stages of manufacturing. Using the DT to program equipment allowed for the analysis of machine paths in an integrated environment, incorporating both the AFP and ACSIS inspection systems. The use of the DT model allowed inspection paths to be directly based on their corresponding AFP paths. Additionally, the DT allowed programmers to know exactly what the profilometers on ACSIS would be inspecting during each pass, relative to the programmed paths from the AFP.

The DT model also allowed for analysis of results from the AFP experiments to all be analyzed and correlated in one environment, and for these results to be contextualized against the original planned machine paths. This contextualization allows researchers to analyze how various design decisions impact the as manufactured part and could eventually allow for better design for manufacturability in the AFP space. This is further compounded by the use of kinematic path reconstruction and motion capture techniques to recreate real machine motion in the DT model and provide a mapping for time series data and process parameters back to the original paths. While current motion capture and kinematic reconstruction techniques were unable to create perfectly accurate reconstructions of machine motion, these analyses were capable of rendering course resolution results, suitable for visualization and data correlation. Future work in motion capture is likely to be capable of recording paths at substantially higher accuracy and could allow for machine motion analysis against planned paths.

The combination of these various DT techniques across two labs demonstrates the utility of a DT of manufacturing in two different use cases. This document has described the creation and use of a DT across these two use cases, and has demonstrated various DT techniques in both settings. Design, programming, commissioning, and system analysis are all improved by the introduction of a DT model, and we expect future developments in DT techniques to provide additional improvements in the manufacturing space. The DT is a powerful tool in the engineer's toolbox, and will likely continue to see use in the manufacturing field.

5.4 SITUATION OF RESEARCH

This work represents a small portion of the automation and manufacturing research performed by the neXt team at the University of South Carolina's McNair Center. Various portions of this work tie in, either directly or indirectly with other research projects and technologies being developed by the team. Much of this research is based in the field of advanced manufacturing and draws inspiration from Harik and Wuest's *Introduction to Advanced Manufacturing* [56]. The FF DT model developed in this research saw direct application in multiple other projects, including Xia et al.'s *A Digital Twin to Train Deep Reinforcement Learning Agent for Smart Manufacturing Plants*, Xia et al.'s *Towards Semantic Integration of Machine Vision Systems to Aid Manufacturing Event Understanding*, and Saidy et al.'s *Building future factories: a smart robotic assembly platform using virtual commissioning, data analytics, and accelerated computing* [30] [57] [58]. The AFP DT model and ACSIS and AFP programming techniques developed in this research has contributed to the development and implementation of new techniques for AFP, including Sacco et al.'s *Machine learning in composites manufacturing: A case study of automated fiber placement inspection* and Sacco et al.'s *On the effect of manual rework in AFP quality control for a doubly-curved part* [59] [60]. Additionally the AFP DT demonstrated in this thesis has been referenced in Brasington et al.'s review: *Automated fiber placement: A review of history, current technologies, and future paths forward* [61]. These works, and many others authored by the neXt team, build on each other to constructively create an ecosystem of research and development to push our understanding of advanced manufacturing further.

REFERENCES

[1] Q. Qi, F. Tao, T. Hu, N. Anwer, A. Liu, Y. Wei, L. Wang and A. Y. C. Nee, "Enabling technologies and tools for digital twin," *Journal of Manufacturing Systems,* vol. 58, p. 3–21, 2021.

[2] D. Jones, C. Snider, A. Nassehi, J. Yon and B. Hicks, "Characterising the Digital Twin: A systematic literature review," *CIRP Journal of Manufacturing Science and Technology,* vol. 29, p. 36–52, 2020.

[3] I. E. Sutherland, "Sketchpad a man-machine graphical communication system," *Simulation,* vol. 2, p. R–3, 1964.

[4] M. Grieves, "Digital Twin: Manufacturing Excellence through Virtual Factory Replication," March 2015.

[5] M. Shafto, M. Conroy, R. Doyle, E. Glaessgen, C. Kemp, J. LeMoigne and L. Wang, "Modeling, simulation, information technology & processing roadmap," *National Aeronautics and Space Administration,* vol. 32, p. 1–38, 2012.

[6] W. Kritzinger, M. Karner, G. Traar, J. Henjes and W. Sihn, "Digital Twin in manufacturing: A categorical literature review and classification," *IFAC-PapersOnLine,* vol. 51, p. 1016–1022, 2018.

[7] R. Rosen, G. Von Wichert, G. Lo and K. D. Bettenhausen, "About the importance of autonomy and digital twins for the future of manufacturing," *IFAC-PapersOnLine,* vol. 48, p. 567–572, 2015.

[8] K. Y. H. Lim, P. Zheng and C.-H. Chen, "A state-of-the-art survey of Digital Twin: techniques, engineering product lifecycle management and business innovation perspectives," *Journal of Intelligent Manufacturing,* p. 1–25, 2019.

[9] Q. Qi and F. Tao, "Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison," *Ieee Access,* vol. 6, p. 3585–3593, 2018.

[10] S. Wang, J. Wan, D. Li and C. Zhang, "Implementing smart factory of industrie 4.0: an outlook," *International journal of distributed sensor networks,* vol. 12, p. 3159805, 2016.

[11] E. Negri, L. Fumagalli and M. Macchi, "A review of the roles of digital twin in CPS-based production systems," *Procedia Manufacturing,* vol. 11, p. 939–948, 2017.

[12] E. Hozdić, "Smart factory for industry 4.0: A review," *International Journal of Modern Manufacturing Technologies,* vol. 7, p. 28–35, 2015.

[13] L. Wang, M. Törngren and M. Onori, "Current status and advancement of cyber-physical systems in manufacturing," *Journal of Manufacturing Systems,* vol. 37, p. 517–527, 2015.

[14] J. Lee, B. Bagheri and H.-A. Kao, "A cyber-physical systems architecture for industry 4.0-based manufacturing systems," *Manufacturing letters,* vol. 3, p. 18–23, 2015.

[15] R. K. Phanden, P. Sharma and A. Dubey, "A review on simulation in digital twin for aerospace, manufacturing and robotics," *Materials today: proceedings,* vol. 38, p. 174–178, 2021.

[16] C. G. Lee and S. C. Park, "Survey on the virtual commissioning of manufacturing systems," *Journal of Computational Design and Engineering,* vol. 1, p. 213–222, 2014.

[17] C. B. Anderton, "Challenges and Benefits of Implementing a Digital Twin in Composites Manufacturing".

[18] W. Liu, Y. Zeng, M. Maletz and D. Brisson, "Product Lifecycle Management: A Survey," *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference,* pp. 1213-1225, 2009.

[19] M. P. Giddaluru and J. Gao, "A data preparation and migration framework for implementing modular product structures in PLM," *IFIP International Conference on Product Lifecycle Management,* pp. 201-210, 2019.

[20] M. Demartini, F. Galluccio, P. Mattis, I. Abusohyon, R. Lepratti and F. Tonelli, "Closed-Loop Manufacturing for Aerospace Industry: An Integrated PLM-MOM

Solution to Support the Wing Box Assembly Process," *IFIP International Conference on Advances in Production Management,* pp. 423-430, 2019.

[21] J. Leng, H. Zhang, D. Yan, Q. Liu, X. Chen and D. Zhang, "Digital twin-driven manufacturing cyber-physical system for parallel controlling of smart workshop," *Journal of ambient intelligence and humanized computing,* vol. 10, p. 1155–1166, 2019.

[22] J. Long, J. Mou, L. Zhang, S. Zhang and C. Li, "Attitude data-based deep hybrid learning architecture for intelligent fault diagnosis of multi-joint industrial robots," *Journal of Manufacturing Systems,* vol. 61, p. 736–745, 2021.

[23] L. V. Guerrero, V. V. López and J. E. Mejía, "Virtual commissioning with process simulation (Tecnomatix)," *Computer-Aided Design and Applications,* vol. 11, p. S11–S19, 2014.

[24] J. Guo, N. Zhao, L. Sun and S. Zhang, "Modular based flexible digital twin for factory design," *Journal of Ambient Intelligence and Humanized Computing,* vol. 10, p. 1189–1200, 2019.

[25] Z. Samadikhoshkho, S. Ghorbani and F. Janabi-Sharifi, "Modeling and control of aerial continuum manipulation systems: A flying continuum robot paradigm," *IEEE Access,* vol. 8, p. 176883–176894, 2020.

[26] K. T. Park, Y. W. Nam, H. S. Lee, S. J. Im, S. D. Noh, J. Y. Son and H. Kim, "Design and implementation of a digital twin application for a connected micro smart factory," *International Journal of Computer Integrated Manufacturing,* vol. 32, p. 596–614, 2019.

[27] V. Kuts, G. E. Modoni, W. Terkaj, T. Tähemaa, M. Sacco and T. Otto, "Exploiting factory telemetry to support virtual reality simulation in robotics cell," in *International Conference on Augmented Reality, Virtual Reality and Computer Graphics*, 2017.

[28] A. Kusiak and A. Verma, "Analyzing bearing faults in wind turbines: A data-mining approach," *Renewable Energy,* vol. 48, p. 110–116, 2012.

[29] M. Phillips and M. Likhachev, "Sipp: Safe interval path planning for dynamic environments," in *2011 IEEE International Conference on Robotics and Automation*, 2011.

[30] K. Xia, C. Sacco, M. Kirkpatrick, C. Saidy, L. Nguyen, A. Kircaliali and R. Harik, "A digital twin to train deep reinforcement learning agent for smart manufacturing plants: Environment, interfaces and intelligence," *Journal of Manufacturing Systems,* vol. 58, p. 210–230, 2021.

[31] L. Hu, Z. Liu, W. Hu, Y. Wang, J. Tan and F. Wu, "Petri-net-based dynamic scheduling of flexible manufacturing system via deep reinforcement learning with graph convolutional network," *Journal of Manufacturing Systems,* vol. 55, p. 1–14, 2020.

[32] J. Oyekan, M. Farnsworth, W. Hutabarat, D. Miller and A. Tiwari, "Applying a 6 DoF Robotic Arm and Digital Twin to Automate Fan-Blade Reconditioning for Aerospace Maintenance, Repair, and Overhaul," *Sensors,* vol. 20, p. 4637, 2020.

[33] D. Hughes, S. Keir and C. Meggs, "Digital twin methodology for compression moulded thermoplastic composite optimisation," *Flow Processes in Composite Materials (FPCM),* 2018.

[34] G. L. Knapp, T. Mukherjee, J. S. Zuback, H. L. Wei, T. A. Palmer, A. De and T. J. A. M. DebRoy, "Building blocks for a digital twin of additive manufacturing," *Acta Materialia,* vol. 135, p. 390–399, 2017.

[35] C. Sacco, A. B. Radwan, T. Beatty and R. Harik, "Machine learning based AFP inspection: A tool for characterization and integration," in *SAMPE 2019 Conference & Exhibition. Charlotte, NC: SAMPE. doi*, 2019.

[36] Z. Liu, N. Suchold and C. Diedrich, "Virtual commissioning of automated systems," in *Automation*, IntechOpen, 2012.

[37] E. Yildiz, C. Møller and A. Bilberg, "Virtual factory: digital twin based integrated factory simulations," *Procedia CIRP,* vol. 93, p. 216–221, 2020.

[38] D. Zuehlke, "SmartFactory—Towards a factory-of-things," *Annual reviews in control,* vol. 34, p. 129–138, 2010.

[39] K. Lerman, C. Jones, A. Galstyan and M. J. Matarić, "Analysis of dynamic task allocation in multi-robot systems," *The International Journal of Robotics Research,* vol. 25, p. 225–241, 2006.

[40] S. Stieber, A. Hoffmann, A. Schiendorfer, W. Reif, M. Beyrle, J. Faber, M. Richter and M. Sause, "Towards real-time process monitoring and machine learning for

manufacturing composite structures," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2020.

[41] C. Liu, H. Vengayil, Y. Lu and X. Xu, "A cyber-physical machine tools platform using OPC UA and MTConnect," *Journal of Manufacturing Systems,* vol. 51, p. 61–74, 2019.

[42] F. Tao, Y. Zuo, L. Da Xu and L. Zhang, "IoT-based intelligent perception and access of manufacturing resource toward cloud manufacturing," *IEEE Transactions on Industrial Informatics,* vol. 10, p. 1547–1557, 2014.

[43] C. Zhang, G. Zhou, J. He, Z. Li and W. Cheng, "A data-and knowledge-driven framework for digital twin manufacturing cell," *Procedia CIRP,* vol. 83, p. 345–350, 2019.

[44] H. Bayram and H. I. Bozma, "Coalition formation games for dynamic multirobot tasks," *The International Journal of Robotics Research,* vol. 35, p. 514–527, 2016.

[45] S. Pellegrinelli, A. Orlandini, N. Pedrocchi, A. Umbrico and T. Tolio, "Motion planning and scheduling for human and industrial-robot collaboration," *CIRP Annals,* vol. 66, p. 1–4, 2017.

[46] Z. Liu, W. Chen, C. Zhang, C. Yang and Q. Cheng, "Intelligent scheduling of a feature-process-machine tool supernetwork based on digital twin workshop," *Journal of manufacturing systems,* vol. 58, p. 157–167, 2021.

[47] W. Xu, J. Cui, L. Li, B. Yao, S. Tian and Z. Zhou, "Digital twin-based industrial cloud robotics: framework, control approach and implementation," *Journal of Manufacturing Systems,* vol. 58, p. 196–209, 2021.

[48] S. Zambal, C. Eitzinger, M. Clarke, J. Klintworth and P.-Y. Mechin, "A digital twin for composite parts manufacturing: Effects of defects analysis based on manufacturing data," in *2018 IEEE 16th International conference on industrial informatics (INDIN)*, 2018.

[49] S. Haag and R. Anderl, "Digital twin–Proof of concept," *Manufacturing Letters,* vol. 15, p. 64–66, 2018.

[50] T. Chettibi, H. E. Lehtihet, M. Haddad and S. Hanchi, "Minimum cost trajectory planning for industrial robots," *European Journal of Mechanics-A/Solids,* vol. 23, p. 703–715, 2004.

[51] C. Zhang, W. Xu, J. Liu, Z. Liu, Z. Zhou and D. T. Pham, "A reconfigurable modeling approach for digital twin-based manufacturing system," *Procedia Cirp,* vol. 83, p. 118–125, 2019.

[52] B. A. Talkhestani, N. Jazdi, W. Schlögl and M. Weyrich, "Consistency check to synchronize the Digital Twin of manufacturing automation based on anchor points," *Procedia Cirp,* vol. 72, p. 159–164, 2018.

[53] A. Haleem, M. Javaid, R. P. Singh, S. Rab, R. Suman, L. Kumar and I. H. Khan, "Exploring the potential of 3D scanning in Industry 4.0: An overview," *International Journal of Cognitive Computing in Engineering,* 2022.

[54] P. K. Malik, R. Sharma, R. Singh, A. Gehlot, S. C. Satapathy, W. S. Alnumay, D. Pelusi, U. Ghosh and J. Nayak, "Industrial Internet of Things and its Applications in Industry 4.0: State of the Art," *Computer Communications,* pp. 125-139, 2021.

[55] A. Brasington, *Integration of Process Planning Into the Automated Fiber Placement Design for Manufacturing Cycle,* 2021.

[56] R. Harik and T. Wuest, Introduction to Advanced Manufacturing, SAE, 2020.

[57] K. Xia, C. Saidy, M. Kirkpatrick, N. Anumbe, A. Sheth and R. Harik, "Towards Semantic Integration of Machine Vision Systems to Aid Manufacturing Event Understanding," *Sensors,* vol. 21, no. 13, p. 4276, 2021.

[58] C. Saidy, K. Xia, C. Sacco, M. Kirkpatrick, A. Kircaliali, L. Nguyen and R. Harik, "Building future factories: a smart robotic assembly platform using virtual commissioning, data analytics, and accelerated computing," in *SAMPE Conference, Virtual Presentation Series*, Amsterdam, The Netherlands, 2020.

[59] C. Sacco, A. B. Radwan, A. Anderson, R. Harik and E. Gregory, "Machine learning in composites manufacturing: A case study of automated fiber placement inspection," *Composite Structures,* vol. 250, p. 112514, 2020.

[60] C. Sacco, A. Brasington, C. Saidy, M. Kirkpatrick, J. Halbritter, R. Wehbe and R. Harik, "On the effect of manual rework in AFP quality control for a doubly-curved part," *Composites Part B: Engineering,* vol. 227, p. 104932, 2021.

[61] A. Brasington, C. Sacco, J. Halbritter, R. Wehbe and R. Harik, "Automated fiber placement: A review of history, current technologies, and future paths forward," *Composites Part C: Open Access,* vol. 6, p. 100182, 2021.

APPENDIX A: AFP MOTION RECONSTRUCTION COMPARISON FIGURES AND VECTOR FIELDS

This appendix contains images of AFP kinematic motion reconstruction results for all plies, as well as their corresponding error vector fields.

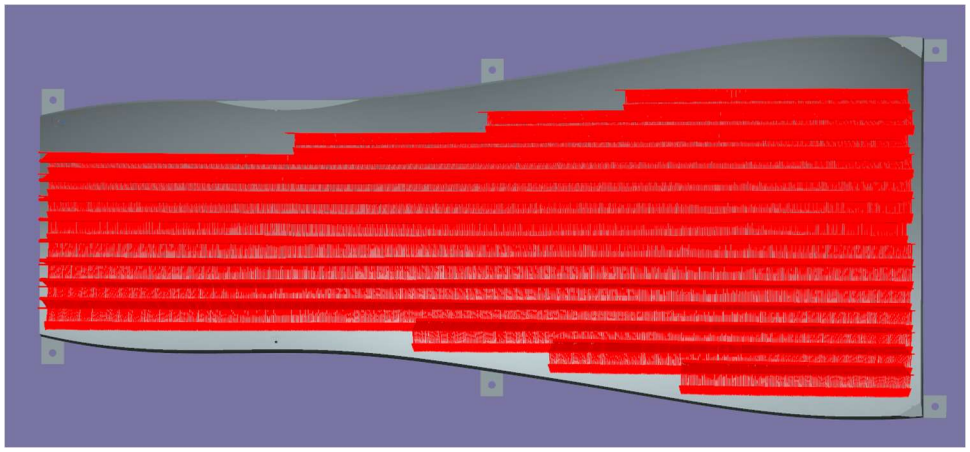

Figure A.1. Motion Reconstruction for Ply 0101

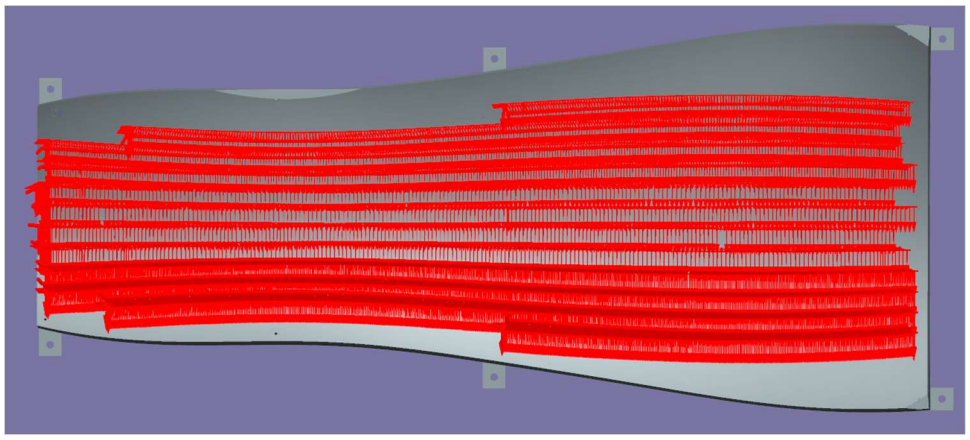

Figure A.2. Motion Reconstruction for Ply 0102

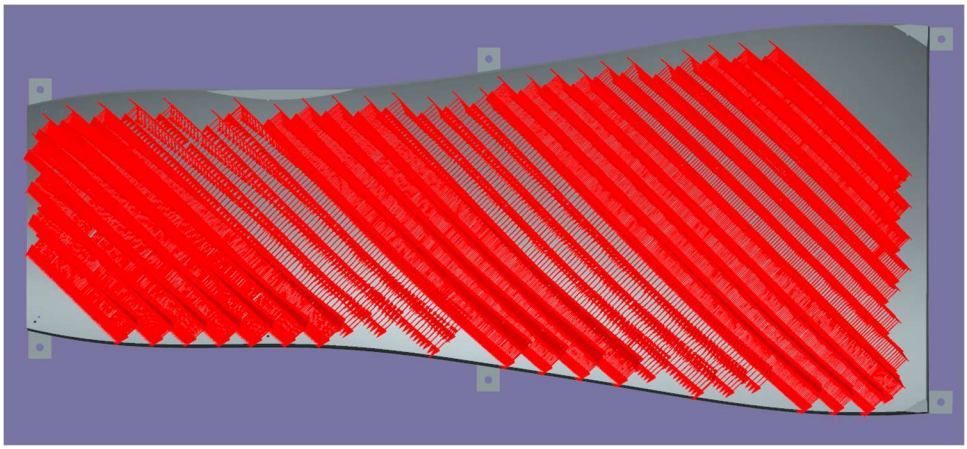

Figure A.3. Motion Reconstruction for Ply 0105

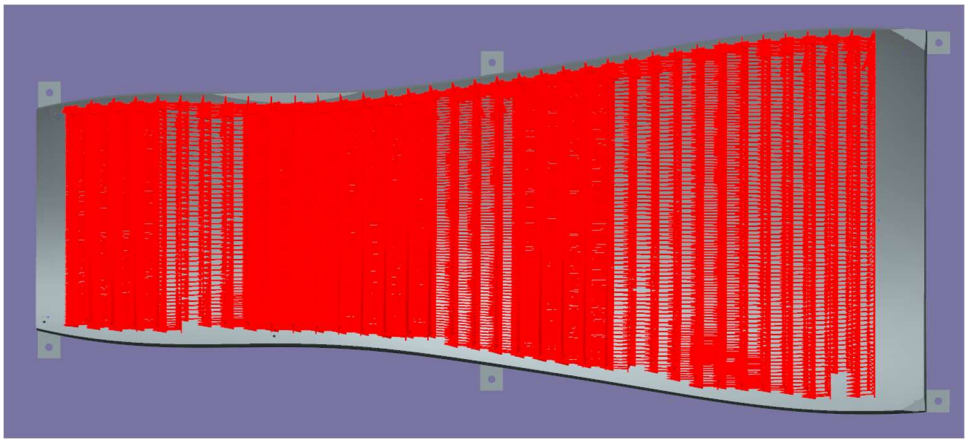

Figure A.4. Motion Reconstruction for Ply 0112

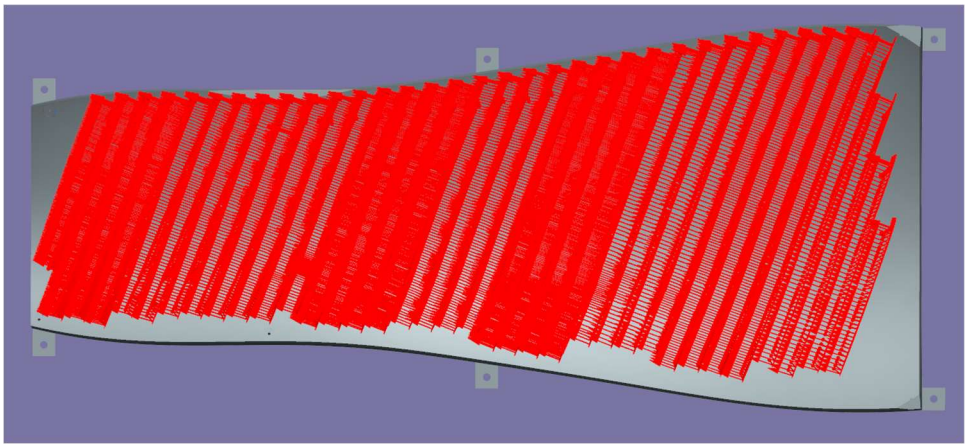

Figure A.5. Motion Reconstruction for Ply 0117

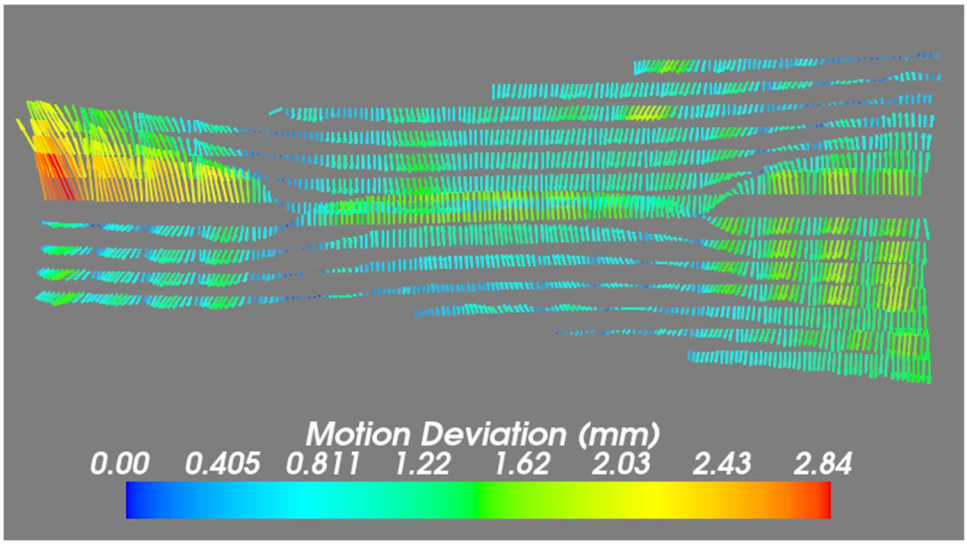

Figure A.6. Motion Comparison Error Vector Field for Ply 0101 with an Average Deviation of 0.99mm

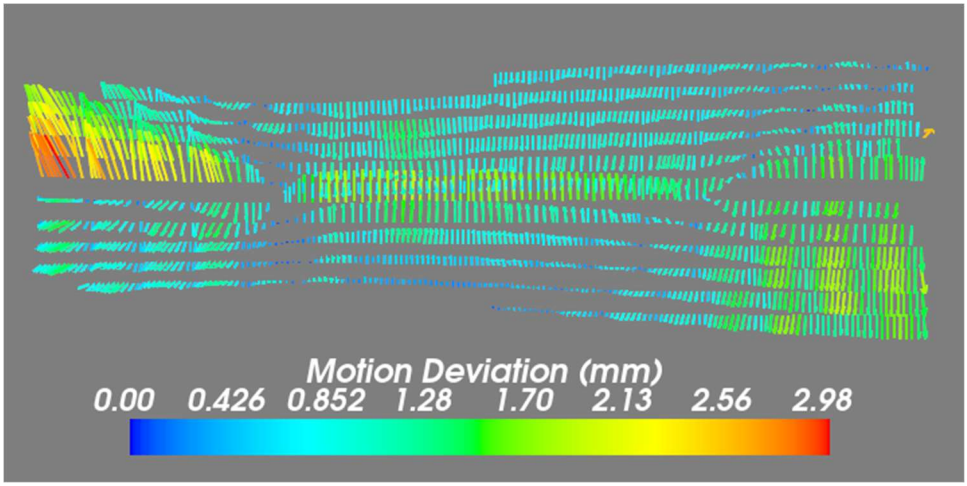

Figure A.7. Motion Comparison Error Vector Field for Ply 0102 with an Average Deviation of 0.92mm

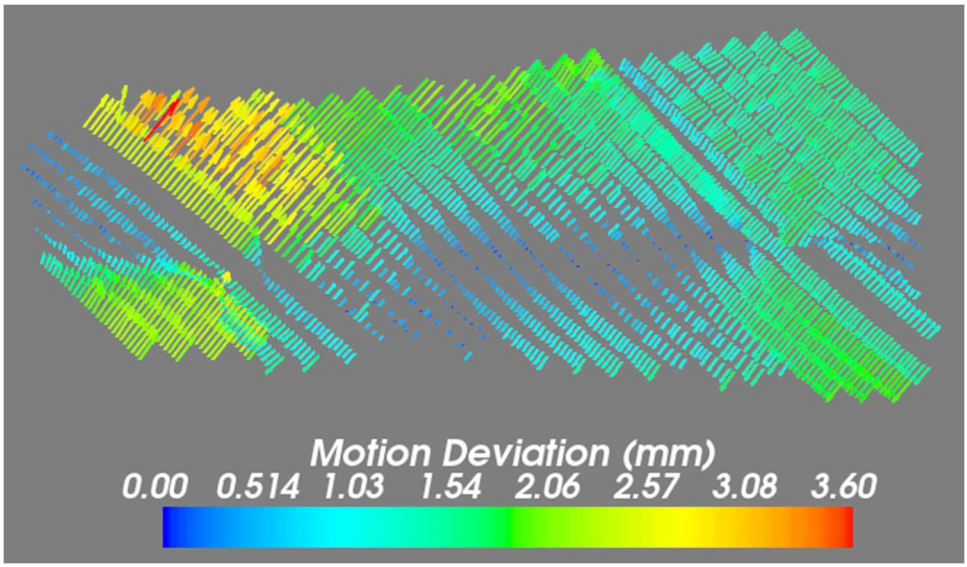

Figure A.8. Motion Comparison Error Vector Field for Ply 0105 with an Average Deviation of 1.26mm

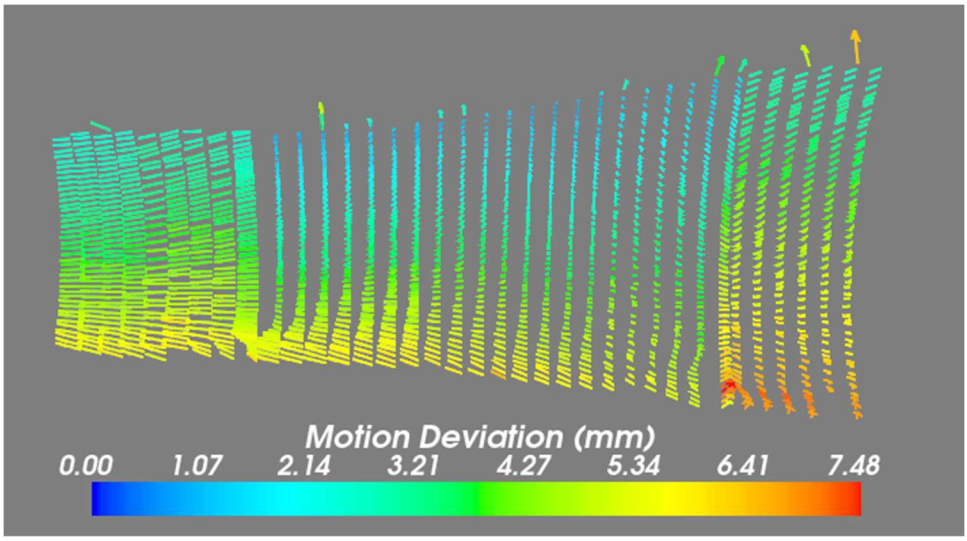

Figure A.9. Motion Comparison Error Vector Field for Ply 0112 with an Average Deviation of 3.53mm

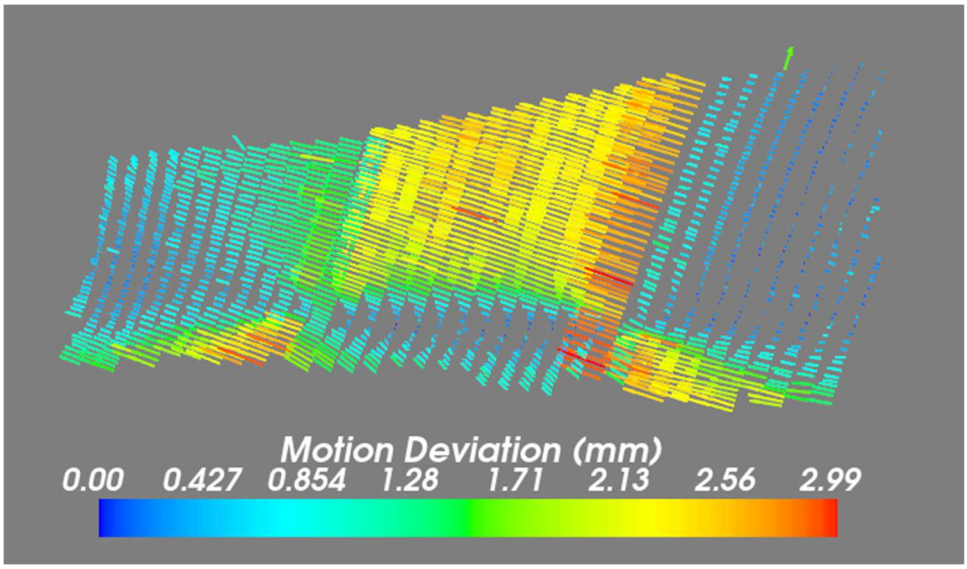

Figure A.10. Motion Comparison Error Vector Field for Ply 0117 with an Average Deviation of 1.23mm