Energy Digital Twins in Smart Manufacturing Systems

Anna Billey
West Virginia University, ab00092@mix.wvu.edu

Follow this and additional works at: https://researchrepository.wvu.edu/etd

Part of the Industrial Engineering Commons

Recommended Citation
https://researchrepository.wvu.edu/etd/11859

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.
Energy Digital Twins in Smart Manufacturing Systems

Anna Billey

Thesis submitted
to the Benjamin M. Statler College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements for the degree of
Master of Science in Industrial and Management Systems Engineering

Thorsten Wuest, Ph.D., Chair
Zhichao Liu, Ph.D.
Yuhe Tian, Ph.D.
Department of Industrial and Management Systems Engineering

Morgantown, West Virginia
2023

Keywords: Smart Manufacturing; Digital Twin; Industry 4.0; Energy Efficiency; Energy Optimization; Sustainability

Copyright 2023 Anna Billey
Abstract

Energy Digital Twins in Smart Manufacturing Systems

Anna Billey

In this thesis, an Energy Digital Twin for smart manufacturing systems was developed and evaluated. In particular, the study focused on bidirectional parameter communication between the physical and the virtual part with the aim of optimizing the energy used in the manufacturing process. Rising costs and the environmental impacts related to energy consumption have grown in importance worldwide. There are elevated concerns in sectors like manufacturing, leading to an urgent quest to reduce energy consumption. A recent advancement in Industry 4.0 technology, the Digital Twin, represents a promising smart technology and tool that researchers are investigating to help reduce energy costs at the shopfloor level by analyzing and optimizing energy consumption. Energy Digital Twins are a relatively new research field, only gaining popularity in applications and academia within the last five years. As a new area of interest, the state of the art of Energy Digital Twins in smart manufacturing was studied through a comprehensive literature review to lay the foundation for this study. The review uncovered that there is only one ‘true’ Energy Digital Twin application in published research based on the search criteria used. To address this research gap, the aim of this thesis is to create and evaluate an application of an energy optimizing Digital Twin of the Festo CP Lab Heating Tunnel station. Following the definition of a Digital Twin, the research methodology and experimental setup have three major components: i) the Heating Tunnel as the physical object, ii) the digital counterpart constructed using Python to house the digital control logic and linear energy optimization feedback model, and iii) the connecting fabric, in form of a bidirectional OPC UA communication protocol. The optimization model ingests input parameters of setpoint temperature, power level, and targeted overshoot time, and after running the simulation, returns a calculated value of the required turn off temperature to the real-time heating process of the physical system. The proposed methodology was validated by performing an array of trial runs varying the input parameters. Results show that the Energy Digital Twin is effective at maintaining the maximum temperature range of the Heating Tunnel during the heating process, in addition to reducing the energy consumption and cost for all trial runs compared to the original process. Overall, the research completed in this thesis successfully achieved to create a functioning energy optimizing Digital Twin with bidirectional, automated feedback. The results of this research emphasize the potential impact of Energy Digital Twin applications in any manufacturing process and show the promise of future work in this realm of Energy Digital Twins.
Acknowledgements

I would like to first and foremost thank my family and friends for their endless support not only through my graduate career, but throughout my entire life. Mom and Dad, you both have encouraged and pushed me to be the best and most successful woman I can be both in and out of the classroom. I would not be who I am today or have achieved the accomplishments thus far without you both. Additionally, I would also like to thank my brother for always supporting me and being there for when I need someone. Ally and Makenzie, I want to say a special thanks to both of you for being the best friends anyone could ask for. You have both been there through the good and the bad times, and I am ever so grateful I have you both to lean on for the rest of my life.

A significant amount of appreciation and gratitude is given to Dr. Thorsten Wuest for being a phenomenal research advisor and mentor. I would like to thank you for taking the chance and accepting me as a transfer graduate student in your Smart Manufacturing Lab as a research assistant. Without your continual guidance and belief in me as a student throughout my graduate career, this degree would not be possible. Even though there were some tough times along the way, you have truly made this an outstanding and valuable graduate experience.

Lastly, I would like to thank the members of the Smart Manufacturing Lab for their willingness to help and constant support the last two years. Not only were friendships formed, but the assistance in troubleshooting and problem solving throughout the research was a key in completing this thesis.
# Table of Contents

Title Page .................................................................................................................................................. i  
Abstract .................................................................................................................................................... ii  
Acknowledgements ................................................................................................................................... iii  
List of Figures ........................................................................................................................................ vi  
List of Tables ........................................................................................................................................... viii  
List of Abbreviations ............................................................................................................................... ix  
1. Introduction and Problem Statement ................................................................................................. 1  
2. Literature Review ................................................................................................................................. 4  
  2.1 Energy Digital Twins ......................................................................................................................... 4  
  2.2 Type of digital tech.: Digital Model, Shadow, or Twin .................................................................... 5  
  2.3 Communication Protocols ............................................................................................................. 5  
  2.4 Energy Consumption Applications ................................................................................................ 5  
  2.5 Energy Meter .................................................................................................................................... 6  
  2.6 Research Gaps .................................................................................................................................. 6  
3. Research Objective ................................................................................................................................. 8  
4. Research Plan and Methodology .......................................................................................................... 10  
  4.1 Physical Space ................................................................................................................................. 10  
    4.1.1 Project SMS Testbed: Festo CP Lab ......................................................................................... 10  
    4.1.2 CP Lab Heating Tunnel .......................................................................................................... 13  
    4.1.2 CP Lab Heating Tunnel Limitations and Solution Approach ................................................ 15  
    4.1.3 Festo Energy Measurement Box ............................................................................................. 19  
  4.2 Connection: OPC UA ...................................................................................................................... 19  
  4.3 Digital Space .................................................................................................................................... 20  
    4.3.1 Monitoring Parameters ............................................................................................................ 21  
    4.3.2 Energy Optimization .............................................................................................................. 22  
    4.3.3 Digital Twin ............................................................................................................................ 32  
5. Data Processing and Validation ......................................................................................................... 35  
  5.1.1 Physical Space Setup ................................................................................................................. 36  
  5.1.2 Connection: OPC UA .............................................................................................................. 36  
  5.1.3 Digital Space Setup ................................................................................................................... 37  
6. Results and Discussion ......................................................................................................................... 39
6.1 Energy DT and Optimization Testing Results ........................................................................39
  6.1.1 Static DT Simulation Results ..................................................................................39
  6.1.2 Base Run Results ...............................................................................................40
  6.1.3 Validation Runs ...................................................................................................48
  6.1.4 Energy Optimization Discussion .........................................................................64
6.2 Parameter Monitoring ...............................................................................................69
6.3 General Discussion ....................................................................................................70
7. Conclusion ....................................................................................................................73
8. References ..................................................................................................................74
9. Appendix A ..................................................................................................................76
List of Figures

Figure 1: Digital Model data flow (based on [9]) ................................................................. 2
Figure 2: Digital Shadow data flow (based on [9]) ............................................................... 3
Figure 3: Digital Twin data flow (based on [9]) ................................................................. 3
Figure 4: Energy Applications (Based on graphic from [15]) ............................................. 6
Figure 5: Methodology Structure based on diagrams established in [16] ......................... 10
Figure 6: WVU SM CP Lab .................................................................................................. 11
Figure 7: CP Lab Part Housing ......................................................................................... 12
Figure 8: Pallet .................................................................................................................. 12
Figure 9: HMI .................................................................................................................... 13
Figure 10: Festo MES Efficiency Report Screen ............................................................... 13
Figure 11: a) Heating Tunnel Power ON, b) Entrance of Heating Tunnel, c) Front of Heating Tunnel Station ............................................................. 14
Figure 12: Heating Tunnel Operational Sequence ........................................................... 15
Figure 13: Flowchart for System Control State Logic ....................................................... 18
Figure 14: Festo Energy Measurement Box (EMB) ........................................................... 19
Figure 15: Connection Overview ...................................................................................... 20
Figure 16: Ignition Data Highway (based on [21]) ............................................................. 22
Figure 17: Example Ignition Dashboard (from [21]) ......................................................... 22
Figure 18: Temperature of Heating Tunnel at 500W and 1000W vs Time ......................... 23
Figure 19: 500W Piecewise Functions Graphed ............................................................... 25
Figure 20: Proportional Triangles for 500W .................................................................. 26
Figure 21: Reduced Temperature Overshoot at 500W .................................................... 27
Figure 22: 1000W Piecewise Functions Graphed ............................................................. 29
Figure 23: Proportional Triangles for 1000W ................................................................. 30
Figure 24: Reduced Temperature Overshoot at 1000W .................................................. 31
Figure 25: Network Connection Overview .................................................................... 37
Figure 26: Flowchart of Two Python Interpreters ............................................................ 38
Figure 27: Static Energy DT Output Data Example ......................................................... 40
Figure 28: Static Energy DT Output Graph Example ....................................................... 40
Figure 29: 500W Initial Base Run Temperature Data and Simulation Results Graph ....... 42
Figure 30: 1000W Initial Base Run Temperature Data and Simulation Results ............ 43
Figure 31: 500W Second Base Run Temperature Data and Simulation Results Graph .... 45
Figure 32: 1000W Second Base Run Temperature Data and Simulation Results .......... 46
Figure 33: 500W Setpoint: 45°C, Overshoot: 10s Graph ............................................... 48
Figure 34: 500W Setpoint: 50°C, Overshoot: 10s Graph ............................................... 49
Figure 35: 1000W Setpoint: 45°C, Overshoot: 10s Graph ............................................... 50
Figure 36: 1000W Setpoint: 50°C, Overshoot: 10s Graph ............................................... 51
Figure 37: 500W Setpoint: 40°C, Overshoot: 15s Graph ............................................... 52
Figure 38: 500W Setpoint: 45°C, Overshoot: 15s Graph ............................................... 53
Figure 39: 500W Setpoint: 50°C, Overshoot: 15s Graph ............................................... 54
Figure 40: 1000W Setpoint: 40°C, Overshoot: 15s Graph ............................................... 55
Figure 41: 1000W Setpoint: 45°C, Overshoot: 15s Graph ............................................... 56
List of Tables

Table 1: Parameters of Interest ............................................................................................................... 21
Table 2: Base Runs (parameter setting) ................................................................................................. 35
Table 3: Validation Runs (parameter setting) ......................................................................................... 35
Table 4: 500W Initial Base Run Results ............................................................................................... 42
Table 5: 1000W Initial Base Run Results .............................................................................................. 43
Table 6: 500W Second Base Run Results ............................................................................................ 45
Table 7: 1000W Second Base Run Results .......................................................................................... 46
Table 8: 500W Setpoint: 45°C, Overshoot: 10s Results .......................................................................... 48
Table 9: 500W Setpoint: 50°C, Overshoot: 10s Results ......................................................................... 49
Table 10: 1000W Setpoint: 45°C, Overshoot: 10s Results ..................................................................... 50
Table 11: 1000W Setpoint: 50°C, Overshoot: 10s Results ..................................................................... 51
Table 12: 500W Setpoint: 40°C, Overshoot: 15s Results ....................................................................... 52
Table 13: 500W Setpoint: 45°C, Overshoot: 15s Results ....................................................................... 53
Table 14: 500W Setpoint: 50°C, Overshoot: 15s Results ....................................................................... 54
Table 15: 1000W Setpoint: 40°C, Overshoot: 15s Results ..................................................................... 55
Table 16: 1000W Setpoint: 45°C, Overshoot: 15s Results ..................................................................... 56
Table 17: 1000W Setpoint: 50°C, Overshoot: 15s Results ..................................................................... 57
Table 18: 500W Setpoint: 40°C, Overshoot: 20s Results ....................................................................... 58
Table 19: 500W Setpoint: 45°C, Overshoot: 20s Results ....................................................................... 59
Table 20: 500W Setpoint: 50°C, Overshoot: 20s Results ....................................................................... 60
Table 21: 1000W Setpoint: 40°C, Overshoot: 20s Results ..................................................................... 61
Table 22: 1000W Setpoint: 45°C, Overshoot: 20s Results ..................................................................... 62
Table 23: 1000W Setpoint: 50°C, Overshoot: 20s Results ..................................................................... 63
## List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>°C</td>
<td>Celsius</td>
</tr>
<tr>
<td>A</td>
<td>Amperes</td>
</tr>
<tr>
<td>ABET</td>
<td>Accreditation Board for Engineering and Technology</td>
</tr>
<tr>
<td>AI</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>CP</td>
<td>Cyber-Physical</td>
</tr>
<tr>
<td>CPS</td>
<td>Cyber-Physical Systems</td>
</tr>
<tr>
<td>DM</td>
<td>Digital Model</td>
</tr>
<tr>
<td>DS</td>
<td>Digital Shadow</td>
</tr>
<tr>
<td>DT</td>
<td>Digital Twin</td>
</tr>
<tr>
<td>EC</td>
<td>Energy Consumption</td>
</tr>
<tr>
<td>EIA</td>
<td>Energy Information Administration</td>
</tr>
<tr>
<td>EMB</td>
<td>Energy Measurement Box</td>
</tr>
<tr>
<td>EU</td>
<td>European Union</td>
</tr>
<tr>
<td>HMI</td>
<td>Human-Machine Interface</td>
</tr>
<tr>
<td>IIoT</td>
<td>Industrial Internet of Things</td>
</tr>
<tr>
<td>kWh</td>
<td>kilowatt-hours</td>
</tr>
<tr>
<td>L/min</td>
<td>Liter/minute</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>MES</td>
<td>Manufacturing Execution System</td>
</tr>
<tr>
<td>min</td>
<td>Minutes</td>
</tr>
<tr>
<td>ML</td>
<td>Machine Learning</td>
</tr>
<tr>
<td>NAMRC</td>
<td>North American Manufacturing Research Conference</td>
</tr>
<tr>
<td>NASA</td>
<td>National Aeronautics and Space Administration</td>
</tr>
<tr>
<td>OEE</td>
<td>Overall Equipment Effectiveness</td>
</tr>
<tr>
<td>OPC UA</td>
<td>Open Platform Communication - Unified Architecture</td>
</tr>
<tr>
<td>PLC</td>
<td>Programmable Logic Controller</td>
</tr>
<tr>
<td>PLM</td>
<td>Product Lifecycle Management</td>
</tr>
<tr>
<td>PPC</td>
<td>Production and Planning Control</td>
</tr>
<tr>
<td>RFID</td>
<td>Radiofrequency Identification</td>
</tr>
<tr>
<td>s</td>
<td>Seconds</td>
</tr>
<tr>
<td>SCADA</td>
<td>Supervisory Control and Data Acquisition</td>
</tr>
<tr>
<td>SM</td>
<td>Smart Manufacturing</td>
</tr>
<tr>
<td>SMS</td>
<td>Smart Manufacturing System</td>
</tr>
<tr>
<td>SOAP</td>
<td>Simple Object Access Protocol</td>
</tr>
<tr>
<td>US</td>
<td>United States</td>
</tr>
<tr>
<td>USB</td>
<td>Universal Serial Bus</td>
</tr>
<tr>
<td>USD</td>
<td>US Dollars</td>
</tr>
<tr>
<td>V</td>
<td>Volts</td>
</tr>
<tr>
<td>VA</td>
<td>Volt-Amperes</td>
</tr>
<tr>
<td>VAR</td>
<td>Volt-Amperes Reactive</td>
</tr>
<tr>
<td>W</td>
<td>Watt</td>
</tr>
<tr>
<td>Wh</td>
<td>Watt-hour</td>
</tr>
<tr>
<td>Ws</td>
<td>Watt-second</td>
</tr>
<tr>
<td>WV</td>
<td>West Virginia</td>
</tr>
<tr>
<td>WVU</td>
<td>West Virginia University</td>
</tr>
</tbody>
</table>
1. Introduction and Problem Statement

Energy consumption has historically been a concern in the manufacturing sector. In recent years, energy costs have been hitting all-time highs directly stemming from the COVID-19 pandemic, international disputes, climate impact, and resource limitations. The concern for energy consumption is greater than ever. According to the United States (US) Energy Information Administration (EIA), out of the four major types of industries (Agriculture, Construction, Manufacturing, and Mining) that are included in the Annual Energy Outlook 2022, manufacturing accounts for 81% of the total industrial energy consumption [1]. The Short Term Energy Outlook forecast published by the EIA, electricity prices alone in the US industrial sector are to increase approximately 21.29% from US$0.067 in cents per kilowatt-hour in 2020 to projected US$0.0809 cents per kilowatt-hour in 2023 [2]. The US is not the only country on high alert for the rising costs in energy. The European Union (EU) is notably more impacted by the rise of energy costs, and in result is more concerned with their energy consumption rates in the manufacturing sector. The availability of energy sources has always been a concern for the EU, but recently has escalated to a top apprehension. Europe has seen a drastic increase in energy prices, reaching all-time highs in 2022 [3]. To put into perspective, Germans are paying approximately $0.474 per kilowatt hour, which is nearly six times as much per kilowatt hour than the American industrial consumer is paying. The high costs have both manufacturers and consumers seeking alternative methods to reduce their energy consumption, and in result save money.

Sustainability pushes have also been a growing trend in manufacturing sector throughout the world. The most common used sustainable development definition comes from the Brundtland Report where it is defined as a “development that meets the needs of the present without compromising the ability of future generations to meet their own needs” [4]. In more modern terminology, sustainability is described as “the integration of environmental health, social equity and economic vitality in order to create thriving, healthy, diverse and resilient communities for this generation and generations to come” [5]. With this ideology, companies are moving towards finding new ways to protect the environment with their practices, but still satisfy their customers. In manufacturing, sustainability practices include finding different ways to reduce energy consumption. With the reduction in energy usage, it will lead to a decrease the demand for energy, and therefore decreasing a need for energy sources that generate carbon emissions: coal and natural gas. The reduction of carbon emissions can help reach the goal of sustainability, and one of the ways this can be completed is through the implementation of digital technologies in Smart Manufacturing Systems (SMS).

Currently, the Fourth Industrial Revolution, or Industry 4.0, is moving towards integrating digital technologies into manufacturing systems to create process improvements. The US commonly refers to this term as ‘Smart Manufacturing’ (SM). SM is defined as “the information-driven, event-driven, efficient and collaborative orchestration of business, physical and digital processes

1 Information from this section is either adapted or copied from accepted NAMRC 51 paper “Energy Digital Twins in Smart Manufacturing Systems - A Literature Review” [11]
within plants, factories and across the entire value chain” [6]. The revolution is being spearheaded by the implementation of digital technologies into the manufacturing sector. Examples of the technology include Machine Learning (ML) and Artificial Intelligence (AI) techniques, Industrial Internet of Things (IIoT), Cyber-Physical Systems (CPS), data analytics, and Digital Twins (DTs) [7]. In this research, the DT is focused on helping decrease energy consumption, increase efficiency, and increase sustainability in the manufacturing domain.

DTs are a technology that have been growing over recent years with the rise of Industry 4.0 in manufacturing. Michael Grieves of the National Aeronautics and Space Administration (NASA) in 2003 was the first to use the term DT to describe a system that comprises a connected physical object and virtual replica. As technological advances have occurred throughout the years, Grieves has refined his DT concept to include a physical object in real space, a virtual object in virtual space, and connections of data and information that tie the physical and real objects together [8]. With this refinement, there is still gray area on the method of data connections and information flow that ties both objects together. Even though the concept of the DT exits, to this day, there is still not a generally accepted definition of a DT, meaning the definition is up to the interpretation of the author. To illustrate a full DT concept, the Grieves’ idea can be refined even further to identify the method of data flow in the DT. Hence, this research adopts the definition from Kritzinger et. al [9]. The authors describe a DT as a digital representation of a physical object that uses automated data flow in the directions of physical to virtual object and virtual to physical. Misunderstandings of a DT are very common in published research. There are three major levels of integration of digital technology in manufacturing established in the classification literature review conducted by Kritzinger et. al [9]: (1) Digital Model (DM), (2) Digital Shadow (DS), and (3) Digital Twin (DT).

A DM is described as a digital representation of a physical object that does not use automated data flow in either direction, from physical to digital or digital to physical, as illustrated in Figure 1. All data flow in this setup is manual [9].

![Figure 1: Digital Model data flow (based on [9])](image)

A DS is described as a step above a DM, with an automated data flow in the direction from the physical object to the digital environment. If a parameter changes in the physical object, there is a reflection in the digital environment [9], as displayed in Figure 2. However, the data flow from digital to physical is not automated in this variant.
The final phase of integration according to [9], a DT, features automatic data flow in both directions - digital to physical as well as physical to digital. A change in parameters at the physical object leads to a change in the digital object, as well as a change in state at the digital level is communicated automatically to the physical object [9]. Figure 3 shows the principal data flow of a DT.

DTs have multiple different purposes they can be utilized for in the manufacturing sector. Kritzinger et al. in their literature review published in 2017 uncovered several focus areas of DTs, including production and planning control (PPC), maintenance, product lifecycle management (PLM), general manufacturing, layout planning, and process design [9]. Yet, the beauty of DT applications in manufacturing is their ability to be implemented in a wide variety of domains with numerous different objectives relating to analysis or optimization.

With the increase in energy prices and sustainability pushes from companies, manufacturers are seeking different solutions to decrease their energy consumption, as well as help protect the environment. One solution to this problem is looking into integration of digital technologies, like DTs, to manufacturing processes to help improve energy efficiency. Utilizing Energy DTs to analyze energy consumption and optimize energy efficiency can help find the source of high energy consumers leading to process improvements and cost savings.
2. Literature Review

A total of two publications were published or accepted to the North American Manufacturing Research Conference (NAMRC). The first publication, “Status Quo of Smart Manufacturing Curricula Offered by ABET (Accreditation Board for Engineering and Technology) accredited Industrial Engineering Programs in the US,” was accepted to NAMRC 50 and published open-access in Manufacturing Letters in October 2022. This publication researched and investigated the status quo of Smart Manufacturing curricula offered by ABET accredited Industrial Engineering programs throughout the United States. The research results showed there is a growing interest in SM at the university level. The main topics of the curriculums offered include Data Driven Analytics and Basic SM knowledge including DTs and ML. But there is a lack there of in the Energy Analytics category. This observation helps support the need for further research in the energy realm, and even more so in the combination of energy and DT applications. For further information about the background, topics, and results established in this publication, refer to [10].

In order to uncover the research completed in the energy realm of DT applications in the manufacturing sector, a comprehensive literature review titled “Energy Digital Twins in Smart Manufacturing Systems - A Literature Review [11]” was conducted. One purpose of the review was to analyze the status quo of the energy applications in published research. Another purpose included discovering how many published papers close the DT loop with bidirectional feedback focused on an energy perspective and how they achieve the task from a technical view. According to the author’s search criteria, the review focused on the analysis of publications that included three subject matters: DTs, manufacturing, and energy consumption/optimization. The sections below go into further detail on the status of the state of the art, as well as discovering several research gaps. The paper is accepted to the NAMRC 51 conference and will be published open-access in Manufacturing Letters after the conclusion of the conference.

2.1 Energy Digital Twins

As discussed in the introduction, DTs can have multiple different purposes in manufacturing, including but not limited to, PPC, maintenance, PLM, general manufacturing, layout planning, and process design [9]. But the beauty of DT applications is their ability to be adaptable and implemented into a wide variety of domains relating to data analysis and optimization. With the rise of the energy crises worldwide and increase of sustainability practices companies are attempting to implement into their processes, energy related DTs are starting to become more popular. Energy DTs are a specific type of DT that analyze and/or optimize energy consumption to increase energy efficiency in a manufacturing perspective.

---

2 Results from this section are either adapted or copied from NAMRC 51 accepted paper “Energy Digital Twins in Smart Manufacturing Systems - A Literature Review” [11]
2.2 Type of digital tech.: Digital Model, Shadow, or Twin

Mentioned in the introduction, the authors adopted the definition of a DT from [9] that features automatic data flow in both directions - digital to physical as well as physical to digital. From the literature review, it was discovered that there was only one paper by Park et. al [12] that implemented a ‘true’ DT that closed the feedback loop from virtual DT to physical object with an energy focus. Please refer to the literature review [11], section 4.3 Type of digital tech.: Digital Model, Shadow, or Twin for the detailed explanation of this DT. Based on the observations and analysis conducted, more research is needed in the area of closing the loop of the DT from virtual object to physical from an energy perspective.

2.3 Communication Protocols

An objective during the literature review was to investigate the different communication protocols used to transfer data between physical and virtual assets and vice versa in the Energy DTs. A common goal of Industry 4.0 is interoperability. In the case of SMS, a promising approach to reach this goal of interoperability is using standards and information models. The goal of manufacturing information models can be described as the capability of manufacturing enterprises to easily exchange information in a consistent manner within or between each other [13]. The first step to reaching interoperability with the information models is understanding the protocol of communication used by the machine tool, for example, OPC UA, MTConnect, MQTT, etc. The managing director of the MTConnect Institute, Russell Waddell, described communication protocols as the standard on how data is passed from machine to machine, or system to system, whereas the information model is the semantics associated with the data [14]. As the communication protocol can be considered the first step in achieving interoperability, the authors believe it is important to analyze what protocols are used and specified in the literature review papers.

For this use case, it was specifically focused on investigating the utilization of Open Platform Communication - Unified Architecture (OPC UA) in the Energy DTs as the communication protocol. It was uncovered that only 3 papers utilize this communication protocol in their research, which shows there is a need to further research and develop in this aspect. In addition, the only true Energy DT uncovered in the literature review by Park et. al [12] did not utilize OPC UA, rather than Simple Object Access Protocol (SOAP) in their platform.

2.4 Energy Consumption Applications

Another aspect investigated in the literature review was to analyze and classify the energy application proposed and/or implemented in the Energy DT. The types of energy consumption (EC) applications were organized into three levels of implementation: (1) EC Monitoring, (2) EC Analysis, and (3) EC Optimization. These categories are defined by Zhang et al. [15] in their categorical literature review. They distinguished these three categories of EC applications (Figure 4), where EC Monitoring is the most basic approach, then progresses to EC Optimization. Below is further explanation on each of the three EC applications.
EC Monitoring branches into two sub-categories: self-monitoring and physical to virtual monitoring. Self-Monitoring conducts a comparison of the real time energy consumption data from the physical equipment to the historical data in the virtual DT. Whereas physical to virtual monitoring compares the real time EC and parameters with the data simulated by the virtual DT [15].

EC Analysis can be generalized as functions that take energy parameters and associated data and perform different analyses to draw conclusions about the equipment. Four different subcategories fall under EC Analysis. These include Multilevel and Multi-Stage Analysis, Statistical and Predictive Analysis, Sensitivity Analysis, and Behavioral Analysis. Definitions of these four types of analyses are defined in [15].

The final step of EC applications is optimization. EC Optimization contains three different types defined in [15], including Parameter Optimization, Scheduling Optimization, and finally Equipment Upgrading. All of these optimization techniques are the final phase that one can implement into their Energy DT. For the detailed results and analysis, please refer to [11], section 4.5 Energy Applications.

2.5 Energy Meter

Another point of interest was to uncover the different energy meters used to collect data from the physical asset to feed into the virtual DT. This category was included to see if there were any common methods or tools researchers were using to collect energy data from their physical assets. Interestingly enough, no duplicate energy meters were reported. As this research focuses on using the Festo Energy Measurement Box, it was also another point of interest to see if any published research for Energy DTs utilized this proprietary tool for data collection. To review the additional energy meters used by the published research, refer to [11], section 4.6 Energy meters employed.

2.6 Research Gaps

From the review of the literature focusing on Energy DTs, there were several research gaps uncovered. First, there needs to be a push for a universal definition of a DT. All of the papers in
the pool claim they include a DT with an energy focus in their framework or application. In reality, most are either DMs or DSs in accordance with the interpretations of the different types of digital technologies by the authors of the literature review.

Secondly, there was only one true application of a closed loop DT with an energy focus by Park et al. [12], which represents a clear research gap in this area. With the abundance of applications of DS and DT models, it is transparent to see the relative ease of achieving automatic data flow from the physical to virtual object because these two digital technologies require at least this one way automatic data flow for inclusion in the category. The next question is what other approaches can be applied to close the feedback loop from the virtual to the physical object once energy analysis and optimization techniques are conducted at the DT level. In order to start to close this gap, details like communication protocols and energy meters used in the applications need to be included in future DT energy publications. Researchers need to be informed how data is collected and transferred to supply the virtual object and then send data back to the physical to close the feedback loop to allow for replication of results.

Lastly, the review concludes that there needs to be more focus on EC Optimization implementation in Energy DTs. In the research, EC Optimization has not reached the top of the pyramid in popularity, as it still falls behind EC Analysis. Almost all papers analyzed covered EC Analysis within their papers, this category is starting to become saturated. Researchers need to start focusing to the furthest progression of EC applications by incorporating EC Optimization in their DTs. Several papers close the loop of the DT with human interaction, rather than an automated feedback loop. In the examples of an established DT lacking true closed loop feedback, the first step is to focus on the closing the loop in the energy DT, and then move toward the integration of EC Optimization within the process. With that being said, there is a research gap identified to connect the EC Optimization with the energy DT to create a closed loop with the physical asset. Yet, for the cases where an established DT exists with closed loop feedback, the implementation of EC Optimization should be focused on.
3. Research Objective

In the midst of the Fourth Industrial Revolution, or Industry 4.0, there has been a rise in the cost of energy prices all across the globe, placing additional stresses on manufactures as well as consumers. In addition, there has been a sustainability push in the manufacturing sector to protect and save the environment for generations to come. With the implementation of Industry 4.0, the integration of digital technologies has also been the main driver and change from the Third Industrial Revolution. One method manufacturers are implementing into their processes is the use of the DT. As stated in the earlier section, DTs can be designed to have many different purposes and objectives. With all of the existing economic and environmental issues, manufacturers are focusing on Energy DTs to help improve their processes from an energy consumption standpoint.

The main research problem in this area is the lack of examples of full scale Energy DTs with a closed feedback loop. There is an excess of examples of one way data flow, or DSs, but only one ‘true’ DT uncovered during the literature review. Under this consideration, the focus of this research will be to help close this research problem by contributing an example of an Energy DT to the state of the art.

The objectives of this investigation are to examine the use of OPC UA as the data highway from physical to digital assets, as well as the feedback from digital back to the physical asset, to review the literature, and test real-time feedback from an energy optimizing algorithm in the digital space back to the physical. There are three main areas of concentration in the investigation:

1. Investigate previous approaches to understand the data flow of OPC UA from physical space and digital space, and vice versa, and study the effectiveness of the approaches in problem solving.
2. Develop an Energy DT use case and the accompanying method to generate energy optimizing feedback in the digital space back to the physical space.
3. Test and evaluate the developed method, and thus contributing to state of the art of Energy DTs.

3.1 Research Question

How can we use the features of OPC UA to close the energy optimization feedback loop between the physical SM lab system Heating Tunnel and virtual Python DT environment?

Since DTs are a relatively new digital technology, there has been a growing interest and promising research results in the academia sector. Due to their ability to be modeled for many different applications, researchers are finding new ways to implement this technology to create different process improvements. In order to be considered at DT, the method must contain a (1) physical asset, (2) digital asset, (3) automatic data flow between physical to virtual and vice versa. Other than these three requirements, the rest is up to interpretation of the designer. As there was only one ‘true’ Energy DT fitting all the DT criteria in the literature review, there is a clear research gap in this area to investigate different methods to close the feedback loop between assets and provide another example of a ‘true’ Energy DT. There is no evidence in literature utilizing OPC UA for energy optimizing feedback. For this research, the case study will consist of the SMS Festo Cyber-
Physical (CP) Lab Heating Tunnel station as the physical asset. The digital asset requirement will be satisfied in a DT model that monitors the Heating Tunnel station real-time and historic energy consumption data and conducts energy parameter optimizing of the Heating Tunnel heat up time to reduce overshoot from the setpoint value. Lastly, the connection between physical and virtual, and corresponding feedback will be satisfied by the connection of OPC UA. The detailed methodology to solve the problem will be explained in the following section.
4. Research Plan and Methodology

The focus of this research is i) to create a DT of a physical asset in a smart manufacturing system, the CP Lab Heating Tunnel, ii) establish a network connection to its digital asset, and iii) create a close loop feedback with energy optimization in the digital space and conduct real time process control updates to the physical asset. Figure 5 graphically illustrates the structure of the method to achieve the research goal. Each piece of the methodology is explained in the following sections.

![Methodology Structure based on diagrams established in [16]](image)

4.1 Physical Space

This subsection will explain the research methodology pertaining to the physical space of the DT structure including the overall Festo CP Lab testbed (section 4.1.1), the Heating Tunnel station (sec. 4.1.2), the Heating Tunnel limitations and Solution Approach (section 4.1.3), and the Energy Measurement Box (section 4.1.4).

4.1.1 Project SMS Testbed: Festo CP Lab

For this research, the physical SMS testbed consists of the Festo CP Lab as illustrated in Figure 6. The lab is located in the SM lab on the West Virginia University (WVU) Evansdale campus. The CP Lab is an Industry 4.0 learning system that includes industry-level technologies (HMI, RFID, etc.), software (MES, etc.), and components (PLCs, sensors, actuators, etc.) that are used to teach the IIoT, as well as other Industry 4.0 topics [17].
The CP Lab simulates the manufacturing assembly process of cell phones, starting with placing pre-fabricated front covers on the conveyors all the way through to the final output. The WVU CP Lab consists of eight stations, each focused on a specific manufacturing process with individual controls, connected in a square:

1. **CM-AM-MAG-FRONT (Magazine Front):** Places front covers on pallets to begin the manufacturing process using pneumatics.
2. **CP-AM-iDRILL (iDrilling):** Simulates drilling corresponding holes and features into the front of the phone case.
4. **CP-AM-MAG.BACK (Magazine):** Places back covers on top of the front using pneumatics.
5. **CP-AM-MPRESS (Muscle Press):** Uses hydraulic actuated arms to press the top and bottom covers together using an interference fit.
6. **CP-AM-HEAT (Heating Tunnel):** Uses heat to simulate the fusing of the top and back covers together.
7. **CP-AM-TURN (Turning):** Uses a series of pneumatics to flip the workpiece front side up.
8. **CP-AM-OUT (Output):** Automatic removal of parts from the conveyor to corresponding “good” and “bad” quality part compartments.

As mentioned above, the CP Lab simulates the production of cell phones by placing two parts of the housing, top and bottom, together to make a complete part throughout the assembly processes applying a variety of manufacturing processes, shown in Figure 7. The workpiece travels along the connected conveyor belts to its sequential station via a pallet (Figure 8) equipped with a radiofrequency identification (RFID) chip as well as additional identifiers (data matrix code, magnetic identifier). Consequently, each station is equipped with a RFID reader to identify
pallets and corresponding parts. It has to be noted that in the WVU CP Lab, the stations are purposely not arranged in sequential order of assembly. The RFID is used to track the location of each individual pallet in the assembly process and determines whether the part passes through the station without manipulation, or the required manufacturing operation is conducted.

![Top View of Part Housing](image1.png)  
![Bottom View of Part Housing](image2.png)  
![Complete Part](image3.png)  

Figure 7: CP Lab Part Housing

![Pallet Image](image4.png)  

Figure 8: Pallet

In addition, each station features a Human-Machine Interface (HMI) device (Figure 9) for the operator to start and stop the station, as well as analyze inputs and outputs, errors, and process graphs. All stations have a Siemens Simatic ET 200SP programmable logic controller (PLC) that is programmed to perform control functions to read inputs, implement logic, and energize outputs. Each individual station’s PLC is connected through a series of ethernet switches through Profinet protocol and connected to the main CP Lab desktop computer through a singular ethernet connection. On the CP Lab desktop computer resides the Festo Manufacturing Execution System (MES) 4 software where operators can manage, develop, and visualize orders, as well as show production data to increase productivity. Examples of data the software analyzes is an equipment efficiency report (Figure 10) and overall equipment effectiveness (OEE).
4.1.2 CP Lab Heating Tunnel

The main CP Lab station of focus for this research is the CP-AM-HEAT, or Heating Tunnel (Figure 11). In this setup, the station simulates the fusing of the two components that are press-fitted together in the prior process (Muscle Press). The sequence of operations of the Heating Tunnel is summarized in Figure 12. During normal operation, a carrier with a workpiece present breaks a light barrier sensor which initiates the pallet to be stopped at the station and the RFID to
be read. If the RFID tag read indicates the operational sequence to start, automatic program mode starts on the Heating Tunnel. The temperature is measured, and the cross flow blower is turned on. If it has not reached its setpoint of 40°C, the cross flow blower heating element is powered on and operates until the setpoint has been reached in the environment. Once the setpoint is reached, the cross flow blower is switched off, and the drying time starts. After the drying time is complete, and the pallet is released, leaving the station. All information regarding the Heating Tunnel operation was obtained from the Festo Heating Tunnel manual [18].

Figure 11: a) Heating Tunnel Power ON, b) Entrance of Heating Tunnel, c) Front of Heating Tunnel Station
4.1.2 CP Lab Heating Tunnel Limitations and Solution Approach

On a more technical controls side of the CP Lab Heating Tunnel, several limitations regarding the machine and its programming were uncovered over the course of this project. The first limitation
relates to the automatic process execution of the Heating Tunnel operational sequence in Figure 12. Each station in the CP Lab has two modes: Automatic and Setup.

In Automatic mode, the PLC and components run the automatic sequence programmed and installed by Festo, in addition to communicating with the Festo MES. The MES software is located on a local PC that controls the recipes and execution of orders that are sent to the CP Lab. Once an order is sent from the MES, the sequence of processes through the CP Lab stations initiates and is tracked until completion of the order. In addition, when in Automatic mode, the user is only able to read data from OPC UA server, not write back any data.

In Setup mode, the user is able to test different inputs and outputs manually, either by using the HMI attached to the station, or writing values using the OPC UA client. As a drawback, the MES is disconnected from the stations and unable to be used as intended to start and track orders. In result, the sequence of operations in Figure 12 is not completed automatically. In this mode, the users are able to both read values from the OPC UA server, in addition writing values to the OPC UA server.

A main focus of this research is to create a closed loop DT with two-directional automated data flow by reading values from and writing values back to the server and thus making real-time, informed adjustments. To overcome the writing back to OPC UA server hurdle that is established in Automatic mode, the control logic is moved to a digital program in a Python environment while the Heating Tunnel is in Setup mode. Using the Python OPC UA package with modules client and ua, input and output values can be read and written back and forth from the Heating Tunnel OPC UA server to the digital control logic.

For the control logic in Python, a state programming method is implemented. The program is formatted to continuously read the inputs, determines the state of the system based on specific logic conditions of the inputs, and then executes a specific function based on state. The state of the Heating Tunnel could be determined as one of the following three:

- **State 1: No Pallet**
- **State 2: Waiting**
- **State 3: Execute Heating Tunnel sequence**

Each state and their occurrences in the flow chart are indicated by the corresponding state color Figure 13. For **State 1: No Pallet**, the inputs are continuously read using the OPC UA Python client until the program detects a change at the front conveyor start proximity sensor (xBG5). Once this value changes from false to true, the conveyor at the station is turned on, and state is switched to **State 2: Waiting**. State 2 occurs when either there has been a change at the front proximity sensor and the conveyor is on, but the pallet has yet to reach the stop sensor (xBG1) located in the Heating Tunnel, or when the pallet is released from stop sensor xBG1 and the program scan is waiting for a change at the conveyor end proximity sensor (xBG6). **State 3: Execute Heating Tunnel sequence** occurs when the stop sensor (xBG1) changes from a false to a true state, indicating there is a pallet ready to be processed. To account for additional pallets, there is a condition placed after the operational sequence to reread stopper sensor xBG1. If this value is true, then the **State 3: Execute Heating Tunnel sequence** operates again. Once all pallets are done being processed, the state
changes to *State 2: Waiting* for all pallets to leave the conveyor and BG6 becomes true. Once true, the conveyor is turned off and the state of the system turns to *State 1: No Pallet* and the inputs continue to be read.

As shown in the flowchart of the control logic, there is also a limitation with the Heating Tunnel system on the input power of the crossflow blower. The system is programmed to only have two state levels of power input as a Boolean type: 0 (False) – 500W or 1 (True) – 1000W. The user only has these two options when conducting the heating process. The system offers a ‘auto’ setting, which however is identical with the 500W setting.
Figure 13: Flowchart for System Control State Logic
4.1.3 Festo Energy Measurement Box

In order to get detailed data regarding the energy consumption of the Heating Tunnel, the Festo Energy Measurement Box: Single Phase is connected to the Heating Tunnel station. The Energy Measurement Box (EMB) is used to monitor single phase physical assets, such as the CP Lab, by recording and processing the electricity consumption and compressed air usage communicated via the network [19]. The EMB has the ability to monitor up to three stations with single phase connections for electrical consumption analysis, as well as compressed air consumption, monitoring both air flow and air pressure. Figure 14 displays the components used for power measurement, compressed air measurement, and control and communication with the corresponding colored boxes to illustrate the location of each part of the box.

### Power Measurement
- Siemens SENTRON PAC3220 power meter for 3 measuring channels
- 3 x IEC 60320-1 C13 connector

### Compressed Air Measurement
- 3 x Festo SFAH IO-Link flow sensor
- 3 x Festo SPAU IO-Link pressure sensor
- 3 x In and 3 x Out plug-in connections for 6mm hoses

### Control and Communication
- Festo CPX-E PLC with IO-Link master, web server, OPC UA server
- RJ45 LAN connection

![Figure 14: Festo Energy Measurement Box (EMB)](image)

Once properly connected, the EMB has the ability to monitor several different variables of interest. For the electricity consumption, examples of variables it can monitor include active energy (Wh), active power (W), reactive power (VAR), and apparent power (VA). For air consumption, examples of variables that can be monitored include air flow (L/min) and air pressure (bar). The parameters being monitored of interest will be explained in more detail in Section 4.3.

4.2 Connection: OPC UA

In order to achieve the objective of real-time feedback from the physical to digital assets and vice versa, OPC UA communication protocol and its features will be utilized. Released in 2008, the OPC foundation defines OPC UA as the “an industrial open communications standard by the OPC foundation that integrates the functionality of individual OPC specifications to one extensible framework” [20]. OPC UA effectively uses servers to collect data from physical assets and map them to an address space where a user can access the data in a read or write format based on permissions, manage data subscriptions, and monitor certain events from a distanced location via a client software.
An overview of the OPC UA structure in context with this use case is illustrated in Figure 15. Data is generated by the CP Lab’s Heating Tunnel station from its inputs, outputs, and the Profinet ethernet protocol. In this use case, there are two OPC UA servers: one on the Heating Tunnel station Siemens PLC, and the other in the EMB Siemens PLC. The data generated from the station and EMB then is transferred through its corresponding OPC UA server. Data can travel from the OPC UA server to the user via a server and client model. The client requests a response from the servers, and they send a response back to the client with the answer to request. An advantage of OPC UA server and client interaction is to be able to read data from the servers to the clients, but also write data back to the servers and make changes in real-time. A similar method will be used to interact with the DT environment. The physical asset data will be sent to the DT Python environment where the data will be real-time monitored and the optimization implementation will take place. Once an optimized process control parameter is obtained, it will be sent back to the client, which will write the new parameter to the servers and update accordingly.

4.3 Digital Space

As mentioned in previous section, the digital space of the DT is a large sandbox that is open for many options of analysis and optimization. In this use case, energy optimization will be further investigated as a method a DT can be used for in a SMS.
4.3.1 Monitoring Parameters

The focus of the digital space is to monitor real-time and historic energy data from the physical asset – the Heating Tunnel. Using this data, optimization focused on reducing energy consumption can be applied, where the updated parameters can be sent back and updated automatically. Based on the energy focus of the DT, energy related parameters are needed to be tracked and modeled from the Heating Tunnel and EMB OPC UA servers. Parameters of interest are located in Table 1.

Table 1: Parameters of Interest

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Server</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature</td>
<td>Celsius (°C)</td>
<td>Heating Tunnel</td>
<td>Temperature inside Heating Tunnel</td>
</tr>
<tr>
<td>Active Energy</td>
<td>Watt-hour (Wh)</td>
<td>EMB</td>
<td>Active Energy used by Heating Tunnel (cumulative throughout data collection)</td>
</tr>
<tr>
<td>Voltage</td>
<td>Volts (V)</td>
<td>EMB</td>
<td>Real-time voltage utilization of the system</td>
</tr>
<tr>
<td>Current</td>
<td>Amperes (A)</td>
<td>EMB</td>
<td>Real-time current utilization of the system</td>
</tr>
<tr>
<td>Active Power L2</td>
<td>Watt (W)</td>
<td>EMB</td>
<td>Power drawn by Heating Tunnel components</td>
</tr>
</tbody>
</table>

To actively monitor the parameters during the duration of the servers being active, the software Ignition Supervisory Control and Data Acquisition (SCADA) by Inductive Automation is used. Ignition SCADA incorporates instant web-based deployment with a toolset for visualization, supervisory control, acquisition of manufacturing data in a singular platform [21]. Ignition is simple to use and software that can be easily implemented into any manufacturing data highway, especially PLCs and OPC UA servers. OPC UA servers can be connected and recognized through the Ignition gateway server if connected to the process network. From there, the nodes from the OPC UA server can be connected and different tools can be configured and designed to visualize the data in real-time via a user dashboard. The dashboard is not only restricted to the PC it was created on, but also can be viewed with any device that has web browser capability with the use of web-based clients. In order to have the data be read in real-time, the nodes must be connected to an active database to record historical data. An SQLite Database will be used to record the historical and real-time data at a rate of 1 sample/second. Figure 14 illustrates the movement and connection of data using Ignition, starting from the PLCs, moving through the Ignition server, all the way to the web based clients and designers. An example of a dashboard created with Ignition is illustrated in Figure 15.
4.3.2 Energy Optimization

As discussed in section 2.4, there are three applications of EC in a digital space. **EC Optimization** is focused on parameter optimization, scheduling optimization, and equipment upgrading [15]. The optimization technique used for this case study to incorporate into the real-time feedback of the system is parameter optimization. The graph illustrated in Figure 18 shows the visual representation of the temperature measured in the Heating Tunnel from starting the Heating Tunnel sequence to cooling down to approximately 27°C for an individual pallet (process cycle). At time 0 seconds (s), the crossflow blower turns on to heat up the tunnel. Once the tunnel temperature sensor equals the setpoint of 40°C, the crossflow blower is turned off. Due to residual heat from the crossflow blower heating element and the closed nature of the Heating Tunnel system, the
temperature continues to rise, creating a substantial overshoot (excess heat). At a process control level, the system produces a significant amount of overshoot from the setpoint temperature of 40°C. At a power input of 500W, the maximum temperature reached is approximately 43.60°C, which corresponds to 3.60°C (or ~9%) of overshoot and remains above the setpoint for about 70s. At power input of 1000W, the maximum temperature reached is approximately 47.03°C (or ~14% above setpoint) and remains above the setpoint temperature for about 96s. Since sampling of data is at a rate of one sample per second, all time values will be rounded to the nearest integer.

The overshoot of the temperature in the Heating Tunnel during the process can lead to a number of unwanted factors in a manufacturing process. First, the increase in temperature from the setpoint can lead to overheating the part, and in result can create a variety of quality issues. If the part is overheated, the material properties could alter from the original specifications and not perform as intended. Secondly, the presence of overshoot directly correlates to an increase in energy consumption. If overshoot occurs, that means the process is consuming more power over time than it needs to achieve the intended process outcome. The increase in power consumption overtime results in an increase in avoidable costs as well, as the active energy consumption increases.

![Temperature of Heating Tunnel at 500W and 1000W](image)

**Figure 18: Temperature of Heating Tunnel at 500W and 1000W vs Time**

To reduce the quality issues pertaining to the Heating Tunnel heating up process, the temperature overshoot from both power input levels must be decreased and more controlled. There are several different ways this can be accomplished. One method includes manually adjusting setpoint temperatures and running trial and error tests. Others include fitting the preexisting process data to curves and predicting optimization parameters using mathematical models, and further
extending to training supervised or unsupervised ML/AI algorithms to accomplish the same task. This research selects the option of fitting curves to the original temperature data. From there, the mathematical models will be used to predict the temperature the crossflow blower needs to be turned off in order for the temperature to peak to an approximate setpoint value and designated time the process can be in overshoot. This method is selected due to its ease of implementation in the virtual environment, as well as the ability to achieve the research goal of bidirectional communication from the DT environment to the control logic environment, and vice versa.

The initial values selected for this research are a setpoint temperature of approximately 40°C with an overshoot time of 10s. The setpoint of 40°C is selected as it is the original parameter value provided by the manufacturer, Festo Didactic. Secondly, the processing time at setpoint temperature for one individual pallet is 5s, so the selection of the overshoot time of 10s can ensure that the temperature remains above the setpoint value for the duration of the processing time. The next sections will describe the methodology to fit the data to the best set of curves for prediction.

500W Power Input Level

The 500W input power level to the crossflow blower creates approximately 4°C of overshoot for 70s. As a process requirement for fusing the two parts together without causing quality issues, the temperature must be maintained within a range from the setpoint temperature to a maximum of +5% for at least 5s. The overshoot time for one individual pallet is set to 5s plus the manufacturing requirement of 5s of processing time at setpoint temperature, equating to 10s. This minimum time value selected to ensure the part is heated for at least 5s with a buffer for account for process variability. To summarize, the goal of the optimization is to reduce the overshoot to approximately 10s and maintain the temperature overshoot to not exceed 42.00°C or 5%.

The first step into reducing the overshoot and optimizing the heating up time of the Heating Tunnel is to fit the original temperature curve. There are three piecewise functions that describe the temperature through a full heating and cooling cycle of one individual pallet, shown in (1).

\[
y(t) = \begin{cases} 
0.39t + 25.81, & 0 < t \leq 47 \\
-0.07t + 47.40, & 47 < t < 174 \\
0.000013t^2 - 0.02t + 38.32, & t \geq 174 
\end{cases}
\]  

(1)

The first piecewise function describes the Heat Up phase, as a positive linear relationship. This phase includes the time when the power to the crossflow blower turns on, reaches its setpoint temperature, turns off the crossflow blower, and the residual heat temperature increase to the peak temperature. The cooling phase of the Heating Tunnel can be split into two separate functions. The Rapid Cool phase occurs as a negative linear relationship. This includes the decrease in temperature from the peak to approximately 120s after the peak temperature has been reached. Once Rapid Cool is finished, the residual heat for extended cooling moves into a quadratic relationship until it is cooled to room temperature again, called the Extended Cool phase. The original 500W temperature data, piecewise functions, and setpoint line are graphed in Figure 19.
First linear regression of the *Heat Up* phase is acquired by slicing the original raw temperature data from starting to heat up (t=0s) to maximum temperature (t=53s) of the original data. The data was fit using Python linear regression and showed a positive linear relationship (R=0.98). Secondly, the *Rapid Cool* negative linear equation is determined by fitting the original temperature data from the maximum temperature (t=53s) to the point where it reaches approximately 35°C (t=160s), the maximum temperature using Python linear regression (R=0.99). The *Extended Cool* phase was fit using a parabolic function in Python, where the remaining original data was used in the calculation. Since the *Extended Cool* phase is not truly a concern, as this only occurs when the machine is not being used for a period of time (more than one min) or the machine is shut down for the day, the assumption of linearity for *Rapid Cool* down interval to end where the two graphs intersect can be used, corresponding to t=174s. Also, it does not use active power and increase energy consumption. With this assumption, the piecewise functions can be connected to create a “continuous” graph to model the temperature.

As mentioned earlier, the time in temperature overshoot for the 500W power input setting is 70s. As an optimization input parameter, the time in overshoot is desired to be approximately 10s. It must be noted that the linear fit equations in (1) are used in the determination of the optimized equations. Due to the linearity aspect of the piecewise functions of the *Heat Up* and *Rapid Cool* phases, the time that the two lines cross can be solved for by setting the two equations equal to each other and solving for \( t \). The crossing time is 47s at 44.14°C. Using the *Heat Up* linear equation, the time that the fitted line crosses 40°C can be calculated, which equivalates to 36s. This method was used to find the time that the *Rapid Cool* equation crossed 40°C at 106s.
the difference between peak temperature time and the time when the Heat Up equation passes the 40°C setpoint, the timespan of the overshoot in the Heat Up phase can be calculated as 11s. Similarly, the duration of the Rapid Cool overshoot was determined as 59s. Using the proportional properties of triangles and the design parameters for optimization, the new optimized parameters can be calculated, as shown in in Figure 20.

![Figure 20: Proportional Triangles for 500W](image)

The goal is to reduce the overshoot time to 10s with a maximum overshoot of 42°C. The same Heat Up phase time (t=36s) as the original Heat Up equation that crosses 40°C is used because this value is deterministic of the Heat Up process. A proportion (2) can be set up using the original (70s) to optimized overshoot time length (10s) and the timespan of the original Heat Up phase, \( x_1 \), to calculate the unknown optimized Heat Up phase time length (\( x_3 \)), determined as 2s. Adding the 2s to 36s, a peak time of 38s is established. A second proportion (3) was set up utilizing the original Rapid Cool timespan of 59s to determine the 8s length of the optimized Rapid Cool time, denoted as \( x_4 \). Using the Heat Up equation in (1), the new optimized maximum temperature can then be calculated as 40.63°C. Finally, the new optimized Rapid Cool equation needs to be calculated due to the parallel shift of the equation to the left. Since the slope is the same due to properties of parallel linear functions, point-slope form (4) can be used to achieve the new Rapid Cool equation.

\[
\frac{x_3}{11s} = \frac{10s}{70s} \Rightarrow x_3 = 1.57s \sim 2s \quad (2)
\]

\[
\frac{x_4}{59s} = \frac{10s}{70s} \Rightarrow x_4 = 8.43s \sim 8s \quad (3)
\]

\[ x_3 = \frac{10s}{70s} \Rightarrow x_3 = 1.57s \sim 2s \quad (2) \]

\[ x_4 = \frac{10s}{70s} \Rightarrow x_4 = 8.43s \sim 8s \quad (3) \]
Using the optimized parameters and equations, a new piecewise function can be determined that shifts the temperature curve down to the desired 10s overshoot constraint, as shown in (5). Figure 21 graphs the new fitted piecewise function. The *Heat Up* time has been reduced to 38s. As before, the intersection of the *Rapid Cool* and *Extended Cool* functions is the end of the *Rapid Cool* phase and the beginning of *Extended Cool* respectively. The shift of the *Extended Cool* equation and corresponding intersection needs to be calculated. In the *Rapid Cool* equation of (1), the interval ends at 174s, which corresponds to a temperature of 35.22°C. Using this temperature in the *Rapid Cool* equation of (5), the calculated time is 115s. This is a shift of 59s. As a property of parabolas, adding the 59s to the \( t \) values will result in a horizontal shift left. Using the original equation with the shift incorporated, the new intercept can be calculated as \( t = 126s \).

\[
y(t) = \begin{cases} 
0.39t + 25.81, & 0 < t \leq 38 \\
-0.07t + 47.40, & 38 < t < 126 \\
0.000013(t + 59)^2 - 0.02(t + 59) + 38.32, & t \geq 126
\end{cases}
\]  

(5)

---

**Figure 21:** Reduced Temperature Overshoot at 500W

The based off difference (delta) between the original maximum predicted temperature \( (44.14°C) \) value and the new optimized target maximum temperature \( (40.63°C) \) equaling 3.51°C, the tunnel heating element can be shut off once the temperature reaches approximately 37.12°C. The original temperature curve has a vertical shift down 3.51°C to the optimized parameter, so subtracting the
difference from the optimized target maximum temperature parameter of 40.63°C will create the new “setpoint” value, or signal when the crossflow blower can be turned off. When the temperature reaches the 37.12°C, there should be enough residual heat to reach across the 40°C setpoint for approximately 10s.

_1000W Power Input Level_

A similar approach to was applied to the 1000W input power level to find the optimized setpoint temperature. For the 1000W input to the crossflow blower, the process creates approximately 7°C of overshoot for 96s. To improve this process, the goal, like in the 500W case, is to reduce the overshoot to approximately 10s and maintain the temperature overshoot to not exceed 42.00°C.

Similar to the 500W case, the original temperature curve is fit using three piecewise functions that describe the temperature through a full heating and cooling cycle, shown in (6). The first piecewise function describes the _Heat Up_ phase, as a positive linear relationship. This phase includes the time when the power to the crossflow blower turns on, reaches its setpoint temperature, power turns off to the crossflow blower, and the residual heat temperature increase to the peak temperature. The cooling phase of the Heating Tunnel can be split into two separate functions. The _Rapid Cool_ phase occurs as a negative linear relationship. This includes the decrease in temperature from the peak to approximately 138s after the peak temperature has been reached. Once _Rapid Cool_ is finished, the residual heat for _Extended Cool_ moves into a quadratic relationship until it is cooled to room temperature again. The temperature graph with the original 1000W temperature data, piecewise functions, and setpoint line are graphed in Figure 22.

\[
y(t) = \begin{cases} 
0.58t + 25.91, & 0 < t \leq 38 \\
-0.09t + 50.76, & 38 < t < 177 \\
0.000013t^2 - 0.03t + 39.71, & t \geq 177
\end{cases} \quad (6)
\]
First linear regression of the *Heat Up* phase is acquired by slicing the original raw temperature data from starting (t=0s) to heat up to max temperature (t=44s). The data was fit using Python linear regression and showed a strong positive linear relationship (R=0.96). Secondly, the *Rapid Cool* negative linear equation is determined using the same linear regression method in Python (R=0.99). The original raw temperature data was sliced from maximum temperature (t=44s) to the point where it reaches approximately 35°C (t=161s). The *Extended Cool* phase was fit using a parabolic function in Python, where the remaining original data was used in the calculation. The same assumption of linearity for *Rapid Cool* extended down to where the two graphs intersect at t=177s can be used. With this assumption, the piecewise functions can be connected to create a “continuous” graph to model the temperature.

The time in temperature overshoot for the 1000W power input setting is 96s. As an optimization input parameter, the time in overshoot is desired to be approximately 10s. The linear fit equations in (6) are used in the determination of the optimized equations at 1000W. The *Heat Up* and *Rapid Cool* phase equations can be set equal to each other and solved for $t$ to find the time of intersection and corresponding temperature. The two piecewise functions cross at 38s and 47.95°C. The time the *Heat Up* linear equation crosses the setpoint temperature of 40°C can be calculated as 24s. Similarly, the *Rapid Cool* equation crossed 40°C at 120s. Using the difference between peak temperature time and the time when the *Heat Up* equation passes the 40°C setpoint, the timespan of the overshoot in the *Heat Up* phase can be calculated as 14s. Similarly, the duration of the *Rapid Cool* overshoot was determined as 82s. Using the proportional properties of triangles and the
design parameters for optimization, the new optimized parameters can be calculated, as shown in
in Figure 23.

![Figure 23: Proportional Triangles for 1000W](image)

Again, the goal is to reduce the overshoot time to 10s with a maximum overshoot of 42°C (or
+5%). To create an origin of the optimized triangle, the original Heat Up phase where the original
Heat Up equation in (6) that crosses 40°C at t=24s is used because this value is deterministic of
the Heat Up process. Secondly, a proportion (7) can be setup using the original (97s) to optimized
overshoot time length (10s) and the timespan of the original Heat Up phase (x₅) to calculate the
unknown optimized Heat Up phase time length, denoted by x₇. This value is determined as 1s.
Adding the 1s to 24s, a peak time of 25s is established. A second proportion (8) was set up with
the same overshoot length proportion as (7), instead utilizing the original Rapid Cool timespan
(x₆) to determine the 9s length of the optimized Rapid Cool time, x₈. Using the Heat Up equation
in (6), the new optimized maximum temperature can then be calculated as 40.41°C. Finally, the
new optimized Rapid Cool equation needs to be calculated due to the parallel shift of the equation
to the left. Since the slope is the same due to properties of parallel linear functions, point-slope
form (9) can be used to achieve the new Rapid Cool equation.

\[
\frac{x_5}{14s} = \frac{10s}{96s} \Rightarrow x_7 = 1.46s \sim 1s
\]

(7)

\[
\frac{x_6}{82s} = \frac{10s}{96s} \Rightarrow x_8 = 8.54s \sim 9s
\]

(8)

\[
y - y_1 = m(x - x_1)
\]

where: \(m = -0.09,\) and \((x_1, y_1) = (25s, 40.41°C)\)

(9)
Using the optimized parameters and equations calculated above, a new piecewise function can be defined that shifts the temperature curve vertically to the desired 10s overshoot constraint, as shown in (10). Figure 24 graphs the new fitted piecewise function. The Heat Up time has been reduced to 25s. As before, the intersection of the Rapid Cool and Extended Cool functions is the end of the Rapid Cool phase and the beginning of Extended Cool respectively. The shift of the Extended Cool equation and corresponding intersection needs to be calculated. In the Rapid Cool equation of (6), the interval ends at 177s, which corresponds to a temperature of 34.80°C. Using this temperature in the Rapid Cool equation of (10), the calculated time is 87s. This is a shift of 139s. As a property of parabolas, adding the 139s to the $t$ values will result in a horizontal shift left. Using the original Extended Cool equation with the shift incorporated, the new intercept with the Rapid Cool can be calculated as $t=87s$.

$$y(t) = \begin{cases} 0.58t + 25.91, & 0 < t \leq 25 \\ -0.09t + 42.66, & 25 < t < 87 \\ 0.000013(t + 139)^2 - 0.03(t + 139) + 39.71, & t \geq 87 \end{cases}$$

(10)

![Reduced Overshoot 1000W](image)

**Figure 24: Reduced Temperature Overshoot at 1000W**

Based off difference of the original maximum temperature (47.59°C) value and the optimized (40.41°C) equaling 7.14°C, the tunnel heating element can be shut off once the temperature reaches approximately 32.87°C. The original temperature curve has a vertical shift down 7.14°C to the optimized parameter, so subtracting the difference from the optimized maximum temperature value will create the new “setpoint,” or to signal when the crossflow blower can be turned off. When it reaches the 32.87°C temperature, it will have enough residual heat to reach across the 40°C setpoint for approximately 10s.
4.3.3 Digital Twin

A focus of the research is to create a use case of an Energy DT and contribute to the state of the art. As mentioned previously, DTs are a sandbox that are open to modeling based on the application and process at hand. For this use case, the DT of the Heating Tunnel focuses strictly on aspects relating to EC. The two parts to the EC DT are: (1) simulation using static EC prediction analysis and (2) real-time EC analysis and optimization feedback.

**Static EC Prediction Simulation**

The first part of the DT creates a simulation of the EC of the Heating Tunnel based off of user input parameters, then calculates the active energy used in kilowatt-hours (kWh) and cost of the run in US dollars (USD). The prediction is for one heating cycle of one part. To start, the simulation requests three parameters from the user:

1. Setpoint (in °C, must be between 35 and 60°C)
2. Wattage (0 for 500W or 1 for 1000W)
3. Overshoot time (default 10s)

After the inputs are stored, the sequence of operations to calculate the optimized mathematical models are called in a function to predict the new *Turn off Temperature* parameter. The *Turn off Temperature* is the temperature value in °C during the heating cycle where the crossflow blower is turned off based on the *Heat Up* piecewise equations and proportions established in (5) for 500W and (10) for 1000W in section 4.3.2. The pseudo code below describes the logic to calculate the *Turn off Temperature*.

**Algorithm for Finding Optimization Parameters**

**User inputs:** Setpoint, Wattage, Overshoot Time

**Calls function** to calculate Turn off Temperature

<table>
<thead>
<tr>
<th>Wattage</th>
<th>Calculation Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>Using <em>Heat Up</em> equation in (5): Calculate the time equation crosses Setpoint Calculate <em>Heat Up</em> overshoot time using proportion Calculate <em>Rapid Cool</em> overshoot time using proportion Calculate time at max temperature Using <em>Heat Up</em> (5), calculate maximum temperature Calculate Turn off Temperature Create xHeatUp array of integers the length of time at max temperature Using <em>Heat Up</em> (5), create yHeatUp array of values using xHeatUp as inputs</td>
</tr>
<tr>
<td>1000</td>
<td>Using <em>Heat Up</em> equation in (10): Calculate the time equation crosses Setpoint Calculate <em>Heat Up</em> overshoot time using proportion Calculate <em>Rapid Cool</em> overshoot time using proportion Calculate time at max temperature Using <em>Heat Up</em> (10), calculate maximum temperature Calculate Turn off Temperature Create xHeatUp array of integers the length of time at max temperature Using <em>Heat Up</em> (10), create yHeatUp array of values using xHeatUp as inputs</td>
</tr>
</tbody>
</table>
After the parameters have been calculated, the code then determines the energy consumption and corresponding cost of the order. Equation (13) illustrates the equation used to calculate the total active energy used throughout the Heat Up time frame inWs. The Heat Up time frame in the static DT case consists of the length of seconds the crossflow blower is on and consuming power, which equavalates to the time that Heating Tunnel temperature is below the Turn off Temperature threshold value. It models the summation of power consumption overtime to calculate the running EC total while an order is running, and then returns the final Ws EC total when the order completes. The monitoring starts when the when the heating up sequence starts \( t = 0 \), and then finishes once the heating sequence ends and the crossflow blower is turned off, indicated by the \( t = n \). Each second, the power value indicated by \( P_l \) measured in W, at the corresponding \( t(i) \) value is calculated and multiplied by the 1 second sample time, equavalating to the active energy value for that instance. The summation of all the active energy values throughout the sequence time frame results in the total active energy used for the Heat Up period of the heating sequence. As the heating up time is the only time that a significant amount of power is drawn, this will be the main focus of monitoring. To return kWh, the final active energy total value in Ws can be divided by 3600 and 1000, respectively.

\[
ActiveEnergy_{TOTAL} = \sum_{i=0}^{n} y(t(i)) \times t \\
t = 1s \\
y(t(i)) = P_l
\]  

To calculate cost of an order, the total Active Energy value of the order is multiplied by the industrial electricity rate to provide an end cost. Equation (14) shows the equation to calculate the total electrical cost of running an order. Since this research is taking place in Morgantown, West Virginia (WV), the average industrial rate of WV at $0.0723 per kWh [22] is used in the calculation.

\[
Cost = ActiveEnergy_{TOTAL} \times \frac{ElectricityRate_{Industrial}}{kWh} \\
Cost = ActiveEnergy_{TOTAL} \times \frac{0.0723}{kWh}
\]  

Real-Time EC Analysis and Optimized Feedback

The second aspect of the DT interacts with the control logic of the Heating Tunnel execution program to acquire input parameters, conduct optimization calculations in the DT environment, and then send the updated parameters back to the control logic to reflect real time operation of the system. First, the control logic sends its parameters of the input power value (500W or 1000W) to the DT. Using the same logic and calculations as the static DT simulation, it determines the Turn off Temperature value using predefined setpoint and overshoot time values, normally 40°C and
10s, respectively. These values must be set by the user before the program is initiated due to the limitations of the Heating Tunnel system in Setup mode. While in Setup mode, the logic is unable to communicate with the MES software where it would obtain this data normally set by the operator. To overcome this obstacle, the value must be stored and adjusted manually in the logic. In the original Heating Tunnel process, the programmed logic is built to have the part constantly heating until the setpoint temperature is reached with no maximum temperature limitation. As long as the temperature is above the setpoint for the 5s processing time for the part, it qualifies as a completed part. The process does not take into consideration the maximum temperature achieved or the timeframe the temperature is in overshoot. Due to this logic, the introduction of the overshoot time is considered a new user-defined parameter, so it also must be manually adjusted in the logic. Lastly, it monitors and stores the real time parameters in Table 1 from the system during an order to calculate the energy consumption and corresponding cost of the order using (13) and (14).
5. Data Processing and Validation

In this section, using the methodology framework described in Section 4, the data processing and testing validation will be introduced with the WVU SM CP Lab as the physical testbed for the research.

5.1 Energy DT and Optimization Testing

Testing the Energy DT and optimization model will be conducted by comparing the simulated DT environment results with the actual process data collected through test runs varying the parameter settings of the Heating Tunnel. Table 2 illustrates the runs that will be used to validate the methodology and optimization logic. A base run at setpoint temperature 40°C at 10s overshoot time for both 500W and 1000W will be the first parameters tested and validated. A complete run is considered from the time when the Python control logic detects a pallet at the stop sensor (xBG1) initiating the starting of power draw of the crossflow blower until completion of Python Interpreter 2. This includes the entire Heat Up phase used in the analysis, in addition to partial Rapid Cool phase to fit the curves. Since the focus of the optimization is the Heat Up phase, partial Rapid Cool phase is selected to speed up testing cycle time. A new test is started when the temperature cools down to approximately 26°C (room temperature at the facility at time of tests conducted). With the successful completion of the base runs to validate the setup, the experiment and data collection is expanded by including the varying setpoint temperatures and overshoot times summarized in Table 3.

Table 2: Base Runs (parameter setting)

<table>
<thead>
<tr>
<th>Setpoint Temperature</th>
<th>Wattage Level</th>
<th>Overshoot Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>40°C</td>
<td>500W</td>
<td>10s</td>
</tr>
<tr>
<td>40°C</td>
<td>1000W</td>
<td>10s</td>
</tr>
</tbody>
</table>

Table 3: Validation Runs (parameter setting)

<table>
<thead>
<tr>
<th>Setpoint Temperature</th>
<th>Wattage Level</th>
<th>Overshoot Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>45°C</td>
<td>500W</td>
<td>10s</td>
</tr>
<tr>
<td>50°C</td>
<td>500W</td>
<td>10s</td>
</tr>
<tr>
<td>45°C</td>
<td>1000W</td>
<td>10s</td>
</tr>
<tr>
<td>50°C</td>
<td>1000W</td>
<td>10s</td>
</tr>
<tr>
<td>40°C</td>
<td>500W</td>
<td>15s</td>
</tr>
<tr>
<td>45°C</td>
<td>500W</td>
<td>15s</td>
</tr>
<tr>
<td>50°C</td>
<td>500W</td>
<td>15s</td>
</tr>
<tr>
<td>40°C</td>
<td>1000W</td>
<td>15s</td>
</tr>
<tr>
<td>45°C</td>
<td>1000W</td>
<td>15s</td>
</tr>
<tr>
<td>50°C</td>
<td>1000W</td>
<td>15s</td>
</tr>
<tr>
<td>40°C</td>
<td>500W</td>
<td>20s</td>
</tr>
<tr>
<td>45°C</td>
<td>500W</td>
<td>20s</td>
</tr>
<tr>
<td>50°C</td>
<td>500W</td>
<td>20s</td>
</tr>
</tbody>
</table>
The data that will be collected and calculated during the runs include:

- **Time Power ON**: time in seconds that the crossflow blower is powered on with a value of TRUE.
- **Time at Max Temperature**: time value in seconds from the beginning of a run at \( t = 0 \) s to the time the maximum peak temperature is reached.
- **Maximum Temperature**: maximum temperature value in °C of the run.
- **Overshoot Time**: time range in seconds the temperature stays above the setpoint.
- **Energy used**: calculated value using (13) in kWh
- **Cost**: calculated value using (14) in USD

### 5.1.1 Physical Space Setup

The physical space used for the testing and validation is the WVU CP Lab Heating Tunnel station. As mentioned previously, the Heating Tunnel must be placed in **Setup** mode implementing the logic presented in section 4.1.2 in order to complete the OPC UA bidirectional feedback communication.

Several preparation steps must be completed before a run is performed. First, the appropriate power setting must be selected by the operator on the hardwired switch on the Heating Tunnel, as well as selected on the setup screen on the HMI. Both of these selections must be the same in **Setup** mode to ensure the desired power level is read by the OPC UA client in Python and the correct current value is being drawn from the transformer on the hardware side of the **Heat Up** process. Once this is completed, the Heating Tunnel is prepared for a run and ready to receive a pallet.

Since the MES system is not connected in **Setup** mode, pallets are manually sent into the Heating Tunnel from the previous station (Muscle Press). Once the heating process cycle is complete, the pallet leaves the station and continues on to the next station (Turning). It has to be noted, this research is solely focused on the Heating Tunnel processing sequence.

### 5.1.2 Connection: OPC UA

The data communication method used during testing is OPC UA via a Local Area Network (LAN) connection. The CP Lab Heating Tunnel and EMB are connected via separate ethernet cables to a network switch on the CP Lab. This connection is then routed to a local computer running Windows 10 with a Universal Serial Bus (USB) to ethernet adapter, as shown in Figure 25. A separate network is setup on the local computer in order to communicate with the Python client.

<table>
<thead>
<tr>
<th>Temperature</th>
<th>Power</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>40°C</td>
<td>1000W</td>
<td>20s</td>
</tr>
<tr>
<td>45°C</td>
<td>1000W</td>
<td>20s</td>
</tr>
<tr>
<td>50°C</td>
<td>1000W</td>
<td>20s</td>
</tr>
</tbody>
</table>
PyCharm integrated development environment software is used to create a client using the Python OPC UA package with modules client and ua. The OPC UA servers from both the Heating Tunnel and the EMB establish a connection to the client in the beginning of the code execution. Utilizing this client and server model, input and output values can be read and written back and forth from the Heating Tunnel OPC UA server to the digital control logic. The data can only be read from the EMB servers for data collection purposes.

5.1.3 Digital Space Setup

The digital space setup for testing and validation purposes will now be explained in detail. As mentioned before, a Python Environment in PyCharm is used to house the digital space for running the control logic, as well as the DT and data monitoring.

For the testing runs, a compound run configuration of two Python files is setup. This enables two Python interpreters to run in parallel, allowing the control logic and energy data collection code files to run simultaneously. The parallel configuration is one way to record the energy data for the entire Heat Up and Cool Down phases and conduct analysis without interfering with the execution of the control logic.

The first interpreter executes the digital control logic established in section 4.1.2 using the state-case programming method. Once State 3: Execute Heating Tunnel sequence is recognized, the program calls the Pallet Sequence function. This function executes the manufacturing process of the Heating Tunnel. Within the duration of the Pallet Sequence function, the Energy DT Feedback and Energy Data Collection functions are called as well. Energy DT Feedback is the function where the DT modeling calculations take place. An input parameter of power level of the system (500W or 1000W), denoted by PowerVal, is read from the Pallet Sequence function. The feedback
parameter of temperature the crossflow blower will turn off, denoted by \textit{TurnOffTemp}, is calculated and sent back to \textit{Pallet Sequence} in order to create real-time feedback to the system. Throughout the duration of the \textit{Heat Up} phase, or the time the crossflow blower is on (TRUE), the \textit{Energy Data Collection} function is called to record the parameters stated in Table 1, which will be stored in a Pandas data-frame (df). After the system reaches the new \textit{TurnOffTemp} value and the crossflow blower is turned off, the \textit{Energy Data Collection} is completed. Utilizing the data collected from the \textit{Energy Data Collection} function, the program calculated the total power consumption (13) and cost (14) at the termination of the \textit{Heat Up} phase.

The second Python interpreter executes the program to continuously record the energy data for 150s. This time value (150s) is selected as the termination point as it captures all of the data in the \textit{Heat Up} phase, but also records enough of the \textit{Rapid Cool} phase to fit the curve and is based on initial experimentation. The parameters in Table 1 are recorded, in addition to the state (True or False) of the crossflow blower. This parameter is added to aid in data processing to determine the duration the crossflow blower is active drawing the respective wattage. At the termination of the parallel run configuration, the summary calculations are generated by calling the \textit{AnalysisDT} function. In addition, the graph of the actual recorded temperature data versus the prediction graph is generated. Figure 26 displays the flowchart of the two Python interpreters run in parallel with the corresponding function calls and decisions. For example terminal outputs from both interpreters, please refer to Appendix A.

![Flowchart of Two Python Interpreters](image)

Figure 26: Flowchart of Two Python Interpreters

The data obtained from each run will be compared to the values generated from the DT simulation to validate the methodology framework. With this testing structure, the \textit{Static DT} will also be validated. The \textit{Energy DT Feedback} in control logic, \textit{AnalysisDT}, and the \textit{Static DT} use the same code to obtain the prediction values. The simulation’s prediction results are integrated with the actual testing data results graph, which will be illustrated in the following section.
6. Results and Discussion

This section will describe, analyze, and discuss the results that are obtained from the testing of the Energy DT. Section 6.1 covers the Energy DT and optimization testing results established in the previous chapter. Section 6.2 displays and explains the parameter monitoring dashboards created in the Ignition software. Lastly, Section 6.3 provides a general discussion tying the research question and major topics to the Energy DT example generated throughout this work. Each section first presents the results followed by a detailed discussion of the results immediately after.

6.1 Energy DT and Optimization Testing Results

The following section will present the Energy DT and optimization results obtained from running the base and validation tests, as well as the DT static simulation. Section 6.1.1 will present the results of the static DT simulation. This is followed by Section 6.1.2 presenting and discussing the results of the base runs conducted for the Energy DT and optimization testing utilizing the CP Lab Heating Tunnel and DT environment created in Python. After validation of the base runs, the remaining variants listed in Table 3 are conducted with results presented in Section 6.1.3. Lastly, Section 6.1.4 will provide a comprehensive analysis and discussion of selected results from the testing runs.

6.1.1 Static DT Simulation Results

This section discusses the results of the static simulation proposed in section 4.3.3. The first part of the Energy DT creates a simulation prediction for one heating cycle of one part of the Heating Tunnel EC based off of user input parameters. The simulation then returns calculated values of the active energy used in kWh (13) and cost of the run in USD (14). The three parameters requested from the user are:

1. Setpoint (in °C, must be between 35 and 60°C)
2. Wattage (0 for 500W or 1 for 1000W)
3. Overshoot time (default 10s)

Once these parameters are set by the user, the simulation conducts the calculations and presents the results, shown in Figure 27, in addition to populating a graph based on the simulation outputs (Figure 28). The example below is for a 40°C setpoint, 500W wattage setting, and 10s overshoot time. The simulation will be further validated in the following sections as the static simulation predication results are plotted on the same temperature graph as the designated testing run.
6.1.2 Base Run Results

Two base runs established in Table 2, one with inputs of 500W power, 40°C setpoint temperature, and 10s overshoot time and another with 1000W, 40°C setpoint temperature, and 10s overshoot were conducted to test the initial validity of the methodology. The data collected for the base runs for 500W and 1000W are shown in Table 4 and Table 5, along with the temperature data and simulation results graphed in Figure 29 and Figure 30. The column names include results of the parameters of interest from the following origins:

- **Original Data**: results of the parameters of interest from the original Heating Tunnel process at a setpoint of 40°C with no modifications.
- **Original Fit**: the values of the parameters of interest from the original fitted equations from Section 4.3.2.
- **Simulation**: results from the DT static simulation.
- **Base Run Results**: results from the base run of modified optimization parameters and DT integration.
- **Percent Difference**: percent difference between Simulation and Base Run Results of the calculated values (13) and (14).

\[
\text{Percent Difference} = \left| \frac{\text{Simulation} - \text{Base Run}}{\text{Base Run}} \right| \times 100\% \tag{15}
\]

- **Percent Savings Theoretical**: savings comparing Original Data and Simulation.

\[
\text{Percent Savings Theoretical} = \left| \frac{\text{Original Data} - \text{Simulation}}{\text{Original Data}} \right| \times 100\% \tag{16}
\]

- **Percent Savings Actual**: savings comparing Original Data and Base Run Results.

\[
\text{Percent Savings Actual} = \left| \frac{\text{Original Data} - \text{Base Run}}{\text{Original Data}} \right| \times 100\% \tag{17}
\]
Table 4: 500W Initial Base Run Results

<table>
<thead>
<tr>
<th></th>
<th>Original Data</th>
<th>Original Fit</th>
<th>Simulation</th>
<th>Base Run Results</th>
<th>Percent Difference</th>
<th>Percent Savings Theoretical</th>
<th>Percent Savings Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time PowerON (s)</td>
<td>37</td>
<td>37</td>
<td>30</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time at Max Temp (s)</td>
<td>52</td>
<td>47</td>
<td>38</td>
<td>48</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max Temp (°C)</td>
<td>43.6</td>
<td>44.14</td>
<td>40.63</td>
<td>41.51</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overshoot time (s)</td>
<td>65</td>
<td>70</td>
<td>10</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Energy Used (kWh)</td>
<td>0.0054</td>
<td>0.0054</td>
<td>0.00466</td>
<td>0.00432</td>
<td>3.24%</td>
<td>17.41%</td>
<td>20.00%</td>
</tr>
<tr>
<td>Cost ($USD)</td>
<td>0.00040</td>
<td>0.00039</td>
<td>0.00033</td>
<td>0.00032</td>
<td>3.12%</td>
<td>17.50%</td>
<td>20.00%</td>
</tr>
</tbody>
</table>

Figure 29: 500W Initial Base Run Temperature Data and Simulation Results Graph
Table 5: 1000W Initial Base Run Results

<table>
<thead>
<tr>
<th>1000W at 40°C Setpoint, 10s Overshoot</th>
<th>Original Data</th>
<th>Original Fit</th>
<th>Simulation</th>
<th>Base Run Results</th>
<th>Percent Difference</th>
<th>Percent Savings Theoretical</th>
<th>Percent Savings Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time PowerON (s)</td>
<td>25</td>
<td>24</td>
<td>13</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time at Max Temp (s)</td>
<td>44</td>
<td>38</td>
<td>25</td>
<td>37</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max Temp (°C)</td>
<td>47.03</td>
<td>47.95</td>
<td>40.41</td>
<td>41.27</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overshoot time (s)</td>
<td>99</td>
<td>96</td>
<td>10</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Energy Used (kWh)</td>
<td>0.00655</td>
<td>0.00655</td>
<td>0.0035</td>
<td>0.00369</td>
<td>5.15%</td>
<td>46.56%</td>
<td>43.66%</td>
</tr>
<tr>
<td>Cost ($USD)</td>
<td>0.00048</td>
<td>0.00048</td>
<td>0.00026</td>
<td>0.00027</td>
<td>3.70%</td>
<td>45.83%</td>
<td>43.75%</td>
</tr>
</tbody>
</table>

Figure 30: 1000W Initial Base Run Temperature Data and Simulation Results

As illustrated in the results above, the maximum temperature the Heating Tunnel reached during the respective runs is 41.51°C for the 500W and 41.27°C for the 1000W for the optimized processes. The simulation prediction for the maximum temperature is 40.63°C for 500W and 40.41°C for 1000W. The maximum temperatures differ by approximately 1°C from recorded data to simulated results. After investigation, it was concluded that the discrepancy is due to not accounting for ramp up current time in the crossflow blower in the simulation. The real-time process has approximately 2s when the crossflow blower is TRUE, but the power level is under the value of peak wattage, equating to approximately 1°C in the linear equations. To account for the difference, the Energy DT Feedback program incorporates an additional 1°C shift vertically down from the original calculated Turn Off Temperature. Consequently, the new Turn Off Temperatures are 36.12°C for 500W and 32.87°C for 1000W.
To account for the new 2s and 1°C incorporation in the results graphs in the Python program, each of the original ranges used to graph the simulation lines were shifted along the x-axis by two seconds to the right. With this inclusion, the graphs now accurately reflect the simulation results, where the simulation starts when the power value reaches the peak wattage value.

Another observation from the initial graphs in Figure 29 and Figure 30, is that the actual temperature data peaks, but then experiences a “leveling” during the duration of the overshoot as the temperature phase transitioned from *Heat Up* to *Rapid Cool*. It was not originally recognized in the original fitting of the data in Section 4.3.2. The leveling can be explained by the thermodynamic properties of cooling with Newton’s Law of Cooling. Newton suggests that the rate of energy transfer in the form of heat depends on the difference in temperature between the body and its environment [23]. Since the Heating Tunnel is a relatively contained environment with minimal heat escape routes, the rate the heat dissipates is low, which explains the instance of leveling in the overshoot after peak temperature is reached. As the *Heat Up* phase energy optimization is a main focus in this research, it was decided to run the remainder of the tests without incorporating the leveling in the simulation.

A new set of base run tests was conducted with inputs of 500W power, 40°C setpoint temperature, and 10s overshoot time and another with 1000W, 40°C setpoint temperature, and 10s overshoot to test the validity of the integration of the 2s and 1°C append. The data collected for the new base runs for 500W and 1000W are shown in Table 6 and Table 7, along with the temperature data and simulation results graphed in Figure 31 and Figure 32.
Table 6: 500W Second Base Run Results

<table>
<thead>
<tr>
<th></th>
<th>Original Data</th>
<th>Original Fit</th>
<th>Simulation</th>
<th>Base Run Results</th>
<th>Percent Difference</th>
<th>Percent Savings Theoretical</th>
<th>Percent Savings Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time PowerON (s)</td>
<td>37</td>
<td>37</td>
<td>30</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time at Max Temp (s)</td>
<td>52</td>
<td>47</td>
<td>38</td>
<td>45</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max Temp (°C)</td>
<td>43.6</td>
<td>44.14</td>
<td>40.63</td>
<td>40.59</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overshoot time (s)</td>
<td>65</td>
<td>70</td>
<td>10</td>
<td>21</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Energy Used (kWh)</td>
<td>0.0054</td>
<td>0.00446</td>
<td>0.00424</td>
<td>5.19%</td>
<td>17.41%</td>
<td>21.48%</td>
<td></td>
</tr>
<tr>
<td>Cost ($USD)</td>
<td>0.0004</td>
<td>0.00033</td>
<td>0.00031</td>
<td>6.45%</td>
<td>17.50%</td>
<td>22.50%</td>
<td></td>
</tr>
</tbody>
</table>

Figure 31: 500W Second Base Run Temperature Data and Simulation Results Graph

There are several observable results extracted from the updated base run at 40°C and 10s overshoot time when comparing the actual temperature data and the simulation results. For the 500W case, the simulation and experimental data have the same PowerON value. This shows that the system reacts comparably to the prediction of the energy consumption model. The actual Maximum Temperature value and the simulated value differ minimally (~0.04°C), which demonstrates that the linear fit model of the Heat Up phase can decently forecast the temperature value. For the 500W model, the predicted overshoot time was set at 10s, whereas the actual overshoot time observed in the experiment was 21s. This is approximately double of what the simulation predicted. This can be explained by the “leveling” of the temperature discovered in the base runs.

Secondly, the Energy Used calculations are slightly different values. This can be explained by the simulation using the average power wattage over a duration of an example Heat Up phase in the
calculation to estimate the Energy Used for a run. In contrast, the Base Run Results column calculates the Energy Used using (13), which is the experimental power consumption during the Heat Up phase. As cost is a direct calculation using the Energy Used value, the same difference exists between the simulation and experimental data. The slight difference in percentages for Energy Used and Cost can be explained for in rounding of such small decimal values.

Table 7: 1000W Second Base Run Results

<table>
<thead>
<tr>
<th>1000W at 40°C Setpoint, 10s Overshoot</th>
<th>Original Data</th>
<th>Original Fit</th>
<th>Simulation</th>
<th>Base Run Results</th>
<th>Percent Difference</th>
<th>Percent Savings Theoretical</th>
<th>Percent Savings Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time PowerON (s)</td>
<td>25</td>
<td>24</td>
<td>12</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time at Max Temp (s)</td>
<td>44</td>
<td>38</td>
<td>25</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max Temp (°C)</td>
<td>47.03</td>
<td>47.95</td>
<td>40.41</td>
<td>40.39</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overshoot time (s)</td>
<td>99</td>
<td>96</td>
<td>10</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Energy Used (kWh)</td>
<td>0.00655</td>
<td>0.00655</td>
<td>0.0035</td>
<td>0.00387</td>
<td>9.56%</td>
<td>46.56%</td>
<td>40.92%</td>
</tr>
<tr>
<td>Cost ($USD)</td>
<td>0.00048</td>
<td>0.00048</td>
<td>0.00026</td>
<td>0.00028</td>
<td>7.14%</td>
<td>45.83%</td>
<td>41.67%</td>
</tr>
</tbody>
</table>

Figure 32: 1000W Second Base Run Temperature Data and Simulation Results

For the 1000W run, the simulation and actual testing data for the Power ON differ by 3s. This difference may be explained by a variety of causes. The first comes with a latency between the actual temperature read by the Python Pallet Sequence function and corresponding Turn off Temperature due to the 1s sampling rate. The way the function is set up, the crossflow blower turns off when the sensor temperature is greater than or exceeds the Turn off Temperature. Since
this value is not exact every time, there is a possibly of variability that the crossflow blower can be TRUE for an extra second. Another cause of variability stems from the starting temperature of each testing run. Every run starts approximately at 26°C, but this value is never exactly the same due to the limitations of the sensors and the 1s sampling rate. When a value starts below 26°C, the power could potentially be on a few extra seconds. In the other extreme, if the value starts slightly above 26°C, the power could be on for a few seconds less than expected.

Comparing the simulation Max Temperature to the test run Max Temperature, the difference is minimal (~0.02°C), which demonstrates that the linear fit model of the Heat Up phase can decently forecast the temperature value for this case. The predicted overshoot time for the simulation is 10s, where the actual value for the test run was 16s. The difference can be explained again by the “leveling” effect the temperature faces while it is transitioning from Heat Up to Rapid Cool phases. The Energy Used and Cost calculations are slightly different, similarity to the 500W case, explained by the use of the average power wattage over a duration of an example Heat Up phase in the calculation to estimate the Energy Used for a run in the simulation. As cost is a direct calculation using the Energy Used value, the difference exists as well between the simulation and experimental data. Again, the slight change in percentages in Energy Used and Cost can be explained for in rounding of such small decimal values.
6.1.3 Validation Runs

This section will display and discuss the results of the validation runs established in Table 3. For the 10s overshoot testing results of setpoints 45°C and 50°C at 500W and 1000W, the data values collected and calculated are the same as the results in Table 6 and 7. In order to have comparable results and calculate the Percent Savings columns, runs of the original Festo Process using the MES with a modified setpoints of 45°C and 50°C were conducted at the 500W and 1000W power levels. The Original Data column reflects the results of the original Festo process runs at the respective parameter settings.

500W Setpoint: 45°C Overshoot: 10s

Table 8: 500W Setpoint: 45°C, Overshoot: 10s Results

<table>
<thead>
<tr>
<th>500W at 45°C Setpoint, 10s Overshoot</th>
<th>Original Data</th>
<th>Simulation</th>
<th>Base Run Results</th>
<th>Percent Difference</th>
<th>Percent Savings Theoretical</th>
<th>Percent Savings Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time PowerON (s)</td>
<td>51</td>
<td>43</td>
<td>40</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time at Max Temp (s)</td>
<td>100</td>
<td>51</td>
<td>52</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max Temp (°C)</td>
<td>48.21</td>
<td>45.7</td>
<td>45.81</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overshoot time (s)</td>
<td>46</td>
<td>10</td>
<td>19</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Energy Used (kWh)</td>
<td>0.0074</td>
<td>0.00639</td>
<td>0.00564</td>
<td>13.30%</td>
<td>13.65%</td>
<td>23.78%</td>
</tr>
<tr>
<td>Cost ($USD)</td>
<td>0.00054</td>
<td>0.00047</td>
<td>0.00041</td>
<td>14.63%</td>
<td>12.96%</td>
<td>24.07%</td>
</tr>
</tbody>
</table>

Figure 33: 500W Setpoint: 45°C, Overshoot: 10s Graph
500W Setpoint: 50°C Overshoot: 10s

Table 9: 500W Setpoint: 50°C, Overshoot: 10s Results

<table>
<thead>
<tr>
<th></th>
<th>Original Data</th>
<th>Simulation</th>
<th>Base Run Results</th>
<th>Percent Difference</th>
<th>Percent Savings Theoretical</th>
<th>Percent Savings Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time PowerON (s)</td>
<td>62</td>
<td>56</td>
<td>49</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time at Max Temp (s)</td>
<td>102</td>
<td>64</td>
<td>60</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max Temp (°C)</td>
<td>52.89</td>
<td>50.77</td>
<td>50.35</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overshoot time (s)</td>
<td>41</td>
<td>10</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Energy Used (kWh)</td>
<td>0.00903</td>
<td>0.00832</td>
<td>0.00704</td>
<td>18.18%</td>
<td>7.86%</td>
<td>22.04%</td>
</tr>
<tr>
<td>Cost ($USD)</td>
<td>0.00066</td>
<td>0.00061</td>
<td>0.00052</td>
<td>17.31%</td>
<td>7.58%</td>
<td>21.21%</td>
</tr>
</tbody>
</table>

Figure 34: 500W Setpoint: 50°C, Overshoot: 10s Graph
1000W Setpoint: 45°C Overshoot: 10s

Table 10: 1000W Setpoint: 45°C, Overshoot: 10s Results

<table>
<thead>
<tr>
<th></th>
<th>Original Data</th>
<th>Simulation</th>
<th>Base Run Results</th>
<th>Percent Difference</th>
<th>Percent Savings Theoretical</th>
<th>Percent Savings Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time PowerON (s)</td>
<td>32</td>
<td>21</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time at Max Temp (s)</td>
<td>76</td>
<td>34</td>
<td>42</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max Temp (°C)</td>
<td>51.97</td>
<td>45.63</td>
<td>46.01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overshoot time (s)</td>
<td>81</td>
<td>10</td>
<td>26</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Energy Used (kWh)</td>
<td>0.00828</td>
<td>0.00565</td>
<td>0.00575</td>
<td>1.74%</td>
<td>31.76%</td>
<td>30.56%</td>
</tr>
<tr>
<td>Cost ($USD)</td>
<td>0.00061</td>
<td>0.00041</td>
<td>0.00042</td>
<td>2.38%</td>
<td>32.79%</td>
<td>31.15%</td>
</tr>
</tbody>
</table>

Figure 35: 1000W Setpoint: 45°C, Overshoot: 10s Graph
1000W Setpoint: 50°C Overshoot: 10s

Table 11: 1000W Setpoint: 50°C, Overshoot: 10s Results

<table>
<thead>
<tr>
<th></th>
<th>Original Data</th>
<th>Simulation</th>
<th>Base Run Results</th>
<th>Percent Difference</th>
<th>Percent Savings Theoretical</th>
<th>Percent Savings Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time PowerON (s)</td>
<td>37</td>
<td>30</td>
<td>29</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time at Max Temp (s)</td>
<td>81</td>
<td>43</td>
<td>46</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max Temp (°C)</td>
<td>56.56</td>
<td>50.85</td>
<td>51.14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overshoot time (s)</td>
<td>68</td>
<td>10</td>
<td>24</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Energy Used (kWh)</td>
<td>0.00972</td>
<td>0.00807</td>
<td>0.00732</td>
<td>10.25%</td>
<td>16.98%</td>
<td>24.69%</td>
</tr>
<tr>
<td>Cost ($USD)</td>
<td>0.00071</td>
<td>0.00059</td>
<td>0.00054</td>
<td>9.26%</td>
<td>16.90%</td>
<td>23.94%</td>
</tr>
</tbody>
</table>

Figure 36: 1000W Setpoint: 50°C, Overshoot: 10s Graph
The remaining validation test results are shown in Tables 12-23 and display the same reportable parameter values of interest as the previous tables (row parameters). As a slight difference, the column categories are limited to *Simulation*, *Run Results*, and *Percent Difference* due to excluding the comparison of original runs.

**500W Setpoint: 40°C Overshoot: 15s**

Table 12: 500W Setpoint: 40°C, Overshoot: 15s Results

<table>
<thead>
<tr>
<th>500W at 40°C Setpoint, 15s Overshoot</th>
<th>Simulation</th>
<th>Run Results</th>
<th>Percent Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time PowerON (s)</td>
<td>31</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>Time at Max Temp (s)</td>
<td>39</td>
<td>47</td>
<td></td>
</tr>
<tr>
<td>Max Temp (°C)</td>
<td>41.02</td>
<td>40.83</td>
<td></td>
</tr>
<tr>
<td>Overshoot time (s)</td>
<td>15</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>Energy Used (kWh)</td>
<td>0.00461</td>
<td>0.00474</td>
<td>2.74%</td>
</tr>
<tr>
<td>Cost ($USD)</td>
<td>0.00034</td>
<td>0.00035</td>
<td>2.86%</td>
</tr>
</tbody>
</table>

Figure 37: 500W Setpoint: 40°C, Overshoot: 15s Graph
500W Setpoint: 45°C Overshoot: 15s

Table 13: 500W Setpoint: 45°C, Overshoot: 15s Results

<table>
<thead>
<tr>
<th>500W at 45°C Setpoint, 15s Overshoot</th>
<th>Simulation</th>
<th>Run Results</th>
<th>Percent Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time PowerON (s)</td>
<td>43</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td>Time at Max Temp (s)</td>
<td>52</td>
<td>56</td>
<td></td>
</tr>
<tr>
<td>Max Temp (°C)</td>
<td>46.09</td>
<td>45.71</td>
<td></td>
</tr>
<tr>
<td>Overshoot time (s)</td>
<td>15</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>Energy Used (kWh)</td>
<td>0.00654</td>
<td>0.00619</td>
<td>5.65%</td>
</tr>
<tr>
<td>Cost ($USD)</td>
<td>0.00048</td>
<td>0.00045</td>
<td>6.67%</td>
</tr>
</tbody>
</table>

Figure 38: 500W Setpoint: 45°C, Overshoot: 15s Graph
500W Setpoint: 50°C Overshoot: 15s

Table 14: 500W Setpoint: 50°C, Overshoot: 15s Results

<table>
<thead>
<tr>
<th></th>
<th>Simulation</th>
<th>Run Results</th>
<th>Percent Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time PowerON (s)</td>
<td>57</td>
<td>51</td>
<td></td>
</tr>
<tr>
<td>Time at Max Temp (s)</td>
<td>65</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td>Max Temp (°C)</td>
<td>51.16</td>
<td>50.84</td>
<td></td>
</tr>
<tr>
<td>Overshoot time (s)</td>
<td>15</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>Energy Used (kWh)</td>
<td>0.00847</td>
<td>0.00747</td>
<td>13.39%</td>
</tr>
<tr>
<td>Cost ($USD)</td>
<td>0.00062</td>
<td>0.00055</td>
<td>12.73%</td>
</tr>
</tbody>
</table>

Figure 39: 500W Setpoint: 50°C, Overshoot: 15s Graph
1000W Setpoint: 40°C Overshoot: 15s

Table 15: 1000W Setpoint: 40°C, Overshoot: 15s Results

<table>
<thead>
<tr>
<th></th>
<th>Simulation</th>
<th>Run Results</th>
<th>Percent Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time PowerON (s)</td>
<td>13</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>Time at Max Temp (s)</td>
<td>26</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>Max Temp (°C)</td>
<td>40.99</td>
<td>41.22</td>
<td></td>
</tr>
<tr>
<td>Overshoot time (s)</td>
<td>15</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>Energy Used (kWh)</td>
<td>0.0035</td>
<td>0.00387</td>
<td>9.56%</td>
</tr>
<tr>
<td>Cost ($USD)</td>
<td>0.00026</td>
<td>0.00028</td>
<td>7.14%</td>
</tr>
</tbody>
</table>

Figure 40: 1000W Setpoint: 40°C, Overshoot: 15s Graph
1000W Setpoint: 45°C Overshoot: 15s

Table 16: 1000W Setpoint: 45°C, Overshoot: 15s Results

<table>
<thead>
<tr>
<th>1000W at 45°C Setpoint, 15s Overshoot</th>
<th>Simulation</th>
<th>Run Results</th>
<th>Percent Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time PowerON (s)</td>
<td>22</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>Time at Max Temp (s)</td>
<td>35</td>
<td>39</td>
<td></td>
</tr>
<tr>
<td>Max Temp (°C)</td>
<td>46.21</td>
<td>46.98</td>
<td></td>
</tr>
<tr>
<td>Overshoot time (s)</td>
<td>15</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>Energy Used (kWh)</td>
<td>0.00592</td>
<td>0.00523</td>
<td>13.19%</td>
</tr>
<tr>
<td>Cost ($USD)</td>
<td>0.00043</td>
<td>0.00038</td>
<td>13.16%</td>
</tr>
</tbody>
</table>

Figure 41: 1000W Setpoint: 45°C, Overshoot: 15s Graph
1000W Setpoint: 50°C Overshoot: 15s

Table 17: 1000W Setpoint: 50°C, Overshoot: 15s Results

<table>
<thead>
<tr>
<th></th>
<th>Simulation</th>
<th>Run Results</th>
<th>Percent Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time PowerON (s)</td>
<td>32</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>Time at Max Temp (s)</td>
<td>44</td>
<td>48</td>
<td></td>
</tr>
<tr>
<td>Max Temp (°C)</td>
<td>51.43</td>
<td>52.36</td>
<td></td>
</tr>
<tr>
<td>Overshoot time (s)</td>
<td>15</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>Energy Used (kWh)</td>
<td>0.00861</td>
<td>0.00785</td>
<td>9.68%</td>
</tr>
<tr>
<td>Cost ($USD)</td>
<td>0.00063</td>
<td>0.00057</td>
<td>10.53%</td>
</tr>
</tbody>
</table>

Figure 42: 1000W Setpoint: 50°C, Overshoot: 15s Graph
500W Setpoint: 40°C Overshoot: 20s

Table 18: 500W Setpoint: 40°C, Overshoot: 20s Results

<table>
<thead>
<tr>
<th>500W at 40°C Setpoint, 20s Overshoot</th>
<th>Simulation</th>
<th>Run Results</th>
<th>Percent Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time PowerON (s)</td>
<td>32</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>Time at Max Temp (s)</td>
<td>40</td>
<td>47</td>
<td></td>
</tr>
<tr>
<td>Max Temp (°C)</td>
<td>41.41</td>
<td>41.86</td>
<td></td>
</tr>
<tr>
<td>Overshoot time (s)</td>
<td>20</td>
<td>39</td>
<td></td>
</tr>
<tr>
<td>Energy Used (kWh)</td>
<td>0.00476</td>
<td>0.00421</td>
<td>13.06%</td>
</tr>
<tr>
<td>Cost ($USD)</td>
<td>0.00035</td>
<td>0.00031</td>
<td>12.90%</td>
</tr>
</tbody>
</table>

Figure 43: 500W Setpoint: 40°C, Overshoot: 20s Graph
500W Setpoint: 45°C Overshoot: 20s

Table 19: 500W Setpoint: 45°C, Overshoot: 20s Results

<table>
<thead>
<tr>
<th>500W at 45°C Setpoint, 20s Overshoot</th>
<th>Simulation</th>
<th>Run Results</th>
<th>Percent Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time PowerON (s)</td>
<td>45</td>
<td>41</td>
<td></td>
</tr>
<tr>
<td>Time at Max Temp (s)</td>
<td>53</td>
<td>53</td>
<td></td>
</tr>
<tr>
<td>Max Temp (°C)</td>
<td>46.48</td>
<td>46.25</td>
<td></td>
</tr>
<tr>
<td>Overshoot time (s)</td>
<td>20</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>Energy Used (kWh)</td>
<td>0.00669</td>
<td>0.00575</td>
<td>16.35%</td>
</tr>
<tr>
<td>Cost ($USD)</td>
<td>0.00049</td>
<td>0.00042</td>
<td>16.67%</td>
</tr>
</tbody>
</table>

Figure 44: 500W Setpoint: 45°C, Overshoot: 20s Graph
500W Setpoint: 50°C Overshoot: 20s

Table 20: 500W Setpoint: 50°C, Overshoot: 20s Results

<table>
<thead>
<tr>
<th>500W at 50°C Setpoint, 20s Overshoot</th>
<th>Simulation</th>
<th>Run Results</th>
<th>Percent Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time PowerON (s)</td>
<td>58</td>
<td>51</td>
<td></td>
</tr>
<tr>
<td>Time at Max Temp (s)</td>
<td>66</td>
<td>63</td>
<td></td>
</tr>
<tr>
<td>Max Temp (°C)</td>
<td>51.55</td>
<td>50.94</td>
<td></td>
</tr>
<tr>
<td>Overshoot time (s)</td>
<td>20</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>Energy Used (kWh)</td>
<td>0.00862</td>
<td>0.0072</td>
<td>19.72%</td>
</tr>
<tr>
<td>Cost ($USD)</td>
<td>0.00063</td>
<td>0.00053</td>
<td>18.87%</td>
</tr>
</tbody>
</table>

Figure 45: 500W Setpoint: 50°C, Overshoot: 20s Graph
1000W Setpoint: 40°C Overshoot: 20s

Table 21: 1000W Setpoint: 40°C, Overshoot: 20s Results

<table>
<thead>
<tr>
<th></th>
<th>Simulation</th>
<th>Run Results</th>
<th>Percent Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time PowerON (s)</td>
<td>13</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>Time at Max Temp (s)</td>
<td>26</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>Max Temp (°C)</td>
<td>40.99</td>
<td>41.51</td>
<td></td>
</tr>
<tr>
<td>Overshoot time (s)</td>
<td>20</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>Energy Used (kWh)</td>
<td>0.0035</td>
<td>0.00382</td>
<td>8.38%</td>
</tr>
<tr>
<td>Cost ($USD)</td>
<td>0.00026</td>
<td>0.00028</td>
<td>7.14%</td>
</tr>
</tbody>
</table>

Figure 46: 1000W Setpoint: 40°C, Overshoot: 20s Graph
1000W Setpoint: 45°C Overshoot: 20s

Table 22: 1000W Setpoint: 45°C, Overshoot: 20s Results

<table>
<thead>
<tr>
<th></th>
<th>Simulation</th>
<th>Run Results</th>
<th>Percent Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time PowerON (s)</td>
<td>22</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>Time at Max Temp (s)</td>
<td>35</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>Max Temp (°C)</td>
<td>46.21</td>
<td>47.03</td>
<td></td>
</tr>
<tr>
<td>Overshoot time (s)</td>
<td>20s</td>
<td>37</td>
<td></td>
</tr>
<tr>
<td>Energy Used (kWh)</td>
<td>0.00592</td>
<td>0.00574</td>
<td>3.14%</td>
</tr>
<tr>
<td>Cost ($USD)</td>
<td>0.00043</td>
<td>0.00042</td>
<td>2.38%</td>
</tr>
</tbody>
</table>

Figure 47: 1000W Setpoint: 45°C, Overshoot: 20s Graph
1000W Setpoint: 50°C Overshoot: 20s

Table 23: 1000W Setpoint: 50°C, Overshoot: 20s Results

<table>
<thead>
<tr>
<th>1000W at 50°C Setpoint, 20s Overshoot</th>
<th>Simulation</th>
<th>Run Results</th>
<th>Percent Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time PowerON (s)</td>
<td>32</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td>Time at Max Temp (s)</td>
<td>44</td>
<td>48</td>
<td></td>
</tr>
<tr>
<td>Max Temp (°C)</td>
<td>51.43</td>
<td>52.31</td>
<td></td>
</tr>
<tr>
<td>Overshoot time (s)</td>
<td>20</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>Energy Used (kWh)</td>
<td>0.00861</td>
<td>0.00724</td>
<td>18.92%</td>
</tr>
<tr>
<td>Cost ($USD)</td>
<td>0.00063</td>
<td>0.00053</td>
<td>18.87%</td>
</tr>
</tbody>
</table>

Figure 48: 1000W Setpoint: 50°C, Overshoot: 20s Graph
6.1.4 Energy Optimization Discussion

This section discusses the energy optimization results with respect to the research questions. The areas of discussion include maximum temperature prediction, overshoot time, energy savings, and implementation of leveling in the simulation.

6.1.4.1 Maximum Temperature Prediction

One of the set goals of the energy optimization model for the temperature overshoot was to not exceed 5% of the setpoint temperature. The following subplots in Figure 49 display comparisons of the maximum calculated temperature of the simulation (prediction) versus the actual maximum temperature recorded in the respective testing run. Each graph represents a specific wattage and setpoint temperature value. The contents of the graph include the comparison values of simulation prediction maximum temperature and the actual test run maximum temperature value grouped by the overshoot time appropriately labeled on the x-axis. To aid with comparison, the threshold temperature value (5% of setpoint), is highlighted as a straight horizontal line on each respective chart.
As a first observation, each run exceeds the indicated setpoint value for the test run. Also, all the recorded runs with their designated setpoint values remain below the 5% threshold values set by the energy optimization goals. These are notable combined results because it shows the Turn off Temperature feedback parameter indicated by the Feedback DT sent back to the Heating Tunnel control logic was accurate at maintaining the temperature within the specified design limits. With the linear regression energy optimization method selected, it is difficult to predict the peak temperature precisely. Even though the simulation did not predict the maximum temperature exactly, it is still considered within the optimization goal range. Hence, the results are considered acceptable predictions for the set objective.

For the original optimization scenario of a setpoint of 40°C and 10s overshoot, the simulation was close to predicting the actual max temperature for both 500W (40.63°C to 40.59°C) and 1000W (40.41°C to 40.39°C). As for the remaining test runs of 500W, the prediction was not as accurate, as with the graphs showing a ranging variability between the prediction and actual maximum temperature values. For all the remaining test runs of the 1000W variants, the simulation prediction was less than the actual maximum temperature of the Heating Tunnel, but it remained below the threshold value. The difference in actual maximum temperature value versus the simulation prediction might be explained by the difference in current amperage at each wattage level. The equation for electrical power is $P = VI$. In this instance, voltage remains 240V, so in order to obtain a higher power value, the current must increase. The 1000W power pull of the crossflow blower heating element has a higher current amperage than the 500W at the original 40°C setpoint. In result, the heating element gets hotter and has an increase in energy, and therefore retains more heat causing it to peak at a higher temperature than the simulation predicts. Other process variability factors impacting the maximum temperature prediction versus the test value of a run include the inconsistency in starting temperature for each test run and the sampling rate. Each test run was started at approximately 26°C, but this value is unable to be achieved exactly as predicted every time due to the limitations of the system sensors and sampling rate. The minor variance in this value can cause the power to the crossflow blower to be TRUE, or even FALSE, an additional second(s), impacting the time power is drawn, and either result in increasing or decreasing the max temperature value read by the Heating Tunnel sensors. Similarly, the 1s sampling rate causes variability in the Heat Up process. The control logic turns off the crossflow blower once the
temperature sensor exceeds the *Turn off Temperature*. By applying the knowledge of basic physics, it is known the temperature is going to increase, but the crossflow blower will be on an additional second, consequently drawing additional power and therefore leading to an increase in maximum temperature during the heating process.

6.1.4.2 Overshoot Prediction

The overshoot prediction section will discuss the results revolving around the designed overshoot time value established in the energy optimization methodology and the analyzed time during the test runs. The bar graphs in Figure 50 group the runs based on their testing wattage (500W or 1000W) in separate graphs. In the respective graphs, the runs are clustered based on their predicted overshoot times (10s, 15s, 20s) at each testing temperature (40°C, 45°C, 50°C). Each bar visualizes the actual overshoot time analyzed during a testing run. For visual comparison purposes, a horizontal line based on each overshoot time is placed in each cluster.

![Figure 50: Overshoot Prediction Graphs](image)

As a first result, each of the test runs remained in overshoot, or above the setpoint temperature for the manufacturing process, for the requirement of at least 5s. With this result, the manufacturing process requirement for heating the part was satisfied. As viewed in the 500W overshoot graph, the 50°C trials at 10s and 20s fell the closest to the predicted overshoot time, only differing by 1s compared to the predicted value. For the other actual overshoot times, they exceeded the predicted mark. The 500W test runs do not follow an observable trend other than exceeding the benchmark value. The observation of the temperature experiencing a varying leveling time in Section 6.1.1 Base Run Results when transitioning from *Heat Up* to *Rapid Cool* phases can aid in explaining the difference in actual to predicted overshoot times. In the 1000W overshoot graph, the 40°C at 10s with an actual value of 16s was the nearest test to the actual predicted overshoot time of 10s. The other overshoot times well exceed the predicted overshoot time. This can be supported again by the leveling the temperature experiences when it is in overshoot range transitioning from *Heat Up* to *Rapid Cool* phases. The 1000W cases follow an approximate 10s of leveling aside from the original Base Run of 40°C at 10s.

To support the argument, a theoretical leveling line was implemented into two selected test runs, one at 500W (Figure 51) and one at 1000W (Figure 52). To account for the leveling of the temperature during overshoot, a constant of the maximum peak simulation temperature for a
duration of 10s was added followed by the Cool Down prediction line. The original test runs are located beside the leveling graph for comparison.

Figure 51: 500W Setpoint: 40°C, Overshoot: 10s Original Test Run versus Leveling

Viewed in both implementation of leveling graphs in Figure 51 and Figure 52, the addition of the leveling line improved the prediction of the overshoot and Rapid Cool phase. With the leveling integration, it could further reduce the energy consumption of the Heating Tunnel during the Heat Up phase. It is to be noted that the main focus of this research is the optimization of the Heat Up phase, which is the power drawn to heat up the Heating Tunnel to the appropriate setpoint temperature. For this reason, it was decided to not implement the leveling into the DT model. Instead, a future research direction is opened for investigating the leveling aspect and integration of Rapid Cool phase.

6.1.4.3 Energy Savings

This section will analyze and discuss the energy savings that were realized by implementing the energy optimizing DT into the Heating Tunnel station of the CP Lab. Figure 53 illustrates the
percent energy savings of the optimized feedback DT test results of the targeted 10s overshoot time compared to the original run data of the setpoint value using the Festo MES software. It has to be noted, that given the scale of the testbed, the dollar value of realized savings have limited meaning and the percent savings are considered more indicative of the potential when scaled to industrial systems.

First, all test runs conducted of the energy optimizing feedback DT resulted in energy savings. The realization of savings in all cases shows that the DT model did in fact optimize the energy consumption of the existing Heating Tunnel process compared to the factory implementation. The variability comes with the amount of energy savings generated from the selected test runs. As illustrated in Figure 53, all of the energy savings for the 500W test runs at the 40°C, 45°C, and 50°C setpoint temperatures are all approximately 20%. All of the 500W original data maximum temperature values peaked roughly 3°C above the designated setpoint temperature, and the actual test runs all showed a max temperature similar to the DT predicted maximum temperature value. Therefore, it leads to a consistent energy savings value throughout the different 500W variants.

The maximum percentage of energy savings occurs in then 1000W at 40°C setpoint test case. The test run displayed 40.92% energy savings compared to the original data. The savings in this case are significant, and further proving that the original process could be optimized with the implementation an energy optimizing feedback DT. Viewed in the 1000W test runs, there is an observable steady decrease in energy savings from 40°C, 45°C, and 50°C. The decrease stems from the maximum temperature values collected in the original data. The 40°C case had a peak temperature approximately 7°C above the setpoint, whereas the 45°C and 50°C cases had peaks of approximately 6°C above the designated setpoint value. With the variability in the system due to sampling rate and initial run starting temperature variations, the peak temperatures are not exact every run. Overall, the energy optimizing DT model introduced in the methodology produces
significant energy savings into the Heating Tunnel process, and results in direct cost savings of energy consumption.

6.2 Parameter Monitoring

This section will display and explain the created Ignition Dashboards introduced in Section 4.3.1 used for real-time parameter monitoring of the Heating Tunnel during each test run. Figure 54 presents two screenshots from the created Ignition dashboard designer. In each subfigure, the top left graph visualizes the Current (A), top right the Temperature of Heating Tunnel (°C), bottom left the Voltage (V), and lastly the Active Power (W). It is important to note that for viewing purposes, the Current, Voltage, and Active Power charts have a longer historical view of 10 hours. This is to emphasize the peaks in the values. The Temperature data is within the last 5 minutes to focus on the temperature feedback optimization parameter being tested.

Figure 54: a) Heat Up Phase Dashboard b) Complete Cycle Dashboard
To create the dashboards, the OPC UA servers for both the Heating Tunnel and EMB are connected to the Ignition Server. Once a connection is established, the node addresses for each of the parameters of interest are able to be browsed. The nodes of interest are added to the Ignition designer to be monitored and added to its own individual Power Chart. This type of chart displays real-time data based on a 1s sampling rate and historical node data. Each chart also provides the ability to actively view the minimum, maximum, average, and current value of the node, not to be mixed up with electrical current. All nodes are connected to an SQLite database in order to record and plot the historical data.

Figure 54a displays the real-time data of a Heat Up phase for a test run. The parameters are actively changing in real-time reflecting what is occurring in the system. Current, Voltage, and Active Power are all at their respective values when heating occurs, and the Temperature is actively increasing. Figure 54b illustrates the real-time data but for a complete phase of Heat Up and Rapid Cool. As expected, Current, Voltage, and Active Power are all at resting values and not changing, but the Temperature is decreasing slowly due to cooling.

To mimic the manufacturing environment, Ignition dashboards are a straightforward data monitoring platform to implement on a standalone device. Even though the dashboards were used for a basic purpose of visualizing and monitoring, they were most helpful during testing by providing the ability of viewing the maximum temperature and input power values of the runs in real-time, in addition to ensuring the temperature cooled down to the appropriate temperature for the start of the next run. The operator was able to remain at the computer for longer periods of time, resulting in an increase of value added time during testing focusing on data and process analysis.

6.3 General Discussion

The question asked prior to this research was: how can we use the features of OPC UA to close the energy optimization feedback loop between the physical SM lab system Heating Tunnel and virtual Python DT environment? The research conducted in this thesis has generated a possible solution to this question by utilizing the OPC UA servers in both the Heating Tunnel and EMB, and the OPC UA client-server model. Parameter values can be effectively read to the virtual Python DT environment, and once an optimized parameter, Turn off Temperature, is obtained, it can be written back to the Heating Tunnel server using the OPC UA client-server feature in Python.

The comprehensive literature review [11] showed that there were no examples of a ‘true’ Energy DT utilizing OPC UA to conduct data flow from physical space and digital space, and vice versa. Examples that used OPC UA in their problem solving approach only existed of the DS digital technology, where there was only physical to virtual communication and no feedback implementation. The only previous approach that created a ‘true’ Energy DT was that of Park et. al [12] that utilized a MSB with SOAP. This research contributes another example to the Energy DT state of the art by developing an Energy DT use case and the accompanying method to generate energy optimizing feedback in the digital space back to the physical space. The research effectively created an energy optimizing DT using the CP Lab Heating Tunnel as the physical asset of the DT
structure. For connection, the Heating Tunnel OPC UA servers are connected to a virtual DT environment developed using Python to satisfy the digital space requirement. Within the virtual Python environment, the control logic was recreated due to OPC UA writable feedback limitations of the physical system. The programs used a client-server model for reading and/or writing parameter values. For energy optimization and feedback, the original process of the Heating Tunnel was optimized by creating a linear prediction model for temperature in the Heat Up phase of the manufacturing process. Once parameters of targeted overshoot time, wattage, and setpoint were fed into the DT model, it calculates and returns the new energy optimization parameter, Turn off Temperature. This value was sent back to the manufacturing process control logic in real-time to initiate the crossflow blower to turn off, and therefore save on power consumed.

The method and optimization technique needed to be tested and validated to contribute to the state of the art of Energy DTs. Multiple different validation run tests were conducted to test the effectiveness of the developed DT model and its corresponding feedback parameter. With the Heat Up phase of the Heating Tunnel process as the focal point of optimization, the test results overall showed a reduction in power consumed, and therefore reducing the energy consumption and cost during the Heat Up phase. The implementation of the Energy DT in the Heating Tunnel process incorporated three energy applications mentioned in Section 2.4: EC Monitoring, EC Analysis, and EC Optimization. EC Monitoring was accomplished through the Ignition dashboards and reading OPC UA parameter values during the test runs. EC Analysis was achieved through the model prediction of Turn off Temperature, as well as the model predictions obtained through running the static DT. Lastly, the EC Optimization was accomplished through the parameter optimization of the maximum temperature the Heating Tunnel reached during the manufacturing process and reduction of the overshoot time for various different input parameters.

From the energy optimization results, the maximum temperature the Heating Tunnel reached during each run was contained within the +5% range from the setpoint temperature. The simulation was not a perfect prediction method for maximum temperature, but the value stays below the threshold in all cases and leads to a reduction of excess energy usage. The prediction and optimization method can be further improved with implementation of automatic data analysis and ML within the process. The overshoot prediction time was not as accurate as intended due to the omission of leveling in the DT model. However, the overshoot time still met the manufacturing requirements of 5s of baking at the setpoint temperature. Since the focus of the optimization was the Heat Up phase, a future investigation is needed to include leveling to create more accurate DT of Rapid Cool phase. Overall, an occurrence of energy savings in each test showed that the DT energy optimizing feedback was effective when implemented in the process.

Ultimately, the goal of a closed-loop feedback Energy DT was accomplished in this research. The framework for connection of DT using OPC UA and Python combination with an energy focus is novel and includes a modular optimization model that can be extended to improve the prediction further. It is important to note that the objective of this research was the implementation of the closed-loop feedback with an energy perspective. The energy optimization method and results themselves are considered supplemental to achieving the closed loop feedback goal and are used to evaluate and validate the utility of the implementation.
Several limitations of the research exist and need to be noted. As with all experiments which involve human research efforts, one limitation of this study is the potential for human error during the testing phase. To mitigate this as much as possible, a well-detailed methodology for each phase was formulated and followed. Secondly, the test runs conducted for the results to validate the Energy DT and optimization were conducted using one singular pallet in a lab setup. Each test produces a low energy usage and low cost, but a high energy savings percentage (12.22% to 40.92%). In a real world context, an electromagnetic paint operation oven used in the process of curing paint during the automobile manufacturing process is estimated to consume 9.42x10^6 kWh of energy (from [24], based on [25]). At the US rate of $0.0809 per kWh [2], the consumption would total $762,083. Applying the 40% energy savings would equate to a theoretical potential of $304,833 in savings. Another limitation includes the usage of the sampling rate of 1s in the virtual Python environment. The selected rate can cause delays in parameter readings and writing throughout the testing process, causing slight variations between tests when re-running trials. Lastly, the linear models created in the methodology and applied to the testing runs were assumptions drawn from initial data collection and analysis. As this solution is not the most accurate approach in process control, it ultimately reaches the goal of the research to create and test a closed-loop Energy DT.
7. Conclusion

Energy DTs are becoming more popular in the manufacturing sector as the rise of energy costs have led manufacturers to search for new solutions to conserve energy in their processes. Energy DTs in manufacturing are still a relatively new but growing research topic with minimal examples of ‘true’ DT applications. The aim of this thesis work is to create a case study of an Energy DT utilizing the OPC UA communication method and contribute to the state of the art of the topic. The Energy DT utilizes the physical space of the CP Lab Heating Tunnel, a digital space created in Python including the control logic and the energy optimizing DT model with the bidirectional data highway consisting of OPC UA communication protocol. For the DT, a linear model for predicting temperature over time is established, where input parameters of setpoint temperature for the Heating Tunnel, power level, and overshoot time are set by the user. The parameter of Turn off Temperature is calculated by the DT using the input parameters read in the real-time heating process, and then returned from the DT to the Heating Tunnel to reflect the change in the real-time process.

The method is validated by conducting a set of experiments using an array of trial runs varying the input parameters. After all validation runs were conducted, it was shown that the optimization was in fact effective in reducing energy consumption reflected by all trial runs returning an energy consumption savings. The linear model chosen for the energy optimization method predicted the maximum temperature the Heating Tunnel reached during each run within the +5% range from the setpoint temperature. The model was not precise in predicting the overshoot time due to a discovery of temperature leveling during the time the temperature is in overshoot. As the Heat Up phase of the heating process us is the focus of the thesis research, the integration of the leveling has been viewed as a future working consideration. Overall, the research completed in this thesis successfully completed the task of creating an energy focused DT with bidirectional feedback. Though the DT uses a basic energy optimization approach, it emphasizes the potential impact of Energy DT applications in any process. This work is a steppingstone in academia to explore the promising research to be completed in this realm of Energy DTs.

The future working considerations for this thesis work include integrating a different method of energy optimization into the Energy DT. Possibilities include investigating automatic data analysis and ML techniques to continuously optimize the process based off of previous run data. Since the main focus of this thesis work was to optimize the Heat Up phase of the Heating Tunnel process, the Rapid Cool phase was not altered from the original methodology. Further investigation can be conducted to analyze the leveling trends in the different wattage settings to include a leveling phase into the simulation prediction to create more accurate DT of Rapid Cool phase. Secondly, since the structure of this work closed the feedback loop with OPC UA and a local PC, the next phase would be to stream the machine data to the cloud to store the data and conduct various analyses. Lastly, since this work concentrates on singular pallet energy optimization, the next phase would be to integrate multiple pallets and batch sizes to expand the value of the Energy DT. This would require the incorporation of the MES to manage orders and conduct full part processing using the other stations in the CP Lab.
8. References


9. Appendix A

Figure A- 1: Example Output Python Interpreter 1 for 1000W 45°C, 10s overshoot

Figure A- 2: Example Output Python Interpreter 2 for 1000W 45°C, 10s overshoot